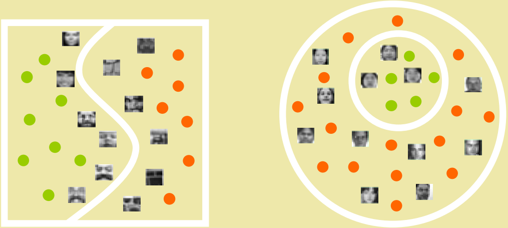
Face Verification for Mobile Personal Devices



Qian Tao

Face Verification for Mobile Personal Devices

Qian Tao

Face Verification for Mobile Personal Devices
Qian Tao
Ph.D. Thesis
University of Twente, The Netherlands

ISBN: 978-90-365-2793-4

The work described in this thesis was performed at the Signals and Systems Group, Faculty of Electrical Engineering, Mathematics, and Computer Science, University of Twente, The Netherlands. The research was financially supported by the Freeband Netherlands for the PNP2008 project, and the European Commission for the 3D-Face project.

FACE VERIFICATION
FOR MOBILE PERSONAL DEVICES

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
prof.dr. H. Brinksma,
on account of the decision of the graduation committee,
to be publicly defended on Friday 6 February 2009 at 15:00

by

Qian Tao

born on 1 January 1980

in Dangyang, China

Promotion Committee:

Prof.dr.ir. C.H. Slump (University of Twente, promotor)
Dr.ir.R.N.J. Veldhuis (University of Twente, assistant promotor)
Prof.dr.ir.G.J.M. Smit (University of Twente)
Prof.dr. S. Stramigioli (University of Twente)
Prof.dr.ir. P.H.N. de With (Eindhoven University of Technology)
Prof.dr.ir. S.M. Heemstra de Groot (Delft University of Technology)
Dr. A.A. Ross (West Virginia University, USA)

Technical skill is mastery of complexity while creativity is mastery of simplicity.

- Christopher Zeeman

Contents

Chapter 1

Introduction

1.1 Biometrics

In a modern world, there are more and more occasions in which our identity must be reliably proved. For example, bank transaction, airport check-in, gateway access, computer login, etc., all such applications are related to privacy or security. But what is our identity? Most often it is a password, a passport, or a social security number. The link between such measures and a person, however, can be weak, as they are constantly under the risk of being lost, stolen, or forged. When the consequence of impostor attack becomes increasingly disastrous, the safety of the traditional identification approaches is brought under question.

Biometrics, the unique biological or behavioral characteristics of a person, is one of the most popular and promising alternatives to solve the secure identification problem. Typical examples are face, fingerprint, iris, speech, and signature recognition. From the user point of view, biometrics is convenient as people always carry it with them, and reliable as it is virtually the only form of authentication that ensures the physical presence of the user. For these reasons, biometrics has been an active research topic for decades. For an detailed review, see [74], [133]. This thesis, again, focuses on the interesting topic of biometrics, using it as the security solution for a specific application, and exploring interrelated research areas, like computer vision, image processing, pattern classification, that are relevant within this context.

1.2 Background

This work is carried out under the larger context of Freeband project PNP2008 (Personal Network Pilot) of the Netherlands [53], which aims to develop a user centric ambient communication environment. Personal Networks (PN) is a new concept based on the following trends:

- People possess more and more electronic devices that have networking functionality, enabling the device to share content, data, applications, and resources with other devices, and to communicate with the rest of the world.

- In the various living and working domains of the user (home, car, office, workplace, etcetera), clusters of networked devices (private networks) appear.

- When people are on the move, they carry an increasing number of electronic devices that communicate using the public mobile network. As such devices in the users personal operating space become capable of connecting to each other, they form a Personal Area Network (PAN).

A personal network is envisaged as the next step in achieving unlimited communication between people's electronic devices. It comprises the technology needed to interconnect the various private networks of a single user seamlessly, at any time and at any place, even if the user is highly mobile. An illustration of the PN is shown in Fig. 1.1.

Containing a lot of personal information, the PN puts forward high security requirements. The mobile personal device (MPD), which links the user and the network in mobile situations, must be equipped with a reliable and at the same time user-friendly user authentication system. This work, therefore, concentrates on establishing a secure connection between the user and the network, via biometric authentication on a MPD in the personal network.

1.3 Requirements

The requirements of biometric authentication for the PNP application can be categorized in three important aspects: security, convenience, and complexity.

1. *Security*

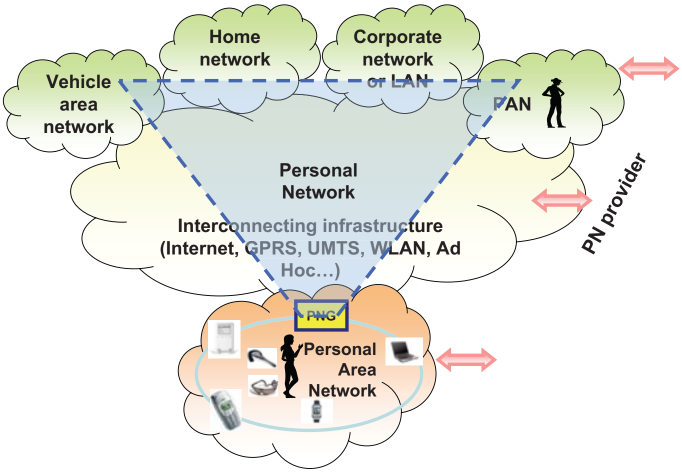


Figure 1.1: The personal network (PN) [53].

Security is the primary reason of introducing biometric authentication into the PN. There are two types of authentication in the MPD scenarios: authentication at logon time and at run time. Compared to the conventional logon time authentication, the run time authentication is equally important because it can prevent unauthorized users from taking an MPD in operation and accessing confidential user information from the PN.

To quantify the biometric authentication performance with respect to security, the false acceptance rate (FAR) is used. The FAR is the measure of security, specifying the probability that an imposter can use the device. The FAR of a traditional PIN (personal identification number) method is 10^{-n}, where n is the number of digits in PIN. At logon time, biometric authentication can be combined with a PIN to further reduce the FAR. At run time, it is not practical to use a PIN any more, and the biometric authentication system should have a sufficiently low FAR itself.

2. *Convenience*

The false rejection rate (FRR), which specifies the probability that the authentic user is rejected, is closely related to user convenience. A false rejection will force the user to re-enter biometric data, which may cause considerable annoyance. This leads to the requirement of a low FRR of the biometric authentication system.

Furthermore, in terms of convenience, a much higher degree of user-friendliness can be achieved if the biometric authentication is *transparent*, which means that the authentication can be done without explicit user actions. Transparency should be also considered as a prerequisite for the authentication at run time, because regularly requiring a user who may be concentrating on a task to present biometric data is neither practical nor convenient.

3. *Complexity*

   Generally speaking, a mobile device has limited resources of computation. The biometric authentication on the MPD, therefore, must have low complexity with respect to both hardware and software. When the authentication has to be ongoing, the requirements becomes even more strict due to the constantly ongoing computation.

   Because the MPD operates in the PN, it offers the possibility that biometric templates be stored in a central database and that the authentication is done in the network. Although the constraints on the algorithmic complexity become much less stringent, the option brings a higher security risk. Firstly, when biometric data has to be transmitted over the network it is vulnerable to eavesdropping [13]. Secondly, the biometric templates need to be stored in a database and are vulnerable to attacks [98]. These are problems difficult to solve. Conceptually, it is also preferable to make the MPD authentication more independent of other parts of the PN. Therefore, it is still required that the biometric authentication be done locally on the MPD. More specifically, the hardware (i.e. biometric sensor) should be inexpensive, and the software (i.e. algorithm) should have low computational complexity.

1.4 Why Face?

When considering the appropriate biometric for the PN application, we must bear in mind the requirements specific for the mobile device. To do this, eight popular biometrics are investigated, namely, fingerprint, hand geometry, speech, signature, gait, 2D face, 3D face, as shown in Fig. 1.2. The applicability of the biometrics are assessed under three explicit criterions, closely related to the three requirements in Section 1.3: *accuracy* which is related to security, *transparency* which is related to convenience, and *cost* which is related to complexity.

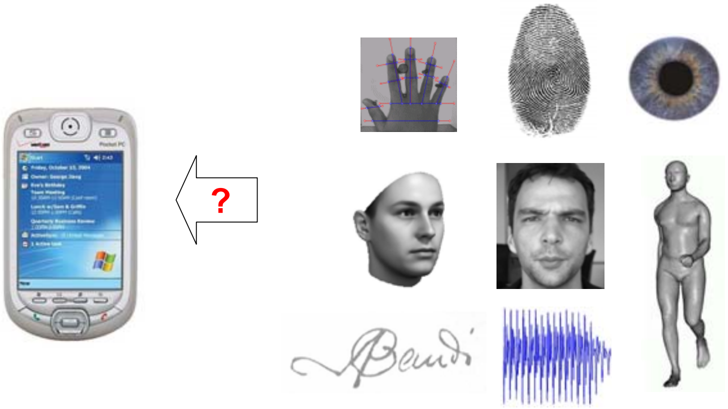


Figure 1.2: Left - a mobile device; right - popular biometrics in use: hand geometry, fingerprint, iris, 3D face, 2D face, gait, signature, and speech.

Fingerprint is one of the oldest and most popular biometric modalities [116]. The accuracy of fingerprint recognition is acceptable: as reported in [115], state-of-art fingerprint recognition systems can achieve an equal error rate (EER) of 2.2% at rather harsh testing conditions, and much better results under ideal circumstances.. Transparency can be realized, given that the user's fingerprint can be sensed at any time and anywhere. This, however, leads to very high hardware cost, as the fingerprint sensor should then cover nearly the entire surface of the mobile device. This not only makes the device expensive, but also renders the device physically vulnerable. Besides, wearing gloves or pressing the device with pen would easily cause failure.

Hand geometry recognition has similar problems. Although the accuracy of hand geometry is high, with an EER as low as 0.3% as lately reported [177], it is largely dependent on the hardware acquisition system. In conventional hand geometry systems [186] [177], a plane larger than hand is required to place the user hand on for scanning the whole rigid hand geometry. Additionally, pegs are installed on the plane to fix the positioning the hand. Such settings, unfortunately, are impossible to implement on a mobile device.

Iris is another important biometrics well-know for its uniqueness and accuracy [40]. A FRR of $1.1 - 1.4\%$ can be achieved at the FAR of 0.1% [126]. The difficulty of iris for the mobile device, however, lies in its high-cost hardware camera, which should be able to catch the high-resolution iris images. In a transparent manner, the requirement is intimidatingly high as the camera has to to track the iris in movement and at uncontrollable distances.

Speech and signature cannot be integrated to the mobile device for ongoing authentication, because the input of such biometric data is explicit and requires much user attention. Gait is not to be considered, as the gait of the user does not always exist (e.g. when the user is seated or standing still). Even when the gait exist, it is not easily detectable from the view of the mobile device. Besides, the accuracy of speech, signature, and gait as biometrics are relatively low as they are not sufficiently consistent, very often subject to change. For example, it is reported in a late evaluation that speech recognition only reaches a FRR of $5-10\%$ at the FAR of $2-5\%$ [130].

Face is the most classical biometric, as in daily life, it is used by everyone to recognize people. Face is also important in many practical cases of identification, such as the mugshot in police documentation, or the photo on a driver's licence and passport. For these reasons, automatic face recognition has been studied ever since computers emerged, and it remains a heated research topic until this day. Extensive reviews can be found in, for example, [24] [191]. There are two types of face recognition: two-dimensional face recognition using face texture images, and three-dimensional face recognition using face shapes and/or face textures. Generally speaking, the accuracy of face recognition is high. According to the latest face recognition vendor test FRVT 2006 [126], the state-of-art two-dimensional face recognition reaches a FRR of $0.8-1.6\%$ under controlled illuminations, and $10-13\%$ under uncontrolled illuminations, both at the FAR of 0.1%. For three-dimensional face recognition, illumination does not have an influence, and a FRR of $0.5-1.5\%$ is reported at the FAR of 0.1%. Transparency, furthermore, is an advantage of the face as a biometric. From the user point of view, no explicit action is needed for data acquisition. In the two-dimensional form, face data can be collected at low cost, with a low-end camera mounted on the mobile device. Besides, the biometric data collected with such cameras are small in size, potentially taking up little space and computational resources. Face in the three-dimensional form is not practical in contrast, as both hardware and software requirements are substantially increased.

Table 1.4 is a summary of the discussion, listing the applicability of the biometrics regarding accuracy, transparency, and cost. It is clear that face in the two-dimensional form is the most appropriate biometric in the PN context, offering high accuracy under controlled illumination, and moderate accuracy under unconstrained illumination, at low cost and in a transparent manner. This thesis, therefore, will concentrate on all the interesting aspects relevant to the two-dimensional face recognition problem.

biometrics	accuracy	transparency	cost
face (2D)	$-$	$\surd$	$\surd$
face (3D)	$\surd$	$\surd$	$\times$
fingerprint	$\surd$	$-$	$\times$
iris	$\surd$	$-$	$\times$
hand geometry	$\surd$	$-$	$\times$
speech	$-$	$\times$	$\surd$
signature	$-$	$\times$	$\surd$
gait	$-$	$\times$	$\surd$

Table 1.1: Applicability of different biometrics. $\surd$: good, $-$: moderate, $\times$: bad.

1.5 Fusion

Biometric fusion has been a popular research topic in recent years. This is based on the consideration that a single biometric is no longer sufficient for many secure applications [133]. Fusion is a way to combine the information from multiple biometric modalities, multiple classifiers, or multiple samples, in order to further improve the performance of the biometric system. In the PNP2008 project, the time sequences taken by the MPD can be seen as multiple information sources that can be fused to achieve higher performances. This strategy not only increases the system security level, in the sense that it avoids the device being taken away by impostors after the user logged on, but also essentially improves the system performance. Another context of our work is the European FP6 project 3D Face [1], which aims to use 3D facial shape data and the 2D texture data together for reliable passport identification in the future. In this context it is also important to study the way to effectively combine the information from the two distinct biometric modalities.

1.6 Outline of the Thesis

The outline of this thesis roughly follows the standard diagram of the face recognition system. From the raw image taken from the mobile device to the final decision of accept or reject, the data pass through such a processing line:

1. Face detection from the image;

2. Finer face registration from the detected face;

3. Illumination normalization to remove external influences;

4. Verification the processed face;

5. Information fusion to strengthen the final decision.

Chapter 2 deals with the first two steps, i.e., face detection and registration. The two steps are combined in one chapter, because we propose to do fast and robust face registration based on detected facial landmarks, which again turn out to be an object detection problem. The face and facial features share common properties as objects, in the sense they both possess large variability either intra-personally or inter-personally. The face detection is done by the Viola-Jones method, which is fast in detection because of its easily scalable features and the cascaded structure. For face registration, we trained 13 facial feature detectors by the specially tuned Viola-Jones method. Compared to face detection, a major problem in facial feature detection is the unavoidable falsely detections. For this purpose, we propose a very fast post-selection strategy, based on the error-occurring model, which is accurate and specific to the detection method as well as to the objects. The proposed post-selection strategy does not introduce any statistical model or iteration steps.

Chapter 3 studies the verification problem[1]. In this step, we proposed to use the likelihood ratio based classifier, which is statistically optimal in theory, and easy to implement in practice. On the mobile device, the enrolment can be done by taking a video sequence of several minutes. Above all, the method is chosen because the verification problem has a largely overlapping distribution of the classes, and therefore can be better solved by density-based methods than boundary-based methods. Furthermore, we have investigated the influence of various dimensionality reduction methods on the verification performance. Besides, we have also compared the single Gaussian model and the Gaussian mixture model.

Chapter 4 discusses the illumination normalization problem. An extensive review of the illumination normalization methodologies are done before the solution is given. We show that the three-dimensional modeling methods are not only complicated in computation, but also too delicate to generalize to the many scenarios we require. Instead, we propose two simple and efficient two-dimensional preprocessing methods: the Gaussian derivative filter in the horizontal direction and the simplified local binary pattern as a filter. The

[1] Note that we introduce the face verification prior to the illumination normalization because it is necessary to know the evaluation methods before studying the illumination problem.

two methods, especially the later, are computationally low-cost, and meanwhile exhibit a high degree of insensitivity to illumination variations.

Chapter 5 and Chapter 6 investigate the information fusion problem. In Chapter 5, we focus on the decision level, and propose the threshold-optimized decision-level fusion. In Chapter 6, we focus on the score level, and proposed an optimal LLR-based score-level fusion. A hybrid fusion scheme is also proposed based on the two proposed fusion methods. The common characteristics of the proposed fusion methods is that the receiver operation characteristics (ROC) of the component system is used as an intermediate in fusion, providing an easy and efficient way to study the problem from the operation points, without the need to tackle the more complicatedly distributed matching scores, as is normally done in biometric fusion.

In the end, Chapter7 presents the practical implementation of the the face verification system, and further sums up the thesis.

Chapter 2

Face Detection and Registration

2.1 Introduction

[1]Face detection is the initial step of face recognition. A general statement of the problem can be defined as follows: given an arbitrary image, determine whether or not there are any faces in the image and, if any, return the location and scale of each face [65] [185]. Although it is an easy visual task for human, it remains a complicated problem for computer, due to the fact that face is a dynamic object subject to a high degree of variability, originating from both external and internal influences. There has been extensive literature on automatic face detection, using various techniques with different methodologies, as will be reviewed in good detail. In our work, we have chosen to use the Viola-Jones face detection method [179], one of the most successful and well-known face detectors, with further adaptations for the mobile application.

Face registration, which aligns the detected face onto a standard scale and orientation, is an equally important step from the system point of view. Research has proved that accurate face registration has an essential influence on the subsequent face recognition performance [9] [131] [11] [10]. Basically, face registration re-localizes the face on a finer scale, which means that a more detailed study of the face content must be done. To achieve real-time performance

[1] This Chapter is based on the publication [10], [11].

and algorithmic simplicity, the proposed face registration is based on firstly detecting a limited number facial features called landmarks, and then from the detected landmarks calculating the registration transformation. This converts the face registration problem into a facial feature detection problem and a geometric transformation. This explains why we combine the face detection and registration into one chapter, as similar detection problems will be addressed. Successful face detectors, however, cannot be directly applied to facial features. More care must be taken due to the fact that facial features are harder objects to detect, insufficient by nature, with much fewer discriminative textures compared to the entire face. In this chapter, we propose easy solutions to customize the Viola-Jones detection method into efficient facial feature detectors, circumventing the intrinsic insufficiencies.

The remainder of this chapter is organized as follows. For the face detection problem, Section 2.2 reviews the existent face detection methods in two groups, and Section 2.3 introduces the Viola-Jones detector and adapts it into the MPD application. For the face registration problem, Section 2.5 reviews the face registration methods, and Section 2.6 presents our solution which satisfies the requirements of speed, accuracy, and simplicity. Section 4.7 summarizes this chapter.

2.2 Review of Face Detection Methods

Face and facial features are both hard objects to detect. Before going into detailed methods, it is interesting to first investigate the inherent difficulties in general for such a problem.

Basically, the difficulties of face and facial features detection lie in the following two aspects:

1. *Choice of Feature*
   Face and facial features are both highly flexible objects, with diverse appearances from different subjects, easily influenced by expression, pose, or illumination. A major difficulty of face or facial feature detection, therefore, lies in the way of selecting appropriate features to represent the object. The feature has to be representative of the object, and robust to the object variations.

2. *Choice of Classifier*
   Selecting features is only part of the work. Extracted features will be fed to certain classifiers. In most cases, the choice of features and the choice

of classifiers are mutually dependent. It is most desirable that simple features be combined with simple classifiers for robustness and simplicity, but in general such combinations cannot solve difficult detection problems. As a compromise, simple features incorporate a complex classifier, like in the Viola-Jones face detector [179], or complex features incorporate a simple classifier, like in the eigenface method [167]. In some work, complex features work together with complex classifiers, but the generalization of such a system will suffer, as too many trained parameters are involved.

It is difficult to partition the large variety of detection methods into widely-separate categories, as the influences of features and of classifiers are interweaved. Nevertheless, depending on the emphasis of the algorithms, we still group the face detection methods in two large categories: heuristic-based detection and classification-based detection. The former has more emphasis on the feature, while the later on the classifier. We do not intend to enumerate all the existent face detection methods in literature, instead, we are more interested in the methodologies underlying the methods, and their pros and cons.

2.2.1 Heuristic-Based Methods

Heuristics, the empirical knowledge of human face, are the first used clues for face detection. The heuristics which can direct the detection are normally very *general*, put into words like "the face region is of skin color", and "the eyes are above the nose". This trait makes the methods very simple and fast. On the other hand, however, due to the difficult nature of the face detection problem, methods using such simple rules tend to fail in difficult image situations, for example, when the skin tone changes under extraordinary illumination, or when the nose is concealed by shadow.

In this section, we will review heuristic-based face detection methods, transferring the human-recognized heuristics into computer-recognized rules. Two most commonly-used heuristics are reviewed: color and geometry.

Color

Skin color is representative of the face. It was found that human skin colors give rise to tight clusters in normalized color space, even when faces of different races are considered [71] [104]. Typical color spaces are RGB (red - green - blue) [71], HSI (hue - saturation - intensity) [95], YIQ (luma - chrominance) [34], YCbCr (luma - chroma blue - chorma red) [181], etc.

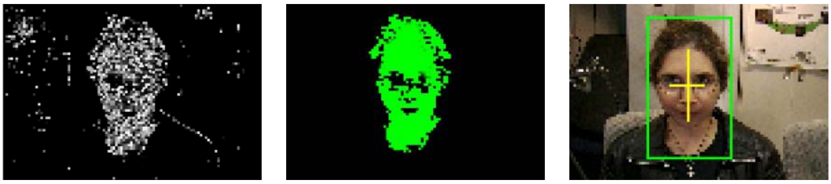


Figure 2.1: Per-pixel skin classification, blob growing, and detected face [118].

Color segmentation is performed by classifying each pixel value in the input image. Skin color can be modeled either in parametric or nonparametric manner. For example, histograms or charts are used in [22] [152], unimodal or multimodal Gaussian distributions are used in [127] [67]. The color models can be learned once for all, or in an online-updating manner [118].

For skin color classification per pixel, an optimal classification criterion is the likelihood ratio, expressed by

$$\frac{p(x|\omega)}{p(x|\bar{\omega})} > t \tag{2.1}$$

where x is the color vector of a certain pixal, $p(x|\omega)$ denotes the possibility that x belongs to the skin-color class ω, and $p(x|\bar{\omega})$ denotes the possibility otherwise. t is a threshold of the likelihood ratio.

By scanning the input image and applying pixel classification, a skin map is generated. In the next step, the skin pixels are grouped together using blob growing techniques to determine the face region [118] [67]. Fig. 2.1 shows an example of skin color based face detection.

Face detection by skin color is among the most simple and direct methods. The low complexity enables swift and accurate face detection in well-conditioned images. The drawback of the method, however, is its relative sensitivity to lighting conditions and camera characteristics, as well as the possibility to cause false acceptances in clustered backgrounds. Moreover, gray images cannot be processed due to the lack of color space information.

Geometry

Face geometry, the face shapes or the facial features layout, is useful heuristics for face detection. To detect such geometry, it is natural to first find out edges and lines which are representative of the geometry. In most geometry-based

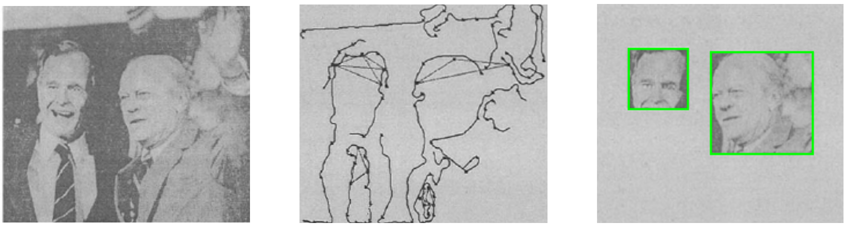


Figure 2.2: Example of edge-based face detection: original image, grouped edges, and detected faces [58].

work, therefore, edges or structural lines are used as features. They are firstly extracted from the input image, and then combined to determine whether a face exists based on the certain geometrical constraints.

In [138], structural lines in the input image are extracted using the greatest gradient method, and then compared to the fixed sub-templates of eyes, nose, mouth, and face contour. Edges in the input image can be detected by the Sobel filter [29], Marr-Hildreth edge operator [58], or derivatives of Gaussian [62], and then grouped together to search for a face. More recent methods include edge orientation map (EOM) and edge intensity map (EIM) [54], which uses edge intensity and edge orientation as features, and at the same time incorporates a fast hierarchial searching mechanism.

Basically, the geometry-based methods first compute features by scanning the entire input image with edge/line operators, then analyze the outcome image by grouping the resultant features. The existence of a possible face is finally determined by the combined evidences. This methodology is very similar to the color-based face detection methods. Fig. 2.2 shows an example of edge-based face detection [58]. In this work, edge contours are used as the basic features. Edges located by the Marr-Hildreth detector are filtered and cleaned to obtain contours. The contours are labeled as left, right and head curves according to their shapes, and then connected in groups. An edge cost function is defined to evaluate which of the groups represents a possible face candidate. Note how close the procedures actually are to [118] in Fig. 2.1. We point this out because in the following, a completely different face detection methodology will be introduced.

Geometry-based methods translate the obvious knowledge of face geometry into face detection rules. It is as simple and direct as the color-based methods. However, the features used by the methods are relatively sensitive to illumina-

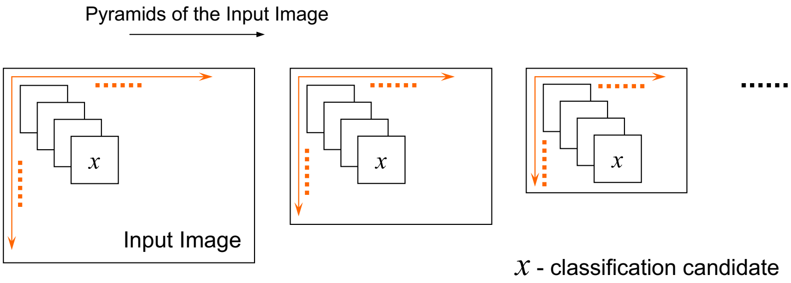


Figure 2.3: Candidates in classification-based methods, where x is the basic classification unit.

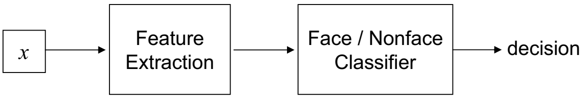


Figure 2.4: Classification of every candidate.

tion changes and noises. Consequently, the face detection methods based on grouping such features inevitably suffer from this susceptibility and cannot perform very well in case of poor illumination conditions and clustered background.

2.2.2 Classification-Based Methods

Generally speaking, heuristic-based methods are not reliable enough under difficult image conditions due to their simplicity. There is still a need for face detection methods that can perform in more or less hostile scenarios, like poor illuminations and clustered background. This has inspired abundant research work on a new methodology, which treats face detection as a pattern classification problem. Benefiting from the huge pattern classification resources, classification-based methods are able to deal with much more complex scenarios than heuristic-based methods.

Classification-based methods transfer the face detection problem into a standard two-class classification problem. Two explicit classes are defined: face class and non-face class. Before discussing the classifiers, we first explain how the input patterns of the classifier are obtained.

As no prior information is known about object location or size, the detection process must go through an exhaustive combination of positions and scales. Fig. 2.3 illustrates the searching strategy, in which every x is a fixed-sized candidate for the classifier, as shown in Fig. 2.4. In this way, a detection problem is transferred into a classification problem. As indicated in Fig. 2.4, the classifier input can be the pre-processed image patch x (like low-pass filtering, histogram equalization), or specially extracted image features (like Gabor features, Haar-like features). Obviously, the computation involved in such a process is very high. For example, in an input image of small size 100×100, the search with a template size of 10×10 and a scaling factor of 1.2 will result in $61,686$ candidates, which implies potentially $61,686$ times feature extraction and $61,686$ times pattern recognition. This puts forward high demands on the designing of the features and classifiers, or sometimes the co-design of them.

In the following session, we will discuss the classification-based face detection methods in two categories depending on the characteristic of the classifiers used, namely, linear methods and nonlinear methods.

Linear Methods

Images of human face lie in a subspace of overall image space. Linear methods construct a linear classifier, assuming a that a linear separation boundary solves the classification problem. In this section the two most important linear methods, principal component analysis (PCA) and linear discriminant analysis (LDA), are reviewed. These two methods embody the key idea of linear methods, namely, reducing the subspace dimensionality (hence complexity) based on optimization of certain criterions through linear transformations. Linear classification methods are simple and clear from the mathematical point of view. Moreover, they can be extended to the nonlinear space by introducing nonlinear kernels.

PCA was firstly used by Sirovich and Kirby for face representation [150], and by Turk and Pentland for face recognition [167]. Given a set of N faces, denoted by $x_1, ..., x_N$, which are vectorized representations of the two-dimensional image. The covariance matrix Σ is computed by

$$\Sigma = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^{\mathrm{T}} \tag{2.2}$$

where $\mu = \frac{1}{N-1} \sum_{i=1}^{N} x_i$ is the mean face vector.

The criterion for PCA is maximal preservation of the distributional energy after the linear projection, as expressed by

$$U_{\mathrm{PCA}} = \arg\max_{U} |U^{\mathrm{T}} \Sigma U| = [u_i, ..., u_k] \tag{2.3}$$

where k is the reduced dimensionality, and U is the orthogonal matrix satisfying $U^{\mathrm{T}}U = I$. This is a eigenvalue problem

$$Su_i = \lambda_i u_i, \qquad i = 1, ..., k \tag{2.4}$$

which can be solved by eigenvalue decomposition of Σ, or singular value decomposition (SVD) the data matrix X that contains the sample x_i as columns.

LDA is a supervised dimensionality reduction approach, seeking to find a projection matrix which maximally discriminates different classes [52]. Generally speaking, LDA is intended for a multi-class problem, but it can also be applied to the two-class face detection problem when the class of face and non-face are clustered into subclasses [183] [154]. This allows more complicated modeling of the face space. Let the between-class scatter be defined as

$$S_{\mathrm{b}} = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^{\mathrm{T}} \tag{2.5}$$

and the within-class scatter be defined as

$$S_{\mathrm{w}} = \sum_{i=1}^{c} \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^{\mathrm{T}} \tag{2.6}$$

where μ_i is the mean of class ω_i, μ is the total mean, N_i is the number of samples in class ω_i, and c is the number of classes. LDA aims to find the projection matrix U which maximizes the ratio of the determinant of the projected between-class scatter and within-class scatter

$$U_{\mathrm{LDA}} = \arg\max_{U} \frac{|U^{\mathrm{T}} S_{\mathrm{b}} U|}{|U^{\mathrm{T}} S_{\mathrm{w}} U|} = [u_i, ..., u_k] \tag{2.7}$$

This is a generalized eigenvalue problem

$$S_{\mathrm{b}} u_i = \lambda_i S_{\mathrm{w}} u_i, \qquad i = 1, ..., k \tag{2.8}$$

which can be solved by simultaneous diagonalization of S_{b} and S_{w} [55].

Linear transformations U simplifies the original high-dimensional space, making it more tractable under specific criterions. Classification can be done in the

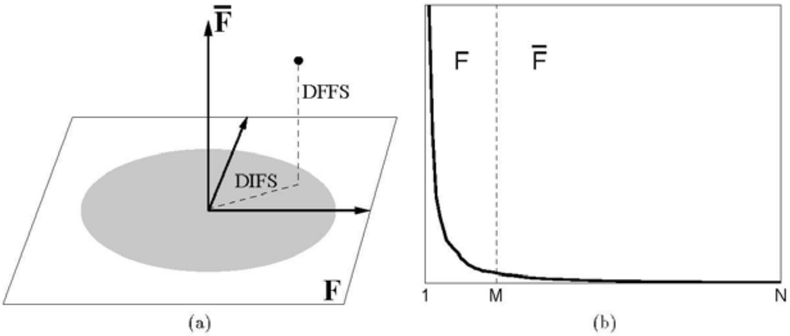


Figure 2.5: (a) Decomposition of the face space to a principal subspace F and a complementary subspace $\bar{F}$ by PCA. (b) A typical eigenvalue spectrum and its division into the two spaces [108].

reduced space in various ways. Turk and Pentland defined a preliminary measure of "faceness" [167], which is the residual error termed by DFFS (distance from face space), indicating how far an input image patch is from the face space. The Mahalanobis distances [44] in the reduced face, DIFS, can also be used as a measure of likelihood that an input vector x belongs to the face class. Both DIFS and DFFS can be calculated by linear manipulations of the vector x, and illustrations are given in Fig. 2.5.

$$\mathrm{DIFS}(x_i) = \big((x_i - \bar{x})^{\mathrm{T}}\Sigma^{-1}(x_i - \bar{x})\big)^{\frac{1}{2}} \tag{2.9}$$

$$\mathrm{DFFS}(x_i) = (I - U^{\mathrm{T}}U)(x_i - \bar{x}) \tag{2.10}$$

Linear methods derive the final quantitative measure of "faceness" by linear transformations of the input pattern x. In the work of Moghaddam and Pentland [110], the measures are further related to the class conditional probabilities under certain simplified assumptions, and statistically optimal classification can be achieved in this respect. We will revisit the linear classification problem in Section 3 for face recognition.

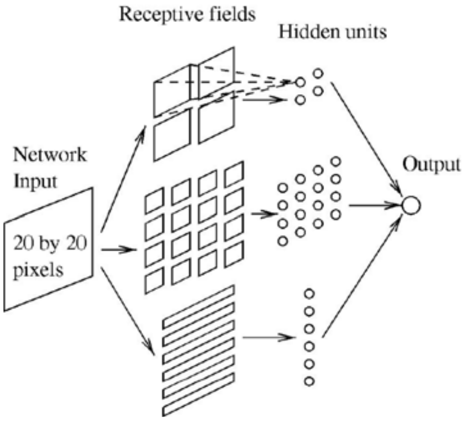


Figure 2.6: Neural network structure used in [135] for face detection.

Nonlinear Methods

Due to the high variability and complexity of the face images, linear models are often not adequate to achieve very high robustness and accuracy for the face detection problem. Nonlinear classification methods, which accommodate more complicated class distributions, have been intensively investigated in this respect [65] [185]. In this section, we will mainly review three of the most renowned and interesting nonlinear classification methods, namely, neural network, support vector machine, and Adaboost.

Neural networks have long been a popular technique for many complicated classification problems [12]. Basically, a neural network contains a number of interconnected nodes, i.e. neurons, resembling human brain structures. The interconnections of these neurons are learned from a set of training samples. In the application of face detection, the network is trained as a discriminant function between the face class and non-face class. Examples are Multi Layer Perceptron (MLP) [81] [135], probabilistic decision-based neural network (PDBNN) [97], sparse network of winnows (SNoW) [134], etc. A representative work is that of Rowley et al. [135], in which a system is proposed incorporating face knowledge in a retinally connected neural network, as shown in Fig. 2.6 [135]. The basic classification unit are of the size 20×20, sampled from the input image in the way described in Fig. 2.3. In the neural network structure, there is a hidden layer with 26 neurons, where 4 of them look at the 10×10 subregion,

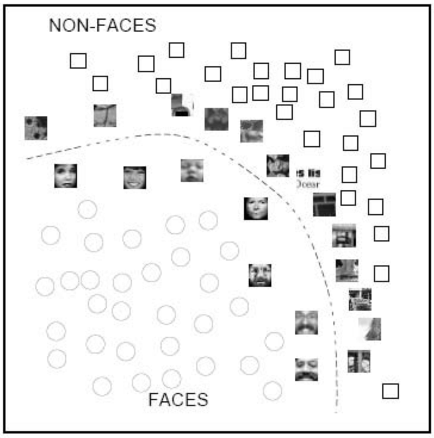


Figure 2.7: SVM-based face detection in [120], support vectors and the decision boundary.

16 look at 5 × 5 subregion, and the rest 6 look at 20 × 5 overlapping horizontal stripes. The network is trained by the back propagation (BP) algorithm [44], using a large set of face and non-face samples. For more reliable performance, multiple neural network of the same structure are trained with different initial weights and different sample sets, and the final decision is based on arbitration of all these networks. The arbitration of multiple classifiers is an important part of the thesis, and we will come to it later in Chapters 5 and 6.

Another interesting point in Rowley et al.'s method is that *bootstrapping* [44] is adopted in training. This is due to the fact that the non-face class is extremely extensive, which is impossible to be covered by limited available samples. Instead of running the training exhaustively on all possible non-face patterns, the idea is to concentrate on the "difficult" non-face patterns which lie close to the boundary of the two classes. The strategy, therefore, is simply to re-train those non-face samples which are misclassified by the previous iterations, thus putting more emphasis on those patterns difficult to classify. Similar ideas will be revisited in the Adaboost approach.

One of the disadvantages of the neural network approach is its high computational complexity, which makes real time face detection difficult. Besides, it is often susceptible to overtraining due to its high flexibility.

Support Vector Machine (SVM) is another important classification technique

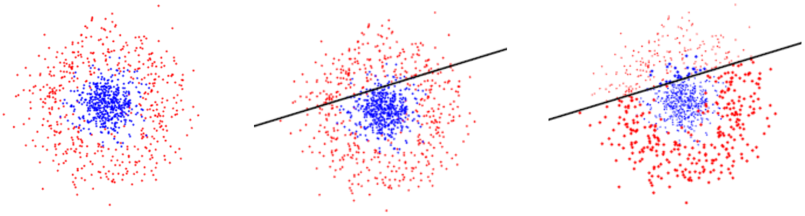


Figure 2.8: The training procedure of the Adaboost classifiers. Left: the original classification problem, samples equally weighted; middle: the first weak linear classifier; right: reweighting of the training samples, where misclassified samples are given higher weights, which are indicated by the dot size.

that can generate rather complicated decision boundaries [30]. The idea behind SVM is that when the original feature vectors are mapped into a higher dimensional (sometimes infinite) space, a simple linear classifier can be expected to achieve good classification performance. In the nonlinearly mapped feature space, SVM constructs a *maximal margin* linear classifier [44], which, back in the original feature space, turns out to be a nonlinear classifier. The so-called ”kernel trick” makes this mapping of space simple, by introducing the nonlinear inner product kernels [148].

SVM has been widely used in the face detection problem [120] [80]. In the work of Osuna et al. [120], the image window of size 19 × 19 is used as the basic classification unit. A second order polynomial kernel is adopted in the SVM formulation. As shown in Fig. 2.7, the faces along the boundary are the ”support vectors” in the two opposite classes. It is easy to see that they represent difficult samples in either class.

The advantage of SVM is its generalization ability. Compared to the neural networks, in which each sample in the specific training set often has an influence on the final network weights, SVM only counts those critical samples, i.e. support vectors, which are most important for classification. The disadvantage of SVM, however, is its high computation load, with respect to both CPU and memory, to solve the quadratic optimization problem. This drawback becomes especially serious when the training set is large.

The third classification method we are going to review is the Adaboost algorithm. In general, *boosting* means an iterative process, which accumulates a number of component classifiers to form an ensemble, whose joint classification performance is higher than any of the component classifier. Adaboost, short for "adaptive boosting", is characterized by the adaptive weight associated with each training pattern. The weight is adapted in such a way that difficult patterns receive higher weights, meaning that they will be given more emphasis during the next iteration. Fig. 2.8 illustrates the Adaboost training process, in which the misclassified samples are given higher weights to train the next component classifier.

There are several desirable properties of the Adaboost classifier. Firstly, it uses simple weak component classifiers, which may only perform slightly better than chance [44], like the simple linear classifier in Fig. 2.8. Secondly, Adaboost can reduce the training error to an arbitrarily low level when the number of weak classifiers is sufficiently large. This is similar to the neural network, but as a third point, Adaboost has much better generalization capabilities [44] [143] compared to neural network. The key insight is that generalization performance is related to the margin of the samples, and that Adaboost rapidly achieves a large margin [179]. Adaboost classifiers have been successfully applied to the face detection problem [179] [153] [145]. In the next section, we will introduce the famous Viola-Jones method, which uses the Adaboost classifier and realizes real-time robust face detection.

2.3 The Viola-Jones Face Detector

The Viola-Jones face detector is one of the most well-known face detection methods in literature. There are three characteristics in this method: Haar-like features that can be rapidly calculated across all scales, Adaboost training to select features, and cascaded classifier structure to speed up the detection.

2.3.1 The Haar-Like Features

Simple Haar-like rectangular features are used in the Viola-Jones face detectors, as shown in Fig. 2.9. The features are calculated as the sum of the pixel values in white rectangles subtracted by the sum of pixel values in the gray rectangles. In [179], three feature different structures are used: two-rectangle, three-rectangle, and four-rectangle. Consequently, features with different structures, different sizes, at different locations relative to the enclosing window (with the size of a basic classification unit x as shown in Fig. 2.3), construct a very large pool of features. For example, when the basic classification unit has a size of 24×24, the exhaustive set of features is about 160,000. This is a over-complete feature

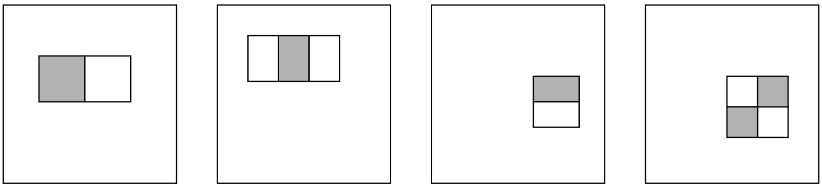


Figure 2.9: Example rectangular features shown relative to the enclosing window [179].

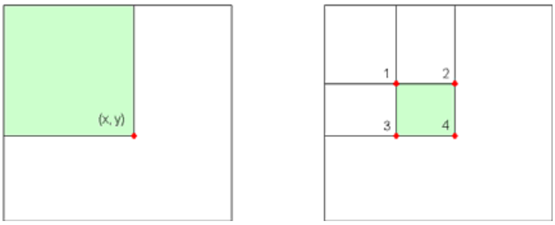


Figure 2.10: Integral image I. Left: the value $I(x, y)$ of the integral image at point (x, y) is the sum of all pixel values in the marked rectangle. Right: the sum of the pixel values within the marked rectangle is simply $I(x_4, y_4) + I(x_1, y_1) - I(x_2, y_2) - I(x_3, y_3)$.

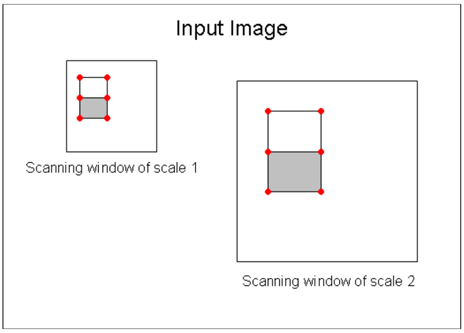


Figure 2.11: Scanning the input image at different scales. It can be seen that at either scale, the calculation of the feature only involves additions and subtractions of 6 values in the integral image.

set.

By introducing an intermediate *integral image*, the pixel values within any rectangle can be easily calculated using only 4 values from the integral image, as clearly illustrated by Fig. 2.10. As discussed in Section 2.3.2, one of the biggest obstacles for real-time detection is the exhaustive scanning at all possible scale and location on an input image. To obtain the image pyramids in Fig. 2.3, in the first place, is very time consuming. Integral image solves this problem by avoiding the image pyramids. As shown in Fig. 2.11, the exhaustive search in the image pyramids is transformed into the scanning of a single input image with windows of different scales, with a certain step like 1.1 or 1.2. Fig. 2.11 shows one specific Haar-like features at two different scales. It can be observed that at either scale, the calculation of this feature is only related to the integral value of the 6 points marked (interpolations can be used when the coordinates of the points are fractional). This implies that feature values at any scale can be easily obtained by calculating the integral image only once. As pointed out in [179], any procedure that requires pyramid calculation will necessarily run slower.

2.3.2 Adaboost Training

It is easy to see from Fig. 2.9 that the Haar-like features are representative of some simple image textures like edges and bars of different orientations. As mentioned in the previous section, the total number of such Haar-like features is huge. Adaboost training aims to select from the huge feature pool a combination of features which can well discriminate the face and the nonface patches.

The weak classifier in the Adaboost training is simply a decision "stump" [179], which consists of a feature f, i.e., the specific Haar-like feature as shown in Fig. 2.9, a threshold θ, and a polarity p

$$h(x, f, p, \theta) = \begin{cases} 1 & \text{if } pf(x) < p\theta \\ 0 & \text{otherwise} \end{cases}$$

where x is the basic classification unit as illustrated in Fig. 2.3, of the size 24×24. The Adaboost training algorithm in the work of Viola and Jones is formally described as in Algorithm 1.

Algorithm 1 The Adaboost training algorithm.

```
Require: The sample images $(x_1, y_1), ..., (x_n, y_n)$ where $y_i = 0, 1$ for negative
  and positive samples, respectively.
Ensure: The strong classifier constituted by a number of selected weak classifiers.
  for $t = 1, ..., T$ do
    Normalize the weights, $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{i=1}^{n}} w_{t,j}$;
    Select the best weak classifier with respect to the weighted error $\epsilon = \min_{f,p,\theta} \sum_i \omega_i |h(x_i, f, p, \theta) - y_i|$.
    Define $h_t(x) = h(x, f_t, p_t, \theta_t$ where $f_t$, $p_t$, and $\theta_t$ are the minimizers of $\epsilon_t$;
    Update the weights: $w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$, where $e_i = 0$ if sample $x_i$ is
    classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon}{1-\epsilon}$.
  end for
  The final strong classifier is:
    $C(x) = \begin{cases} 1 & \sum_{t=1}^{\mathrm{T}} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{\mathrm{T}} \alpha_t \\ 0 & \text{otherwise} \end{cases}$
  where $\alpha_t = \log \frac{1}{\beta_t}$.
```

2.3.3 Cascaded Classifier Structure

The Adaboost training selects a combination of weak classifiers, and the final ensemble classifier can be used to classify every basic classification unit. As

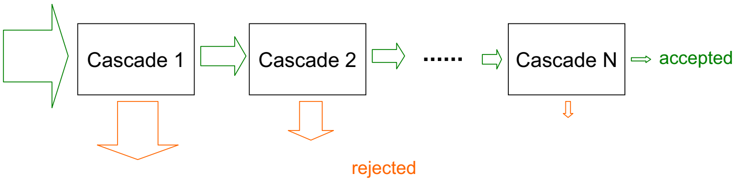


Figure 2.12: Cascaded classifier structure.

analyzed in Fig. 2.3, however, the number of all possible classification units, even in a small sized input image, are vastly huge. To go through all of them with all the selected weak classifiers is forbiddingly complex. Based on the fact that most of the basic classification units are negative, cascaded classifier structure as shown in Fig. 2.12 radically reduces the computation time while improving detection accuracy.

The classifier is designed in such a way that the initial cascade is able to eliminate a large percentage of negative candidates with very little processing. Subsequent cascades eliminates additional negatives but requires some additional computation. After several cascades the number of classification units have been reduced dramatically, and the later cascades only focus on very promising candidates.

In practice the cascaded structure is realized by successive Adaboost learning. Each stage is trained using the scheme described in the previous section. For a single cascade, weak classifiers are accumulated until certain performance criterion (e.g. FAR or FRR) is met for this stage. Then similar training is continued to select another set of weak classifiers to form the next cascade with the performance criterion. According to the Adaboost rule, the training is consecutively focusing on more difficult samples, therefore, given the same performance criterion, the later cascades will contain an increasing number of weak classifiers.

In the detection process, most of the classification units are rejected after being processed rapidly by the earlier cascades containing fewer weak classifiers. This makes the algorithm extremely efficient for the detection problem as described in Fig. 2.3.

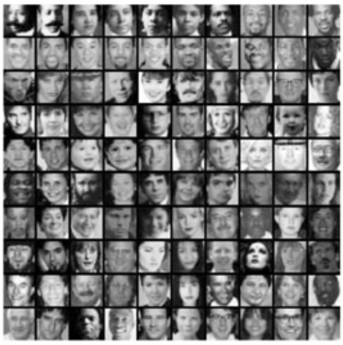


Figure 2.13: The face training set of the size 24 × 24, from [179].

2.4 The Viola-Jones Face Detector Adapted to the MPD

We have adopted the face detector trained in [179]. The face samples are consisted of 4,916 roughly aligned human faces scaled to the base size of 24. Samples of the training face images x are shown in Fig. 2.13 to give some idea of the large variability in the face class. Non-face training samples are randomly chosen from images that do not contain human face.

Although the Viola-Jones face detector proves to be fast and robust for face detection in general, it can be further improved for the application of face detection on the MPD in particular.

The specificity of the face images in the MPD application is related to the distribution of face sizes in the normal self-taken photos from a hand-held device. This information provides useful constraints on the searching and significantly speeds up the implementation. On the left of Fig. 2.14, some typical face images taken from ordinary hand-held PDA (Eten M600) are shown.

Suppose with a very high probability $1 - \epsilon$, $\epsilon \sim 0$, the detected face size s lies in a scope between $s_{\min}$ and $s_{\max}$, i.e. $p(s|s_{\min} \leq s \leq s_{\max}) = 1 - \epsilon$, where p is the probability distribution function of s. Then there are two steps to reduce the computational efforts for face detection:

1. Down-scale the original image first before detection. The down-scaling factor, for example, can be set around $\frac{s_{\min}}{s_{\text{face}}}$, where s_{face} is the minimal detectable scale [179]. In the trained detectors, $s_{\text{face}} = 24$.

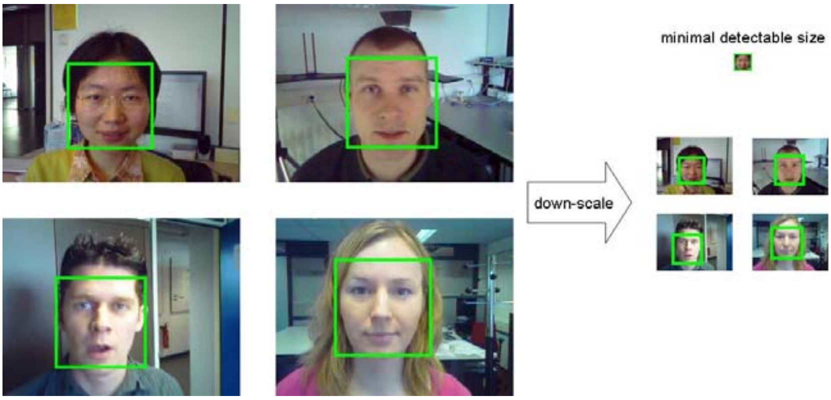


Figure 2.14: Left: typical face images taken from ordinary hand-held PDA (Eten M600), with the size 320 × 240. Right: down-scaled face images with the size 100 × 75. Face detection results are shown in both cases.

2. In the reduced image, restrict the scanning window (as shown in Fig. 2.11) to be from the minimal size 24 to the maximal size $24\frac{s_{\max}}{s_{\min}}$.

Referring to Fig. 2.3, it can be easily seen that the number of candidates for classification increases exponentially with the size of the input image. The first step, therefore, radically reduces the number of possible classification units. Despite the fact that the Viola-Jones face detector has good scaling property and efficient cascaded structure, this rescaling strategy is still very useful to speed up the detection. In addition, the second step spares the unnecessary search for faces of too small or too large sizes. This furthermore reduces the number of classification units to a large extent.

Fig. 2.14 shows detection results both in the original and in the reduced image. We observed in the experiments that in the latter, equally good results are obtained with far less effort. In practice, as the minimal detectable size is small enough, 24 × 24 (also shown in Fig. 2.14), the original image can always be down-scaled as long as the face in it is no smaller than this size. As a result, considerable calculation time is saved for the MPD application. One possible drawback of down-scaling, however, is that the detected face scale could be somewhat coarser than that in the original image, since fewer scales have been processed. This nevertheless hardly affects the final face recognition, as registration of the detected faces on a finer scale will follow.

2.5 Review of Face Registration Methods

Although an satisfactory solution has been found for face detection problem on an MPD, location of the detected face is not precise enough for further analysis of the face content. This is due to the fact that considerable variations containing poses, illuminations, and expressions, as shown in Fig. 2.13, are necessary in the face training set to achieve a robust face detector. This inevitably leads to imprecision of face localization, as the training faces themselves are not strictly aligned.

For the subsequent face recognition task, face registration must be done first to align the detected face onto a finer scale, i.e., to a standard orientation, position, and resolution. It has been emphasized in literature that high quality face registration is very important for the face recognition performance [9] [131]. The problem of face registration, however, is to some extent overlooked in academic face recognition research, as many databases used in the experimental evaluation, such as the BioID database [171], FERET database [172], FRGC database [173], have manually labeled landmarks available for the registration purposes. Using these manual labels for registration in face recognition experiments leads to optimistic performance estimates, as in reality, these labels are not available and one has to rely on automatic landmark detection, which may be less accurate.

We categorize the automatic face registration methods into three groups: *holistic* methods, *local* methods, and *hybrid* methods, depending on the methodologies used to look at the face content.

2.5.1 Holistic Face Registration Methods

In the holistic face registration methods, the face image is used as a whole, and the registration problem is converted into an optimization problem. Examples of the optimization criterion are correlation [142], mutual information [180] and matching score [15], as a function of the holistic face image content. The registration problem is formulated as finding the transformation parameter which best matches the input image to the template

$$\theta = \arg\max_{\theta} \{F(x, r, \theta)\} \tag{2.11}$$

where θ denotes the transformation parameters, including translation, rotation, and scaling, F is the criterion function, x is the holistic image (results of rough face detection), and r is the template. This equation is further illustrated by

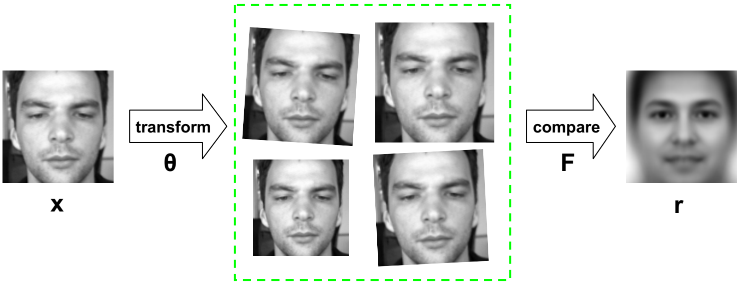


Figure 2.15: Holistic face registration methods. The input image is transformed to find the optimal match to the template.

Fig. 2.15. Note that x, the holistic image content is taken into consideration in calculating the matching criterion.

The advantage of this category of methods is that the registration can be robust with respect to global noise or illumination effects, and can work with low quality or low resolution images on which local analysis is not possible. The disadvantage of holistic methods, however, is its computational complexity, as the iterations in the optimization process involve every pixel value in the detected face image. The complexity of such an nonconvex optimization problem, arising from the local minima and the high-dimensional parameter space, also adversely influences the registration performance.

2.5.2 Local Face Registration Methods

In comparison, local methods only make use of a limited number of local facial landmarks to do face registration. Prominent facial features are detected, such as eye, nose, mouth, etc, and their coordinates are used to calculate the transformation parameter θ as in (2.11).

Various facial feature detection methods have been proposed in literature. On face images obtained under good conditions (e.g. frontal pose, uniform illumination), simple strategies can be used to locate the eyes and the mouth just by some heuristic knowledge, such as brightness of the facial features and symmetry of the face [137] and [68]. To deal with a larger range of face images, more complicated local facial landmark detectors are developed. In [20], multi-orientation, multi-scale Gaussian derivative filters are used to detect the local feature. Furthermore, the detection is coupled with a relative statistical model

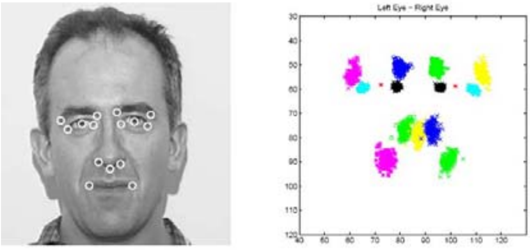


Figure 2.16: Local face registration methods. Left: located facial features, right: statistical map of the facial feature locations when the two eyes are used as the reference [20].

of the spatial arrangement of facial features to yield robust performance. Fig. 2.16 right shows the learnt relative statistical distribution of facial landmarks when the two eyes are used as the reference points. This work identifies two important aspects of local face registration methods: a robust detector, and a geometrical shape model. Similar ideas can be found in [31] and [32], in which the facial features are first detected by the Viola-Jones method, and then a geometrical model called pairwise reinforcement of feature response (PRFR), together with an active appearance model (AAM), are taken to further refine the results.

Another interesting work is [50], in which Gabor wavelet networks (GWN) are applied in a hierarchial way: the first-level GWN is used to match the face and estimate the approximate facial feature locations, and the second-level GWNs are individual facial feature detectors aiming to fine-tune the facial features locations. This method resembles the elastic bunch graph matching (EBGM) method [184], in the sense that in both methods, facial information is derived in a top-down manner.

It can be noticed that in all these local methods based on facial feature localization, geometrical shape information are incorporated either as an additional constraint [20] [32] or as a prior [50]. This implies the insufficiency of facial feature detectors in general, as observed by Burl et al., *facial feature detectors based on the local brightness information are simply not reliable enough* [20]. In Section 2.6, we the characteristics of facial feature detectors will be further investigated, and more insights will be given.

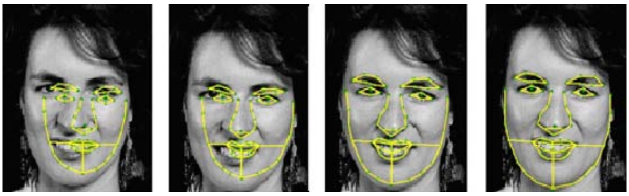


Figure 2.17: Active shape model (from left to right: initialization, iteration, more iteration, convergence.)

2.5.3 Hybrid Face Registration Methods

Hybrid face registration methods combine the holistic facial texture and the local facial landmark information. Well-known examples are the active shape models (ASM) [27] and active appearance models (AAM) [26] by Cootes et al.

In the ASM method, the shape, which is the combination of the marked feature points as shown in Fig. 2.17, is modeled in a PCA space. The eigenvectors and the corresponding eigenvalues describe and restrict this space. The texture information around the feature points is used to guide the fitting of those feature points onto the face image, by analyzing the profile vector [27] or the wavelet features [194] in the proximity of the feature points.

The fitting of ASM is basically an iterative optimization process, as shown in Fig. 2.17, which can be summarized very briefly in Algorithm 1.

Algorithm 2 The Active Shape Model Algorithm.

```
Require: An input face image and an initialization of the shape on the face.
Ensure: The registration of the shape to the face image.
  while The shape difference between the two consecutive rounds exceeds a
  predefined small value, do
    Update the shape: for each feature point on the shape, search in its
    neighborhood for a local best matching, based on the analysis of the
    local textures;
    Refine the shape: apply PCA model constraints to the shape obtained in
    the previous step.
  end while
```

Using a similar framework, AAM further incorporates texture analysis in addition to the shape. More specifically, a Delauney triangulation of the face

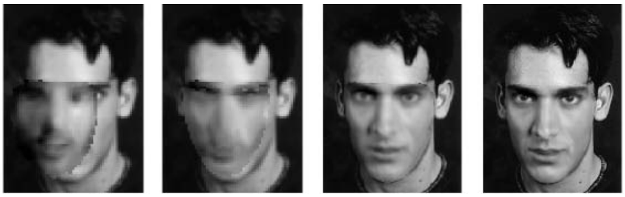


Figure 2.18: Active appearance model (from left to right: initialization, iteration, more iteration, convergence.)

image is first performed and then the enclosed texture region in the triangles are normalized to form the texture vector [26]. The updating is done by minimizing the difference between the current texture and the texture predicted by the model. Fig. 2.18 illustrates the iterative process of AAM fitting.

Both the ASM and the AAM use structural constraints to help locate the feature points and thus align the face. With the assistance of such shape or texture constraints, the requirements on the detectors used in the updating step (2) is much lower as compared to those in the local methods of Section 2.5.2. However, the hybrid methods have two drawbacks: first, initialization influences the convergence, or in other words, local minima may occur in the optimization and cause registration error; second, iterative steps takes time, especially in the case of AAM when much information is to be processed each time. The second drawback, especially, makes hybrid registration method unfavorable under our real-time application context.

2.6 Face Registration on MPD by Optimized VJ Detectors

To do face registration on the MPD, we have chosen for the local face registration method as described in Section 2.5.2, because of its directness, i.e., no iterative process is required as in the global or the hybrid methods. This potentially speeds up the registration. The challenge, however, lies in the designing of reliable facial feature detectors. From the previous analysis of local methods, it has been clear that this is a very difficult task, see the comments of Burl et al. at the end of Section 2.5.2.

We will stick to the Viola-Jones approach as described in Section 2.3, but tactically optimize it for the facial feature detection problem. The reason to choose the Viola-Jones method is its speed, accuracy, and robustness, which we wish to take advantage of again. In order to achieve equally satisfactory performance on facial features as on face, however, additional work must be done to cope with the inherent problems of facial features.

2.6.1 Problems of Facial Features as Objects

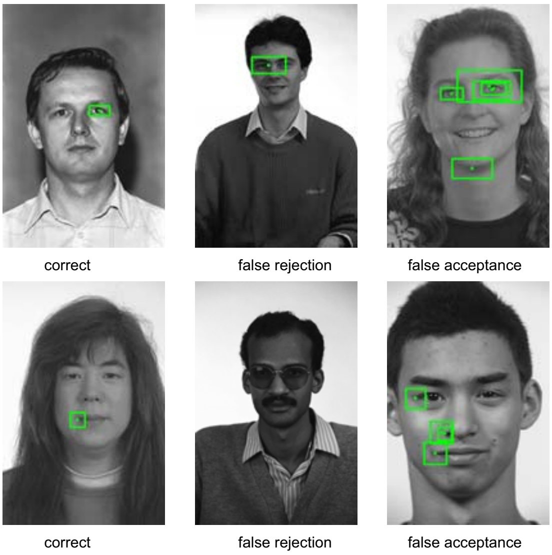
Facial features are difficult objects to detect. The reasons are twofold:

- Firstly, the structures of facial features are not constant enough, both intra- and extra-personally. For the same individual, differences in expressions and poses can alter the shape of facial features considerably. Consider the same face being happy and being sad. For different individuals, the variability of the facial feature are also large, e.g. big round eyes v.s. small narrow eyes. This will eventually lead to false rejections in the detection.

- Secondly, the structures of facial features do not contain enough discriminative information, or distinct local structures. In other words, chances are not small that the structure of a background patch coincides with that of a certain facial feature. For example, an eye basically has a white-black-white pattern, which a nostril also possesses. This will lead to false acceptances in the detection[2].

The two points listed above advances a controversy in the facial feature detection problem. If a detector is trained to be more or less specific, it easily misses many true objects that deviate from the training set. On the other hand, if the detector is trained somewhat looser, it tends to accept many false background patterns. From a statistical point of view, this implies that the facial-feature class and non-facial-feature class have large overlap in distribution (in the Haar-like feature space as is specific for the Viola-Jones method, and imaginably in other type of feature spaces for other detection methods), which leads to inherently high Bayesian classification error that cannot be reduced. Fig. 2.19 shows some examples of the facial feature detection results by directly applying the Viola-Jones method, where the dots denote the landmark center,

[2] The second point explains why facial feature detection is even more difficult than face detection. Face, although with large variation, does possess relatively distinct local structures, i.e., the specific layout of eyes, nose, mouth, etc, which a random image cannot easily resemble. Check Fig. 2.7 for some interesting false accepted faces, shown as the support vectors in the negative class. Those false acceptances are not likely to occur very often though.



correct   false rejection   false acceptance

correct   false rejection   false acceptance

Figure 2.19: Examples of the left eye (upper) and the right mouth corner (lower) detection results, using the FERET database [172]. Both false rejections and false acceptances are observed.

and the rectangles denote on which scale the landmarks are detected. The figure gives a clear view of the underlying risks: concurrent false rejections (miss) and false acceptances (multiple detections).

To find the facial features in the first place, a common compromise is that the detectors are tuned at an operation point with a low false rejection rate, and inevitably a high false acceptance rate. In other words, the facial features are detected at the cost of many false detections. This gives rise to a large number (exponentially related to the total number of facial features) of possible combinations of different facial features. To choose the best one out of them usually costs extra statistical shape models like in [32][31][20]. In our work, however, we try to get rid of these additional shape models, thus avoiding the trouble of learning such models, as well as the additional errors that may be introduced by them.

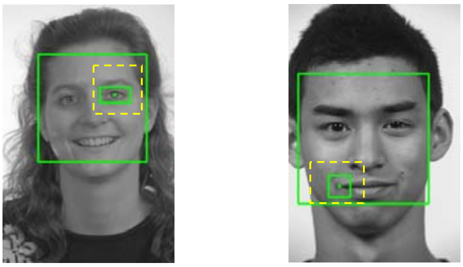


Figure 2.20: ROIs with respect to the detected face rectangle for the left eye and the right mouth corner.

In the remaining part of Section 2.6, we will present a series of solutions. These solutions are simple, but in combination they result in a fast, accurate, and robust facial feature detection system, which works under a large range of image resolutions and illumination conditions.

2.6.2 Constraining the Detection Problem

The facial feature detection problem can be re-defined as a *constrained* object detection problem. Unlike the case of face detection, where faces have sufficient local structures that a random patch in the background is not quite likely to coincide with, the facial features have relatively simple local structures that random patches in the same image could also possess. In order to reduce the chance of false acceptances, we define the range of facial feature detection to be only within a constrained region around the true features. In practice this can be done by first detecting the face, and then setting an approximate ROI (region of interest) with respect to the detected face rectangle, as shown in Fig. 2.20. The figure shows the effects of ROI on false detections, where the dashed rectangles indicates the ROI for the left eye and the right mouth corner. The false detections in Fig. 2.19 (a3) and (b3) are easily eliminated. By reducing the searching area, this also speeds up the detection considerably.

The constraint not only makes a difference in the detection process, but also in the training process. Under the new definition, the range of negative training samples is restricted to be only within the ROI of the facial features, instead of being arbitrary as in the original Viola-Jones work [179]. This makes the trained detectors more specific, discriminative, and accurate, as these negative candidates are most likely to occur during detection. A most discriminative combination of Haar-like features will be selected during the training stage to sharply locate the facial feature within its ROI during the detection stage.

A third benefit is that the ROIs of different facial features constitute a loose but effective geometric constraint, which is extremely easy to implement, without the need to learn any parameters for shape representation, like PCA or conditional probability density function, as is often done in local face registration methods [32] [32] [20]. Therefore, no further correction or selection based on geometrical models is needed to refine the detection results, i.e., the detectors are purely independent[3].

2.6.3 Effective Training

For effective training, the selection of facial feature templates is important. The templates should be as consistent as possible, and at the same time as discriminative as possible. In Fig. 2.21 we show the original 19 manually labeled facial feature landmarks from the BioID database [171], and the 13 landmarks selected by us. Fig. 2.22 shows 5 representative templates out of these 13 (the remaining other 8 templates are similar according to the symmetry). For effective training, the eye template does not contain any part of the eyebrow, which possesses much larger variations. For the eyebrow, the two eyebrow ends (landmark 5,6,7,8) are used as the feature instead of the whole eyebrow region. For the nose, we use two nostrils (landmark 16 and 17) in combination to add the local structures. We did not choose the temples (landmark 9 and 14) due to its ambiguity in texture, nor the upper and lower lip edges (landmark 18 and 19) because it is easily imagined that horizontal shifts are very likely to happen in detection due to similar textures.

The sizes of the facial feature templates in the training also influence performance. In the work of Viola and Jones [179], a template size of 24×24 is chosen for finding an entire face. This may lead to the selection of smaller sizes for the facial feature templates, but it is dangerous to do so. When the size of a facial feature template is too small, the local structure of this feature will appear even more insufficient to train a reasonable detector. As in the Viola-Jones training algorithm, the weak classifiers keep being added until a certain good performance is achieved, this very possibly results in a detector containing an enormous number of weak classifiers, which is very slow in detection, but still not be able to achieve good performance.

[3] Independent means that the final localization of one facial feature is not dependent on the detection results of any other landmark positions.

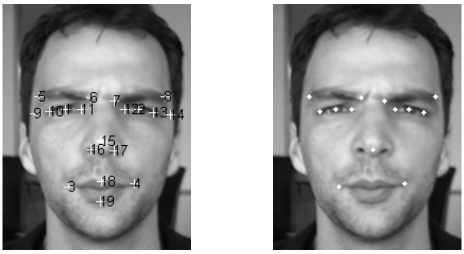


Figure 2.21: Left: the original 19 manually labeled landmarks from the BioID database, right: the 13 landmarks selected by us.

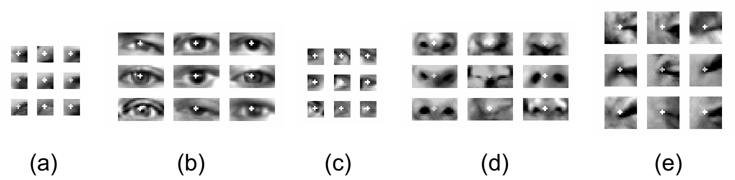


Figure 2.22: The 5 representative facial feature templates: (a) left inner eyebrow end with size of 10 × 10 pixels, (b) left eye with size of 14 × 28 pixels, (c) right inner eye corner with size of 10 × 10 pixels, (d) nose with size of 14 × 28 pixels, (e) right mouth corner with size of 20 × 20 pixels.

When the size of a facial feature template becomes too large, on the other hand, there are other risks. Firstly the detector is much slower to train, because the number of Haar-like features increases exponentially with the size. Secondly the inconsistency of texture and shape of the facial feature will play a substantial role when the details are enlarged. Thirdly the facial feature with smaller sizes will be missed in detection, like in Fig. 2.19 (a2), as the template size is the minimal scale to start with. Therefore, the sizes of the facial feature template should be large enough to contain its structure information, but not too large to present prominent inconsistency across the facial features. According to the characteristics of the different facial features, we assign different template sizes for them. For example, the eyebrow ends and the eye corners, which have larger variability than other facial features, are assigned smaller template sizes. Fig. 2.22 indicates all the template sizes, which have been validated empirically.

2.6.4 Rescaling Prior to Detection

Rescaling is an important step prior to detection. The true facial feature sizes are firstly rescaled to the size comparable to the training templates. This is done by first estimating the feature size based on the detected face size, and then rescaling the face region accordingly. Fig. 2.23 illustrates how the rescaling of face regions help to reduce the false rejections, and eliminate the false acceptances.

To evaluate the effect of introducing the rescaling, we introduce a quantitative measure d of the landmarking accuracy, which can be expressed by

$$d = \frac{\sum_{i=1}^{i=N} \sqrt{(x_i - x_{\mathrm{g},i})^2 + (y_i - y_{\mathrm{g},i})^2}}{N \cdot D_{\mathrm{i}}} \tag{2.12}$$

where N is the number of detected landmarks, x_i, y_i are the detected coordinates of the i^{th} landmark, and $x_{\mathrm{g},i}, y_{\mathrm{g},i}$ are the ground truth coordinate of the i^{th} landmark. D_{i} is the inter-ocular distance. This accuracy measure d indicates the relative error of detection.

To evaluate the benefits of rescaling, the test is done across a large number of images, and the accumulative histogram of d is calculated. The reason for accumulative histograms is that they are much easier to compare than histograms. To illustrate the benefits of rescaling, we performed tests on the above mentioned 5 facial features for the BioID database at three face sizes: 100×100 (small), 200×200 (rescaled), 500×500 (large). Fig. 2.24 shows the accumulative histogram of d for the 5 landmarks and all the 13 landmarks. Note that d

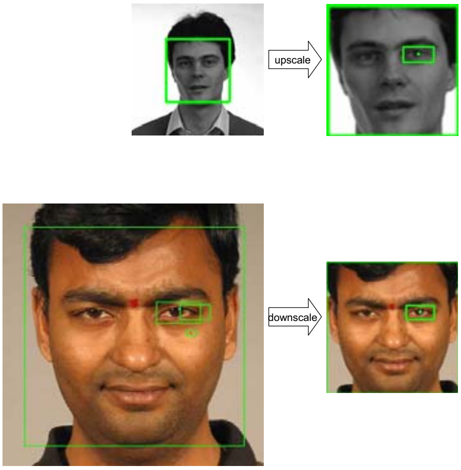


Figure 2.23: Rescaling of the face region for more accurate and robust facial feature detection Up: upscale the small-size face and find the missed landmark. Example is from the FERET [172] database, with the cropped size of 200 × 200. Down: down scale the large-size face and eliminate the false detections. Example is the high resolution image from the FRGC [173] database, with the cropped size as high as 1000 × 1000. On the right column, the face regions have been resized to 200 × 200 prior to detection.

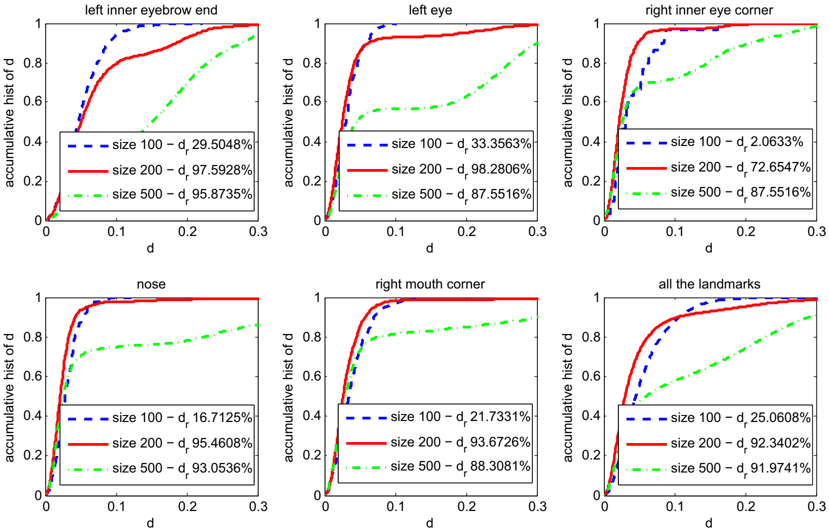


Figure 2.24: The accumulative histogram of d: dashed line - small size 100 × 100, solid line - rescaled size 200 × 200, dotted line - large size 500 × 500. The detection rate d_r is also indicated.

is averaged on all the detections, including the false detections. The detection rate, i.e. the percentage of detection, is also indicated in the figure.

Fig. 2.24 indicates that the rescaling size 200 × 200 gives the best performance with respect to both detection error and detection rate. For sizes too small, too many landmarks are missed: in total only about 25% of the landmarks are detected. For sizes too large, d is in general much larger, indicating there are a lot of false detections. The above experimental results clearly indicates that the simple rescaling procedure eliminates most of the false acceptances, saving considerable trouble in the later stage of post-selection.

2.6.5 Post-Selection using Scale Information

A single correct detection within the ROI is the ideal case for facial feature detection, as is the purpose of all the aforementioned solutions. As will be shown later in Fig 2.30 to 2.32, a single correct detection with the ROI is indeed realized in many cases, indicating that the proposed solutions are effective. In

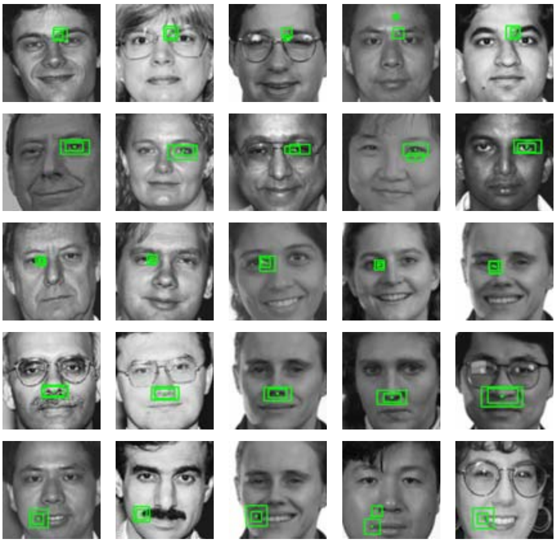


Figure 2.25: Examples of false acceptances for the 5 facial features, both type I and type II false acceptances are included.

some other cases, under more difficult conditions, however, false acceptances and false rejections are still unavoidable. False acceptances, in particular, is the problem that many local face registration methods try to deal with.

The problem of false rejections, i.e. missing of facial features, can be alleviated by training as many facial feature detectors as possible. For example, face registration, which is the main application of facial feature detection, can be reliably done given 3 or more accurately detected facial feature landmarks. The total 13 facial feature detectors, therefore, give room to 11 missing features. An image is not taken into consideration if more than 11 features are missing. We believe that when there is not enough information in the image for the well-trained detectors to localize 3 facial features, reliable face recognition or other face interpretation tasks cannot be expected either.

To look into the false acceptance problem, it is helpful to first examine what type of false acceptances will occur. Fundamentally, the Viola-Jones detector uses a combination of local structures as the template, so all the patterns that have more or less similar structures are likely to be detected. In modeling the false acceptances, two types of false acceptances can be identified. The *type*

I false acceptance is acceptance of the background patches coincidentally have comparable local structures. Examples are the chin region with shadows in Fig. 2.19 (a3), and the spot under eye in Fig. 2.23, both of which are mistaken as eyes. The *type II false acceptances* is the acceptance of the patches centered at approximately the same position as the true facial features, but larger in sizes, as shown in Fig. 2.19 (a3) left eye region. This can be explained by the fact that the facial landmarks are always located at the center of the facial feature templates, as shown in Fig. 2.22. Therefore, when searching on a slightly coarser (larger) scale around this center point, the patch still have similar structure as the true facial feature patch, and will be still bounded by the detection template in a looser way. Examples of both type of errors are shown in Fig. 2.25, in which the rescaling and the ROI constraints have already been applied. An additional geometrical constraint model can possibly eliminate the type I error, but in principle cannot deal with the type II error.

We observed from experiments that the type I false acceptances does not occur very frequently, as shown by examples in Fig. 2.25, indicating that they are mostly eliminated by the rescaling and the ROI constraints. The second type of false acceptances, in comparison, occurs more often. Although they are also reasonable detections, their locations are not accurate enough due to the coarser scales.

We propose to incorporate the scale information to provide more insight. It is interesting to notice that the accuracy of the detected facial feature is to a certain extent determined by the scales on which the facial feature is detected. Based on the above observations and analysis, we extracted a simple principle to remove the false acceptances: *minimal-scale detection within the maximal-scale detection*. The reasoning of this principle is directly related to the mechanism of the Viola-Jones method: firstly, the detections within the maximal-scale detection have less chance of being the type I false acceptances (i.e. random errors), as they have been confirmed several times by the overlapped detections. Secondly, the minimal-scale detection within the maximal-scale detection is most likely to be the accurate one, as it is bounded by the template in the tightest manner, or in other words, it is detected on the finest scale. The type II false acceptances, therefore, are employed as extra information to confirm the localization but are finally eliminated. The applicability of this principle can be illustrated by Fig. 2.25.

The accumulative histogram of d defined in (2.12) is drawn in Fig. 2.26 for the detection before and after applying the post-selection principle. Before selection d is averaged on all the detections. It can be seen from Fig. 2.26 that the error d is reduced by feature selection, indicating that many false acceptances

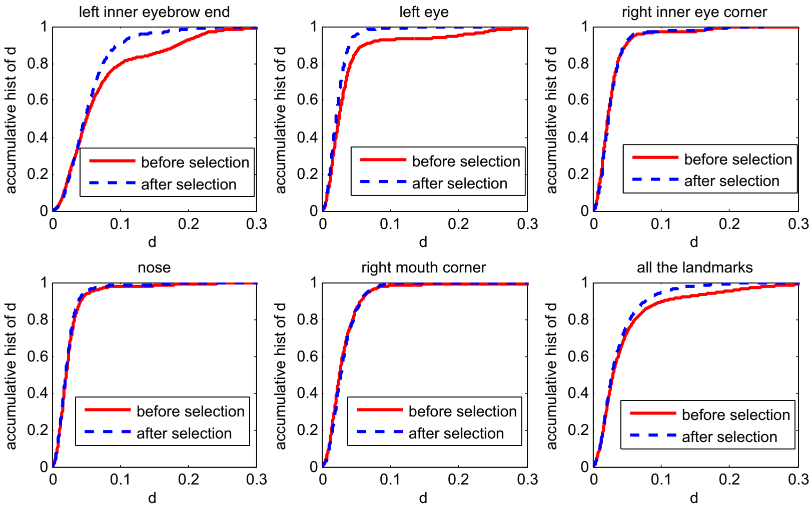


Figure 2.26: The accumulative histogram of d: solid line - before feature selection, dashed line - after feature selection.

are eliminated. For certain facial features, for example the nose, the reduction is not obvious, due to the fact that the multiple nose detections turned out to be mostly concentric.

2.6.6 Experiments and Results

The training of the facial feature detectors is done using the same scheme as in [179]. The BioID database [171] with manually labeled landmark positions is used to obtain both the positive training samples and the negative training samples in the corresponding ROI. For each detector, a positive set of 6,000 and a negative 10,000 is set for the Adaboost training as an empirically good choice. In our work 15 cascades are taken, each with a detection rate of 99.95% on the training set. Fig. 2.27 shows the number of weak classifiers in each trained cascade. Again we take the 5 examples in Fig. 2.22. It can be noticed from the table that the eyes and nose detectors have lighter structure (fewer component Haar-like features) than the others. This can be explained by the fact that eyes and noses have relatively consistent and abundant local structures compared to others, which is more favorable for the AdaBoost training. It can also be

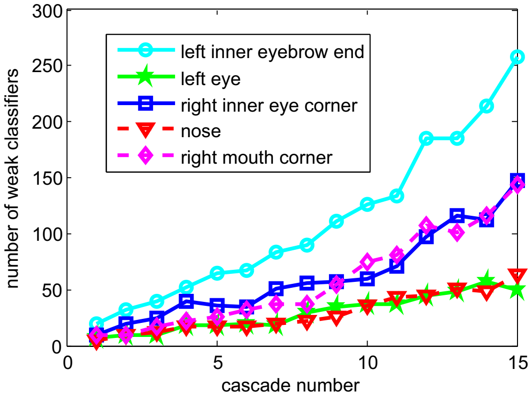


Figure 2.27: Number of weak classifiers in each trained cascade, for the 5 facial feature detectors in Fig. 2.22.

observed that the eyebrow ends are the most difficult to train, due to its large variability on simple structures.

One of the big advantages of Viola-Jones detector is its generalization ability. Once trained on a proper data set, the detector can be applied to the objects under a large range of imaging conditions. To test the generalization of the facial feature detectors, we apply the detectors trained on the BioID database to the FERET [172] and the FRGC [173] database low resolution data. In these two test set, there are 4 manually labeled facial landmarks as the ground-truth information: left eye, right eye, nose tip, and mouth center. Differences exist in the definition of landmarks as we use the center of two nostrils and the two mouth corners. The average of the two mouth corners can be taken as an estimate of the mouth center in the X coordinate, but discrepancies still occur on the Y coordinates. For fair comparison, we compare only the X coordinates of the nose and the average mouth. The accumulative error d are calculated. Fig. 2.28 shows the accumulative error for 4 facial landmarks.

It can be seen from Fig. 2.28 that the facial feature detectors have similar performance on the training set and the testing set. This generalization ability enables the facial feature detection under a large range of image conditions. Detection results from different databases, arbitrary Internet and real life images are shown in Fig. 2.30, 2.31, and 2.32.

Another big advantage of the Viola-Jones detector is its speed, which the fa-

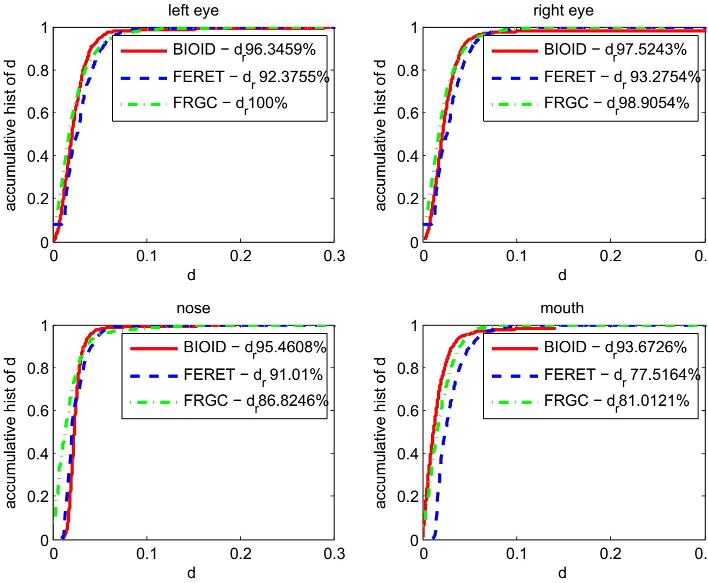


Figure 2.28: The accumulative histogram of d: solid line - BioID dataset (training)), dashed line - FERET dataset (testing), dash-dot line - FRGC dataset (testing).

cial feature detectors readily inherit. As real-time Viola-Jones face detection has already been realized, our facial feature detectors working in confined ROIs with controlled number of cascades can also achieve real-time performance, around 25 frames per second on a Pentium 4, 3.2GHz CPU. It is noteworthy that the post-selection is extremely simple and fast, virtually taking no time.

Finally we compare our detector with the results of the other two state-of-art facial feature detection approaches, which are also based on the Viola-Jones method. In both approaches, the Viola-Jones facial feature detectors are applied first, producing facial feature landmark candidates. In the first approach, CSS (combinatoric shape search) [31] determines the best combination of candidates by shape guided search. In the second approach, a multistage framework is adapted [32], in which the Viola-Jones feature detector, a shape constraint model PRFR (pairwise reinforcement of feature responses), and a AAM (active appearance model) are applied in sequence for gradually refined landmark positions. Fig. 2.29 shows the comparison results of the d on the 4 landmarks: 2 eyes and 2 mouth corners.

Given that the same type of facial feature detectors are used, we can consider the comparison as the one on the post-selection method. Fig. 2.29 indicates that our post-selection methods are effective, despite the simplicity. The reason is that the model we build for the false acceptances, type I and type II, is more accurate than a probabilistic model with respect to the detector. The false acceptances are not distributed in a continuous way around the true landmarks. The type I false acceptances are more often randomly spread, and the type II false acceptances are concentrated around the true positions. Moreover, when there exist type II false acceptances, the true detection is most likely to be present. Therefore, it is more interesting to single out the true one directly based on the judgment of the detectors (i.e., scale and overlapping information), than to estimate or correct it from the pool of uncertain detections, based upon its relationship to other facial features in a probabilistic manner. The proposed post-selection principle, therefore, is more accurate because it takes maximal advantage of the accuracy of the optimized Viola-Jones detectors, and does not include any extra errors from the additional shape or texture models.

2.6.7 Face Registration Based on Landmarks

Face registration can be done easily by aligning the detected landmarks to the corresponding reference facial feature landmarks. The parameter of translation, scaling, and rotation can be calculated accordingly.

Suppose the detected landmarks are denoted by X_{input}, in which the ele-

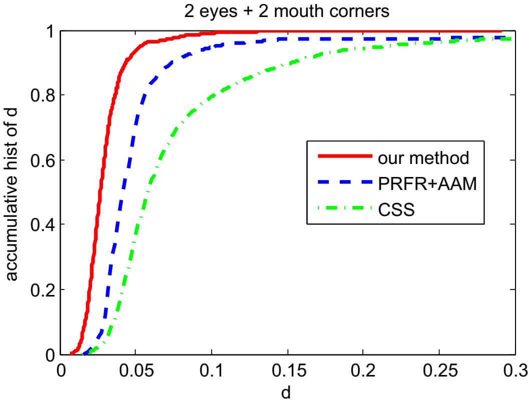


Figure 2.29: The accumulative histogram of d: solid line - BioID dataset (training)), dashed line - FERET dataset (testing), dash-dot line - FRGC dataset (testing).

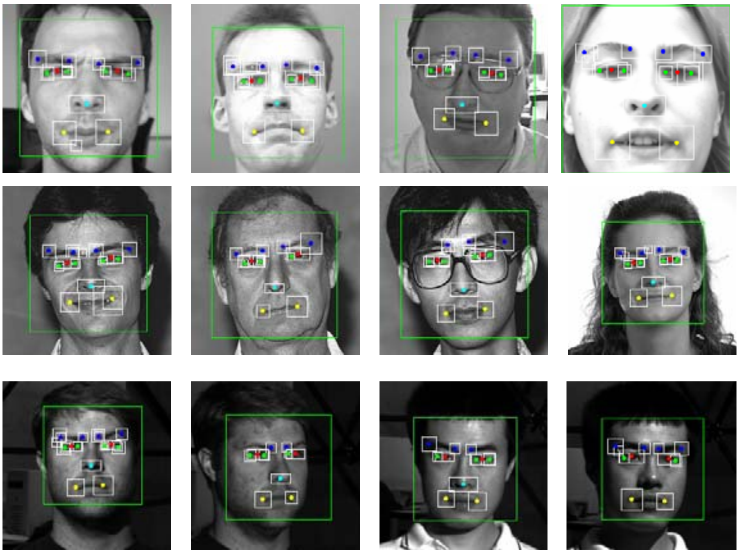


Figure 2.30: Examples from the BioID, FERET, and YaleB databases.

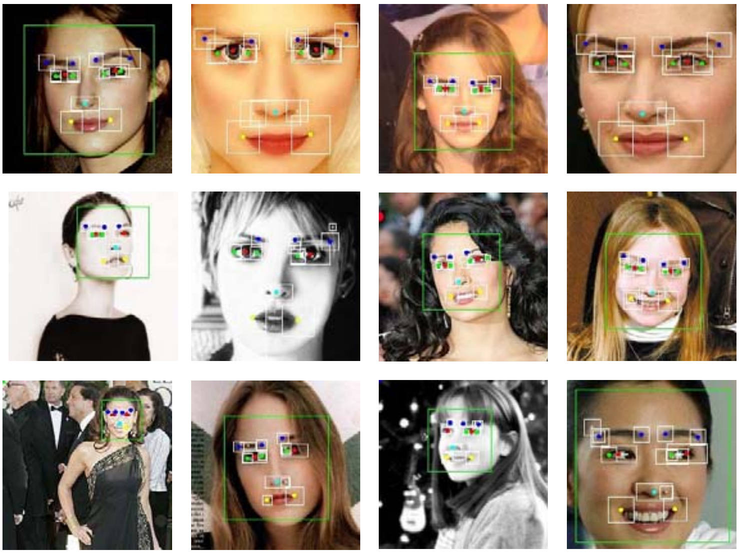


Figure 2.31: Examples from the Internet, unconstrained face images.

ments in the first column denote the x-coordinates of the detected landmarks, the elements in the second column denote the y-coordinates of the detected landmarks. The number of rows corresponds to the number of detected landmarks. X_0 denotes the corresponding standard landmarks to which X_{input} is aligned. Assume s is the scaling factor, θ is the rotation angle, x is the shift in x-direction, and y is the shift in y-direction. The registration can be expressed by

$$T = \begin{pmatrix} s\cos\theta & s\sin\theta \\ -s\sin\theta & s\cos\theta \\ x & y \end{pmatrix}, \qquad (X_{\text{input}}, \mathbf{1})\, T = X_0 \tag{2.13}$$

where T is the 3×2 transformation matrix, $\mathbf{1}$ is a column vector with all 1's. The solution of (2.13) in the least square sense is

$$X_t = (X_{\text{input}}, \mathbf{1})\,, \qquad T = (X_t^{\mathrm{T}} X_t)^{-1} X_t^{\mathrm{T}} X_0 \tag{2.14}$$

Once T is known, s, θ, x, and y can be extracted from it and then applied

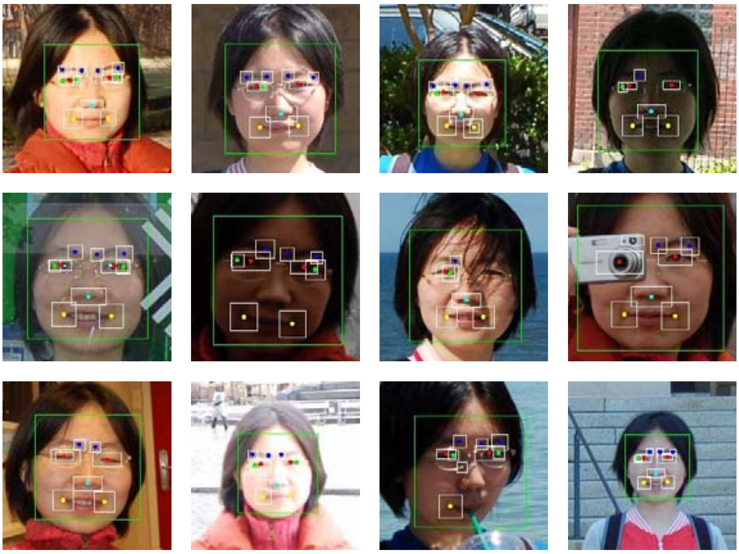


Figure 2.32: Examples of real life unconstrained face images.

to the face image. We show the fully automatic face registration system in Fig. 2.33.

2.7 Summary

This chapter presents a detailed study of face detection and face registration, which are important prior steps for any face interpretation tasks. The importance is in two senses: firstly, the accuracy of face detection and registration directly influences the performance of subsequent face recognition; secondly, face detection and registration is often the most time-consuming part of the entire face recognition system.

Face detection is not as easy for computer vision as for human cognition, due to the large variations of the human faces that cannot be easily decoded. Different methodologies have been proposed in literature, as reviewed in Section 2.2. We adapted the well-known Viola-Jones detection methods specifically for our face detection problem on an MPD. The method combines simple Haar-like rectangle features, complex Adaboost training, and efficient cascaded classifier

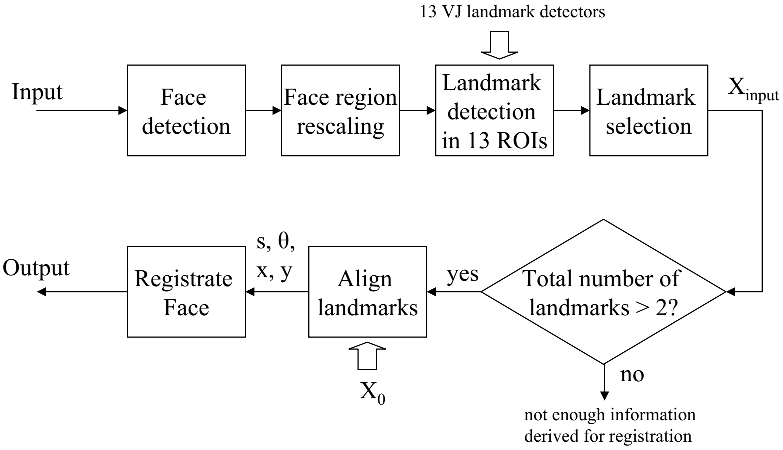


Figure 2.33: Diagram of the automatic face registration system.

structures. The difficulty of the problem is mostly diverted to the long and tedious training, whereas in detection the method works extremely fast. This resembles the natural human learning process, in that much time is spent in the infancy and childhood for learning, and high-complexity recognition ability is reached later on. In Section 2.4, adaption has also been made to boost the performance even further for our specific application.

Inspired by the good performance of the Viola-Jones method on face detection, we optimized the method for facial features, in order to realize a fast, accurate, and robust face registration system that has not been achieved before. The inherent insufficiency of the facial features, i.e., the inconsistent and inadequate structure that account for the failure of most facial feature detectors, have been thoroughly analyzed. In our work, this underlying difficulty are circumvented by a sequence of efficient solutions, namely, re-definition of the problem, regulations for effective training, re-scaling prior to detection, and strategies for post-selection. The resulting facial feature detectors are completely self-standing, without any additional shape or texture constraints that are usually required in many other methods for further processing (often in an iterative way). The trouble of learning such constraint models, together with the possible modeling error, are avoided. The biggest advantage of the proposed solutions over the previous work is the high accuracy at high simplicity. The efficiency of the solutions can be demonstrated by the error curves in Section

2.4, and the facial feature detection results in Fig. 2.30, 2.31, and 2.32. With the face detection and registration both implemented in such a fast, accurate, and robust manner, the whole system is expected to benefit greatly.

Chapter 3

Face Verification

3.1 Introduction

[1]As one of the most successful applications of image analysis and understanding, face recognition has received significant attention during the past decades. For an extensive survey, see [24], [191]. A general statement of the face recognition problem can be formulated as follows: given an image, identify the person in the scene or verify his or her identity using previously learned knowledge. *Identification* and *verification* are the two most important applications of face recognition, different in the number of subjects involved. In identification, there are multiple subjects in the gallery, and the output of identification should be the identity corresponding to the input face[2]. In verification, in contrast, there is only one subject involved, and the output of verification should be a binary decision: true or false. Simply speaking, the identification is a multi-class classification problem, whereas the verification is a two-class classification problem. Because of our application, we are most interested in face verification.

Although different in definition, identification and verification are very close face interpretation topics to study, for they both involve the same two essential parts: firstly, extracting features from the face image, and secondly, classifying those features. This is again similar to what is discussed in Section 2.2 for face

[1] This Chapter is based on the publication [158], [157], [41].

[2] Identification normally refers to a closed-group identification, meaning that the input image always belongs to a subject inside the gallery. Depending on application, identification sometimes may also be open-group, meaning that the input image can possibly be from subjects outside the gallery. In this case, the output is an identity number or "not found".

detection.

The remainder of this chapter is organized as follows. Section 4.2 reviews the face recognition methods in literature, in two categories: holistic methods and local structural methods. Section 6.2.1 proposes to use the theoretically optimal likelihood ratio classifier for our face verification problem, and Section 3.4 elaborates the methods we used for dimensionality reduction of the feature vectors before applying the likelihood ratio classifier. Section 3.5 shows the experimental results of our proposed methods, and Section 4.7 summarizes this chapter.

3.2 Review of the Face Recognition Methods

A great many face recognition methods have been proposed during the past decades. Often, a face recognition system involves techniques motivated by different principles. It has been suggested in some psychological studies that the human perception system uses both holistic and local features for recognition of the face [19] [46]. Following this guideline, we categorize the face recognition methods in three large groups:

- *Holistic Methods*
  These methods use the face region, usually the image pixel intensities concatenated as a feature vector, as the input to a classifier.
- *Local Structural Methods*
  These methods only extract certain parts from the face that are of interest, such as the facial feature locations, and the textures in the locality of facial feature. In addition, those parts are structurally connected.
- *Hybrid Methods*
  Hybrid methods combine both the holistic and local features.

We will mainly review the first two categories of methods, and the hybrid methods follows by fusing the holistic and local facial features.

3.2.1 Holistic Face Recognition Methods

Eigenface [167] and Fisherface [7] are two of the most well-know holistic face recognition methods. There are also many variants of them in literature [192] [190] [189] [85] [87] [151] [33]. Based on the fact that significant statistical redundancies exist in natural images [136], they both derive globally compact

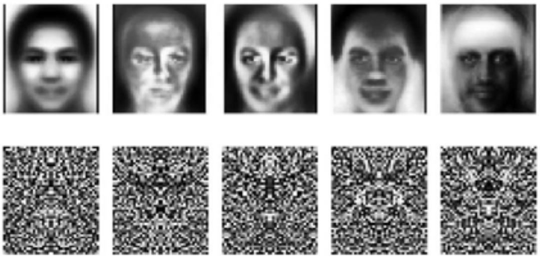


Figure 3.1: Example of eigenfaces (above) and Fisherfaces (below) [191].

representation of the face, but under different criterions: minimum reconstruction error and maximal separation, respectively. The mathematical derivations of eigenface and Fisherface are just the same as PCA and LDA, which has been introduced in Section 2.2.2, but with a major difference in the definition of classes. As a matter of fact, we can see that except for the class definition, face recognition and classification-based face detection are similar problems to solve.

In the eigenface method, an optimal projection is obtained based on the criterion of minimizing the reconstruction error. The columns of the projection matrix are called "eigenfaces", which preserve the most of the energy, i.e., corresponding to a number of the largest eigenvalues. Consequently, any sample face vector can be expressed by a linear combination of the eigenfaces. This face representation has a reduced sensitivity to noise, blurring, and partial occlusion [191]. By means of eigenface, an originally very large face vector of the concatenated pixels (usually in the order of 10,000) can be reduced to a much smaller coefficient vector (usually less than 100). The identification is done by assigning the sample face image to the identity of the one in the gallery whose coefficient vector is the closest, i.e., with the smallest Euclidean distance. Fig. 3.1 shows some examples of the eigenfaces in the top row. Obviously, the eigenfaces can be interpreted as the base for the face space, accounting for different variations. An extension of eigenface is the eigenspace approach in the hybrid manner [122], in which both the global eigenfaces and the local eigenfeatures, such as eigeneyes and eigenmouth, are used for face recognition.

In the Fisherface method, analysis of two scatter matrices is carried out. The optimal projection is obtained based on the criterion of maximizing the ratio between determinant of the between-class scatter matrix and that of the within-class scatter matrix. Solving the problem results in a projection matrix with $c-1$ columns, where c is the number of classes. Those columns of the projection

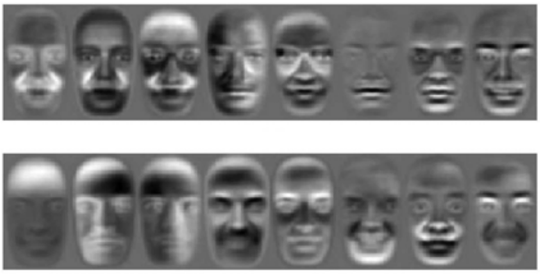


Figure 3.2: Intra-personal eigenfaces (above) and extra-personal eigenfaces (below) [108].

matrix are called ”Fisherfaces”. Similar to the eigenface method, any sample face vector is then expressed by a linear combination of the Fisherfaces. This means that the original face vector is reduced to a coefficient vector of length $c - 1$. Identification is again done by calculating distances between the input and the template coefficient vectors. In contrast to the eigenface method, the Fisherface method is a supervised learning method which makes use of the class information. It has been reported that the Fisherface method can outperform the eigenface method in recognition error [7] [47]. Fig. 3.1 shows some examples of the Fisherfaces in the bottom row. Unlike the eigenfaces, the Fisherfaces do not resemble human faces in a global way, instead, they represent the difference between classes, mainly in details.

Instead of using distances between the coefficient vectors, a probabilistic measure of similarity is used in [108] [110] [109], in which the standard eigenface approach is extended to a Bayesian approach. The multi-class problem is transformed to a two-class one, with two mutually exclusive classes defined: Ω_I representing the *intra-personal* variations between multiple images of the same subject, and Ω_E representing the *extra-personal* variations due to differences in identity. Assuming that both classes are Gaussian-distributed, likelihood functions $p(\Delta|\Omega_I)$ and $p(\Delta|\Omega_E)$ are estimated from the given difference $\Delta = I_i - I_j$, where I_i and I_j are holistic feature vectors. Using the maximum a posteriori (MAP) rule, two face images are determined to belong to the same subject if $\frac{p(\Delta|\Omega_I)}{p(\Delta|\Omega_E)} > 1$, or not if otherwise. Considerable performance improvement over the eigenface and Fisherface methods has been reported in the large scale vendor test of 2000 [125] on the FERET database [172]. We show in Fig. 3.2 the eigenfaces of the intra-personal face space and the extra-personal face space.

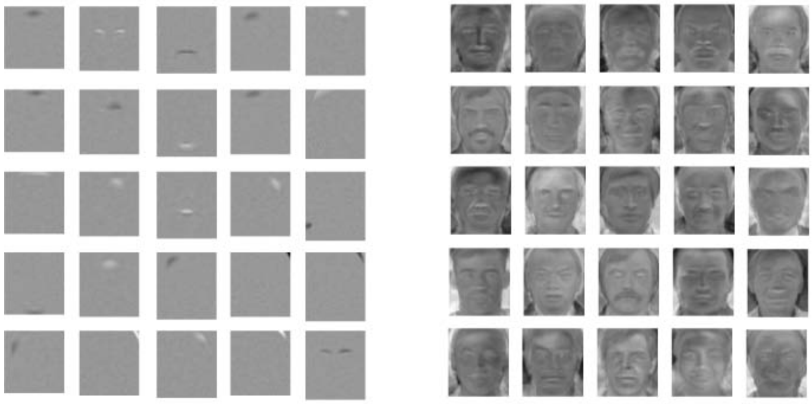


Figure 3.3: ICA basis images using two architectures: left - Architecture I, right - Architecture II [4].

Larger variations can be observed in the extra-personal face space, while subtle variations due to expressions and lightings are presented by the intra-personal face space.

PCA, LDA, and the Bayesian approach take advantage of the information up to the second order statistics of the training data. Based on the argument that important information is contained in higher order statistics, independent component analysis (ICA) has been proposed to solve the face recognition problem [4] [86] [42]. ICA is a generalization of the PCA analysis, in the sense that it decorrelates the higher-order moments in addition to the second-order one. Two ICA architectures have been proposed for face recognition: the first is a set of statistically independent source images as independent image features (Architecture I), and the second is a set of image filters that produce statistically independent outputs (Architecture II). Fig. 3.3 shows the basis images derived using the two distinct architectures. It is easily seen that they provide different interpretations of the face image statistics. The basis images from the Architecture I are spatially local, whereas the basis images from Architecture II are more global, similar to eigenfaces. For recognition, the input face is decomposed onto the ICA basis, and a cosine distance is calculated between the coefficient vectors between the input image and the gallery image [4].

Fig. 3.1, 3.2, 3.3 all show the basis face images derived under different criterions, on which any input face image can be decomposed. This is typical of the holistic face recognition methods, in that analysis is carried out on the holistic content of the face images, and as a result, the basis images are global[3]. In the same manner, many other analysis methods on the holistic face can been applied as well, such as neural networks [93], support vector machines [61] [59], evolutionary pursuit [100], etc.

The holistic face recognition methods look on the face as a whole, and this resembles the situation when we see people from a distance or in small photos, in which situation our visual system can only catch an overall image of the faces. In such cases, most often we are still able to recognize the faces. This suggests that good classification may not necessarily need a very detailed face image.

3.2.2 Local Structural Face Recognition Methods

It is natural to think of recognizing faces in a locally structured manner, for example, matching the input and gallery faces from eye to eye, and mouth to mouth. In the literature, many local structural methods has been proposed, like matching of local feature geometry [83] [3], 1D and pseudo 2D hidden Markov models [140] [139]. Early methods of this category are often limited by their simplicity as they mostly use points or lines, and thus insufficient to represent faces. The Elastic Bunch Graph Matching (EBGM) method, which incorporates more powerful local feature descriptors, i.e. wavelets, is one of the most successful local structural methods [184].

As shown in Fig. 3.4, the graph representation of a face is based on the Gabor wavelet transform, a convolution with a set of wavelet kernels. The set of 40 coefficients (5 frequencies × 8 orientations) obtained for one image point is referred to as a *jet*. A sparse collection of such jets together with information about their relative locations constitutes an image graph. The resulting bunch graph contains local texture information by the jets, and global structural information by the graph. Face recognition is based on straightforward comparison of image graphs. The locally estimated wavelet coefficients are robust to illumination, translation, distortion, rotation, and scaling. Besides, the graph allows considerable pose differences. The EBGM method have been applied to systems of face detection, recognition, pose estimation, and general object recognition tasks.

The EBGM method is typical of this category of face recognition methods, in which local features are analyzed, while structural geometrical constraints are

[3] In Fig. 3.3, the basis images from Architecture I represent local features, but the method still differs from local methods in that the local features are derived from the analysis of the holistic face images, instead of from the local analysis of facial parts.

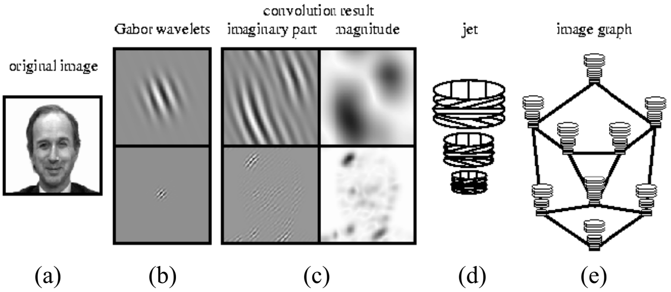


Figure 3.4: The bunch graph representation of the face using elastic graph matching[184]. From left to right: (a) input image, (b) wavelets, (c) convoluted results, (d) jet, (e) bunch graph.

used as additional information. The method is in analog to the situation when we look at a person at a close distance so that we can examine the textures, shapes, and tones of his or her facial features in much detail. In turn, for the EBGM method, this implies a relatively high-resolution image, from which the wavelets (corresponding the biological receptors on the retina) can extract necessary local textural information. For the same reason, sufficient resolution is a prerequisite for most local methods. In contrast, holistic methods are less demanding on the image resolution, as they does not particularly concentrate on the details of certain facial features.

3.3 Likelihood Ratio Based Face Verification

In the previous section, we have discussed face recognition methods in general, and reviewed the methods in two large categories: holistic methods and local structure methods. In this section, we will return to our specific face *verification* problem on the MPD. Due to the limited resolution and quality of the MPD images, we adopt the holistic approach, i.e., using the registered face image[4] to derive holistic feature vectors.

Although similar to the face detection problem, which has been discussed in Chapter 2, in the sense that both are two-class classification problems, the face

[4] The registered face image should first be preprocessed to exclude certain external influences. In this chapter, we only apply some simple preprocessing techniques, while more complicated preprocessing will be presented in Chapter 4.

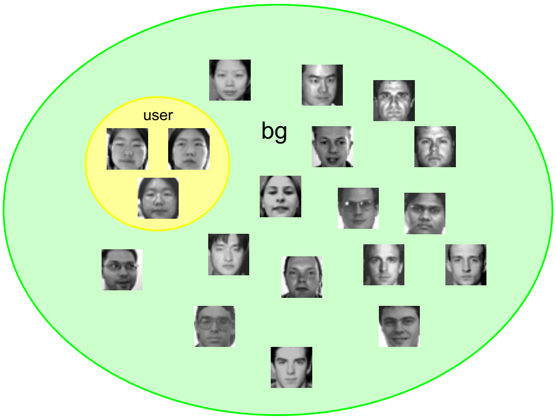


Figure 3.5: The user class and the background class.

verification can hardly achieve as good performances as face detection. The reason is that the two classes in the verification case, the user class and the impostor class, are more closely distributed in the feature space than the two classes in the detection case, the face class and the non-face class. In other words, the margins between the classes are much smaller. This implies that the boundary-based classification methods, like the SVM which relies explicitly on the support vectors, or the Viola-Jones Adaboost method which relies implicitly on the highly-weighted samples, are not suitable for the verification problem as such. In the detection case, a margin-based two-class classifier may still have satisfactory performances even if the margin is not perfectly optimized, as the distributions of the samples in both class over the margin region are really sparse. In the recognition case, the situation is different. The overlapped regions in the feature space need to be accurately classified with minimal possible error, from a statistical point of view. For this reason, we propose to verify the feature vectors in a statistically optimal way using the likelihood ratio. In this method, the two classes are conceptually modeled as two clouds in a high-dimensional space, one encompassing the other, as shown in Fig. 3.5. Our aim is to find the classification rule that yields the best performance in the statistical sense.

3.3.1 Likelihood Ratio as a Similarity Measure

Likelihood ratio is a similarity measure between an input feature vector x and the two opposite classes: $\omega = \omega_{\text{user}}$ and $\bar{\omega} = \omega_{\text{bg}}$. In this section, we will first introduce two geometrical similarity measures: the Euclidean distance and the Mahalanobis distance, and then introduce the Bayesian posterior probability and the likelihood ratio as two statistical similarity measures.

Euclidean Distance

Euclidean distance is simply calculated by

$$d_{\text{Eucl}}(x) = \sqrt{(x-T)^{\text{T}}(x-T)} \tag{3.1}$$

where x is the input feature vector and T is the template feature vector. The Euclidean distance treats all elements of the feature vector as equally important and uncorrelated. Any other information of the user and the background classes are ignored.

Mahalanobis Distance

Mahalanobis distance takes into consideration of the correlation between the elements of the feature vector by introducing the covariance matrix Σ. The distributional information of the user class is therefore included.

$$d_{\text{Maha}}(x) = \sqrt{(x-T)^{\text{T}}\Sigma^{-1}(x-T)} \tag{3.2}$$

where Σ is calculated from the training feature vectors

$$\Sigma = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x}_i)(x_i - \bar{x}_i)^{\text{T}} \tag{3.3}$$

with x_i, $i = 1, ..., N$ the training samples, and $\bar{x}_i = \frac{1}{N-1}\sum_{i=1}^{N} x_i$.

Posterior Probability

Different from the previous two geometrical measures, the posterior probability and likelihood ratio are in the statistical sense. Posterior probability is the optimal statistic in the Bayesian sense

$$p(\omega|x) = \frac{p(x|\omega)p(\omega)}{p(x)} = \frac{p(x|\omega)p(\omega)}{p(x|\omega)p(\omega) + p(x|\bar{\omega})p(\bar{\omega})} \tag{3.4}$$

where $p(x|\omega)$ is the probability density of x given the class ω, $p(\omega)$ is the prior probability of the class ω, $p(\bar{\omega})$ is the prior probability of the class $\bar{\omega}$, and $p(x)$ is the probability density of x in the whole space $\omega \cup \bar{\omega}$.

Likelihood Ratio

The likelihood ratio is an optimal statistic in the Neyman-Pearson sense [174]: at a given false acceptance rate (FAR), the likelihood ratio achieves a minimal false rejection rate (FRR); or at a given FRR, the decision-fused classifier reaches a minimal FAR. Likelihood ratio has been long known as the optimal statistic in the detection theory [174], and the verification problem is very similar. The likelihood ratio is defined as

$$L(x) = \frac{p(x|\omega)}{p(x|\bar{\omega})} \tag{3.5}$$

Since we assume infinitely many subjects in the sets $\omega \cup \bar{\omega}$, exclusion of a single subject ω from it virtually does not change the distribution of x. Therefore, the following holds

$$p(x|\bar{\omega}) = p(x) \tag{3.6}$$

The likelihood ratio contains the full distributional information of two opposite classes. Besides, it has simpler form than the posterior probability as the prior probabilities $p(\omega)$ and $p(\bar{\omega})$ are not needed.

It will be shown later that under the Gaussian assumption, the statistical similarity measures are very closely associated with the Mahalanobis distances.

3.3.2 Probability Estimation: Gaussian Assumption

To obtain the likelihood ratio of an input feature vector x with respect to two classes ω and $\bar{\omega}$, the probability density functions of the two classes $p(x|\omega)$ and $p(x|\bar{\omega})$ should first be estimated. The Gaussian assumption is often applied on a large set of data samples after an appropriate transform such as PCA or LDE, motivated by the Central Limit Theorem [44]. The multivariate Gaussian distribution is expressed by

$$p(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{(x-\mu)^{\mathrm{T}} \Sigma^{-1} (x-\mu)}{2} \right) \tag{3.7}$$

where μ is the mean feature vector, Σ is the covariance matrix, d is the dimensionality of the feature vector. Given N sample feature vectors of the face x_i, $i = 1, ..., N$, both μ and Σ can be easily estimated

$$\mu = \frac{1}{N-1}\sum_{i=1}^{N} x_i, \qquad \Sigma = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)(x_i - \mu)^{\mathrm{T}} \tag{3.8}$$

To avoid the influence of extreme samples, which are possibly caused by extraordinary illumination, pose, expression, or mis-registration, μ can also take the median of the sample vectors at every element: $\mu = \mathrm{median}(x_1, ..., x_N)$.

The two classes involved in face verification are the user class ω_{user} and the non-user background class ω_{bg}. Equivalently, the likelihood ratio in (3.5) can be rewritten

$$\begin{aligned} \ln L(x) &= \ln p_{\mathrm{user}}(x) - \ln p_{\mathrm{bg}}(x) \\ &= \frac{1}{2}\left(\ln|\Sigma_{\mathrm{bg}}| + (x - \mu_{\mathrm{bg}})^{\mathrm{T}}\Sigma_{\mathrm{bg}}^{-1}(x - \mu_{\mathrm{bg}})\right) \\ &\quad -\frac{1}{2}\left(\ln|\Sigma_{\mathrm{user}}| + (x - \mu_{\mathrm{user}})^{\mathrm{T}}\Sigma_{\mathrm{user}}^{-1}(x - \mu_{\mathrm{user}})\right) \\ &= \frac{1}{2}\left((x - \mu_{\mathrm{bg}})^{\mathrm{T}}\Sigma_{\mathrm{bg}}^{-1}(x - \mu_{\mathrm{bg}}) - (x - \mu_{\mathrm{user}})^{\mathrm{T}}\Sigma_{\mathrm{user}}^{-1}(x - \mu_{\mathrm{user}})\right) + c \end{aligned} \tag{3.9}$$

where μ_{user}, μ_{bg}, Σ_{user}, Σ_{bg} are the means and covariances of the user class and background class, respectively. The second term $c = \frac{1}{2}\left(\ln|\Sigma_{\mathrm{bg}}| - \ln|\Sigma_{\mathrm{user}}|\right)$ is a constant that can be absorbed into the thresholds of the likelihood ratio without influencing the final ROC. As (4.23) shows, the logarithm essentially reduces the probability measure to the difference between the two squared Mahalanobis distances in the user and the background class.

3.3.3 Probability Estimation: Mixture of Gaussians

The Gaussian model is a simple and useful model, however, it might oversimplify the situation in cases of arbitrary, complicated feature vector distributions. In contrast, Gaussian mixture models (GMM) are able to represent much more complex probability density functions. The model is expressed as

$$p(x|\Theta) = \sum_{i=1}^{K} w_i p(x|\theta_i) \tag{3.10}$$

where $\theta_i = \{\mu_i, \Sigma_i\}$ is the parameters of the ith Gaussian pdf $p(x|\theta_i)$, w_i is the weight or the prior of the ith component, satisfying $w_i > 0$ and $\sum_{i=1}^{N} w_i = 1$, and K is the total number of the components. Therefore, the whole unknown parameter set is: $\Theta = \{K, w_1, ..., w_K, \theta_1, ..., \theta_K\}$.

Given the sample set $\mathcal{X} = \{x_1, ..., x_N\}$, the standard method to fit this model with the sample data is the expectation-maximization (EM) algorithm, which is an iterative procedure to find the maximum likelihood (ML) estimate of the mixture parameters Θ [105] [106].

$$\log p(\mathcal{X}|\Theta) = \log \prod_{i=1}^{N} p(x_i|\Theta) = \sum_{i=1}^{N} \log \sum_{j=1}^{K} w_j p(x_i|\theta_j) \tag{3.11}$$

$$\hat{\Theta} = \arg\max_{\Theta} \{\log p(\mathcal{X}|\Theta)\} \tag{3.12}$$

The EM algorithm interprets $\mathcal{X}$ as incomplete data in the way that the associations between the samples x_i, $i = 1, ..., N$ and the Gaussian mixture components $p(x|\theta_j)$, $j = 1, ..., K$ are missing. Assume the missing information is contained in $\mathcal{Y}$, $\mathcal{Y} = \{y_1, ..., y_N\}$, where $y_i \in \{1, ..., K\}$, meaning that if $y_i = k$, the ith sample is generated from the kth mixture component. Given $\mathcal{Y}$ and $\mathcal{X}$, the likelihood of mixture parameter Θ is

$$\log \mathcal{L}(\Theta|\mathcal{X}, \mathcal{Y}) = \log p(\mathcal{X}, \mathcal{Y}|\Theta) = \sum_{i=1}^{N} \log w_{y_i} p(x_i|\theta_{y_i}) \tag{3.13}$$

The EM algorithm works in a nested way with two alternating steps: *estimation* and *maximization*. Given the current estimate of $\Theta(t)$, the posterior relationship between the samples and the Gaussian components, i.e., $p(y_i|x_i, \Theta(t))$, can be estimated using the Bayes' Rule

$$p(y_i|x_i, \Theta(t)) = \frac{w_{y_i}(t) p(x_i|\theta_{y_i}(t))}{p(x_i)|\Theta(t)} = \frac{w_{y_i}(t) p(x_i|\theta_{y_i}(t))}{\sum_{k=1}^{K} w_k(t) p(x_i|\theta_k(t))} \tag{3.14}$$

and

$$p(\mathbf{y}|\mathcal{X}, \Theta(t)) = \prod_{i=1}^{N} p(y_i|x_i, \Theta(t)) \tag{3.15}$$

where $\mathbf{y} = (y_1, ..., y_N)$ is an instance of the y_i's, $i = 1, ..., N$.

Plug (3.46) into $\log(\mathcal{X}, \mathcal{Y}|\Theta)$, the target function to be maximized, which is also called Q-function, is defined as

$$\begin{aligned} Q(\Theta, \Theta(t)) &\equiv E\left[\log p(\mathcal{X}, \mathcal{Y}|\Theta)|\mathcal{X}, \Theta(t)\right] \\ &= \sum_{\mathbf{y} \in \mathbb{Y}} \log\left(\mathcal{L}(\Theta|\mathcal{X}, \mathbf{y})\right) p(\mathbf{y}|\mathcal{X}, \Theta(t)) \end{aligned} \tag{3.16}$$

where $\mathbb{Y}$ is the space of vector $\mathbf{y}$.

The maximization step is to update the estimation of the Gaussian mixture parameters by maximizing the estimate acquired in the previous step

$$\Theta(t+1) = \arg\max_{\Theta} Q(\Theta, \Theta(t)) \tag{3.17}$$

The iterative process is repeated until convergence of the Q function. The EM method has been shown to monotonically increase the likelihood in (3.13).

Despite the popularity of the EM algorithm, there are two intrinsic drawbacks, typical of any iterative optimization algorithms, namely, sensitivity to initialization and possibility of convergence to the boundary of the parameter space. To avoid such problems, strategies like multiple random initialization [60] and deterministic annealing [169] can be incorporated. Besides, compared to the single Gaussian model, the GMM method is much more computationally expensive. In Section 3.5, we will show experimental results of likelihood ratio based face verification using GMM models, and compare the results with those of single Gaussian models.

3.4 Dimensionality Reduction

Face images normally lie in a very high-dimensional space. For example, a moderate-sized face image with 50 × 50 pixels already has a dimensionality of 2,500. High dimensionality causes problems for pattern recognition tasks, well known as the *curse of dimensionality.* Basically, high dimensionality implies a high number of parameters for characterizing the samples, which number typically grows substantially with the dimensionality of vector space [44] [96]. Taking the single Gaussian model for example, suppose the feature vector x has a dimensionality of d, then there are in total $d + \frac{(d+1)d}{2}$ parameters to estimate for μ and Σ. For the GMM model, the number is even much higher.

As a result, the number of training samples needed to efficiently represent the high-dimensional data is prohibitively high. In practice, it is often impossible

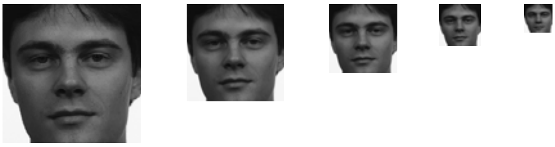


Figure 3.6: The face image pyramid with different scales.

to collect such a huge number of training samples. The reduction of the feature vector dimensionality is therefore a popular research topic. In this section, we will discuss three dimensionality reduction methods, namely, image rescaling and ROI, feature selection, and subspace method.

3.4.1 Image Rescaling and ROI

Image rescaling is the easiest way to reduce the dimensionality of the feature vectors. A larger face image certainly contains more information and represents the face in better detail, but the question is, what is the sufficient scale for the face recognition purpose? See Fig. 3.6 for the face images at different scales. In [14], it is suggested that at the face image resolution of 32×32, the PCA/LDA-based face recognition system yields the optimal recognition performance, while higher resolutions are not more favorable for recognition. Another interesting example is the Viola-Jones face detector, which uses the face template as small as 24×24, but still achieves surprisingly good detection performance [179]. For the human cognition system as well, very often we do not need a large photo to recognize a person, for example, when trying to find someone from a photo with a group of people, see Fig. 3.7, each represented in small resolution.

The dimensionality of the feature vector actually denotes its degree of freedom, and therefore the extent to which it can vary. For the identification problem, the degree of freedom must be able to characterize the differences between different subjects, and for the verification problem, the degree of freedom must allow the characterization of the difference between the user and the background class (non-users). In Section 3.5, we will carry out experiments within the likelihood ratio verification framework, and find the minimal possible scale for the purpose of dimensionality reduction.

Besides image rescaling, the ROI (region of interest) in the face is also worth investigating. A well-chosen ROI can effectively reduce the dimensionality of the feature vector, and at the same time excludes the undesirable influences of certain facial components, which are subject to high variability. We will also

Figure 3.7: A group of people with each face of very small resolution but still recognizable.

show the positive effects of assigning a ROI on the verification performance in Section 3.5.

3.4.2 Feature Selection

Another method of dimensionality reduction is to directly select a number of feature entries from the feature vector. In the face verification context, this can be intuitively understood as automatically selecting a "mask" or ROI in the face region.

Suppose the original feature set is $F = \{f_1, ..., f_d\}$, and the reduced feature set is $F_k = \{f_{\mathrm{idx}_1}, ..., f_{\mathrm{idx}_k}\}$, $(k < d)$. The problem is formally defined as follows: given the input samples and the target classification criterion, the aim of feature selection is to find k dimensions from the original d dimensions, so that the predefined criterion is optimized [121] [72] [90].

The target classification criterion often refers the minimal classification error. This error, however, is closely related to the classifier that is applied afterwards, and is therefore classifier-specific and sensitive. Instead, we adopt an classifier-independent, information theoretic criterion: *maximal mutual information between the feature vector and the class*, defined by

$$I(f; c) = \iint p(f, c) \log \frac{p(f, c)}{p(f)p(c)} df dc \tag{3.18}$$

in which both f and c are two random variables. The realization of f is the training samples in the user class and the background class, while the realization of c is the class label ω_{user} or ω_{bg} corresponding to the training samples.

Suppose F_k is the selected k-feature set $\{f_1, ..., f_k\}$[5], then the mutual information between F_k and c is

$$I(F_k; c) = \int \cdots \int p(f_1, \cdots, f_k, c) \log \frac{p(f_1, \cdots, f_k, c)}{p(f_1, \cdots, f_k)p(c)} df_1 \cdots df_k dc \tag{3.19}$$

Given the criterion of optimization, a search algorithm is needed to find the best subspace. The exhaustive search of all possible k-dimensional subspaces, however, is computationally intractable. Firstly, the total number of candidate subspaces, C_d^k, increases dramatically with the dimensionalities. Secondly, the calculation of mutual information between each candidate subspace and the class in (3.19) is also inhibitive, as the joint probability density functions in the high-dimensional space have to be estimated.

Alternatively, simplified methods have been proposed in literature, like the best individual features, incremental forward or backward search [121] [44] [72], which are much faster to implement and yield nearly optimal results. The idea underlying the simplified methods is to separately evaluate the mutual information between the individual features and the class, so that joint probability estimation can be avoided. A naive way, for example, is to find the k individual features that have the k largest mutual information with the class labels $I(f_i; c)$, $i = 1, ..., k$. The sum of mutual information between the individual features and the class is therefore maximized

$$M = \frac{1}{k} \sum_{i=1}^{k} I(f_i; c) \tag{3.20}$$

It has been well-known, however, that the combination of individually good features does not necessarily guarantee good performance together [75] [28]. One main reason is that the selected features are very likely to have strong dependencies, or redundancies, and thus other features that might otherwise contribute more to the mutual information are neglected. In the face verification case, a simple example can be given: when a feature in the feature vector (i.e., a pixel in the face image) is chosen at the eye location as the one with the largest mutual information with the class labels, the feature with the second largest mutual information might well be a pixel nearby it. As a result, the second feature adds little to the combined feature, possibly less than another pixel in

[5] For simplicity, we will denote $\{f_{\text{idx}_1}, ..., f_{\text{idx}_k}\}$ with $\{f_1, ..., f_k\}$ from now on.

the nose region. Therefore, the dependencies between the selected features must be taken into consideration. In [121], this dependency is expressed again in form of mutual information as

$$R = \frac{1}{k^2} \sum_{i=1}^{k} \sum_{j=1}^{k} I(f_i; f_j) \tag{3.21}$$

Consequently, an incremental search algorithm is proposed in [121], which optimizes the following criterion

$$f_{t+1} = \arg \max_{f \in F - F_t} \left\{ I(f; c) - \frac{1}{t} \sum_{f_i \in S_t} I(f; f_i) \right\} \tag{3.22}$$

where f_t is the t^{th} feature selected, $t = 0, ..., k-1$, $F_t = \{f_1, ..., f_t\}$. Each time a new feature is selected which can maximize the mutual information with the class, while minimize the mutual information with the previously selected features. In [121], the authors have proved that for the first-order incremental search, the criterion in (3.22) is equivalent to the maximization of the mutual information in (3.19). Obviously, the large advantage of this incremental search strategy is that the estimation of the multivariate densities $p(f_1, ..., f_k)$ and $p(f_1, ..., f_k, c)$, which are needed to calculate the mutual information $I(F_k; c)$ as in (3.19), is avoided. Instead, the estimation of the bivariate densities $p(f_i, f_j)$ an $p(f_i, c)$ is much easier and more accurate.

We summarize the feature selection procedure as in Algorithm 3. Taking advantage of the face symmetry, we can reduce the number of features to half of the original d, i.e., the features that we concern only take up half of the face region. Besides, we flipped the other half of the face and used it also as the training data $\mathbf{x}$. This means that we have less number of mutual information to estimate, from $\frac{d(d-1)}{2} + d$ to $\frac{\frac{d}{2}(\frac{d}{2}-1)}{2} + \frac{d}{2}$, and more favorably with training set doubled to $2N$.

3.4.3 Subspace Methods

So far we have discussed two direct dimensionality reduction methods that select the features in an explicit manner, i.e., the selected features can be directly mapped to the image pixels. The dimensionality can be further reduced in an implicit manner by the subspace methods, which decompose the feature space, and select the dimensions according to certain optimization criterion. Many

Algorithm 3 The feature selection algorithm.

```
Require: Given the training samples $\{\mathbf{x}_1, ..., \mathbf{x}_{N_1}\} \in \omega_{\text{user}}$, and
    $\{\mathbf{x}_{N_1+1}, ..., \mathbf{x}_N\} \in \omega_{\text{bg}}$, $N = N_1 + N_2$, $\mathbf{x}_i \in \mathbb{R}^d$.
Ensure: The reduced feature set with $k$ features, $F_{\text{reduced}} \subset F_{\text{full}}$.
  The $d$ random variables $r_1, ..., r_d$ are derived from the training samples, each
  random variable representing a feature, with $N$ realizations. See Fig. 3.8 for
  an illustration. The full feature set is $F_{\text{full}} = \{r_1, ..., r_d\}$.
  The other random variable is the class $c$, obtained by:
      $c = \begin{cases} 1 & \text{if } \mathbf{x}_i \in \omega_{\text{user}} \\ 0 & \text{if } \mathbf{x}_i \in \omega_{\text{bg}} \end{cases}$
  where $i = 1, ..., N$, referring to $N$ realizations of $c$.
  Select $f_1 = \arg\max_{f \in F} I(f; c)$ as the first feature. Define the current feature
  set $F_t = \{f_1\}$.
  for $t = 1, ..., k-1$ do
    Select $f_{t+1} = \arg\max_{f \in F - F_t} \left\{ I(f; c) - \frac{1}{t} \sum_{f_i \in F_t} I(f; f_i) \right\}$;
    Update the incremental feature set $F_t = \{f_1, ..., f_t\}$.
  end for
  The final reduced feature set is $F_{\text{reduced}} = \{f_1, ..., f_k\}$.
```

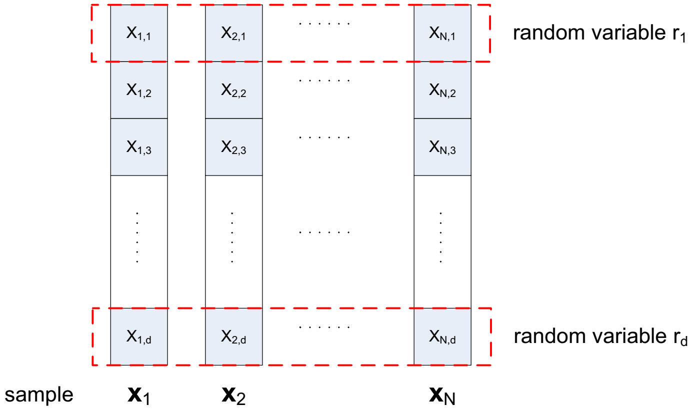


Figure 3.8: The relationship between the samples, the features, and the random variables in feature selection.

well-known face recognition methods, like PCA [167], LDA [47], and ICA [4], are examples of this category. Subspace analysis is inspired by the redundancy of per-pixel representation of the face. Firstly, the appearance of faces is highly constrained. For example, any face image is approximately symmetrical, with eyes on the sides, nose in the middle, and mouth below, etc. Therefore, the points that represent the faces only occupy a limited space in the entire image space. Secondly, the value of a pixel is typically correlated with the value of the surrounding pixels because they usually form a more or less consistent facial pattern together.

The goal of the subspace methods is to find the intrinsic dimensionality and the principal modes of subspaces. For example, Fig. 3.1, 3.2, 3.3 in Section 3.2.1, are all examples of the principal modes of the face subspace, derived under different criterions. In our work, we have proposed four different dimensionality reduction methods for likelihood ratio based face verification, namely, the personal subspace PCA, personal subspace LDA, and their kernel generalizations [157] [158]. The term "personal subspace" arises from the fact that we work particularly on the verification problem, involving only a specific user. This is in contrast to the standard identification problem in which the user-independent transformations are designed from a larger population. The personal subspace method adapts the parameters more specifically to a particular user, and is expected to have better performance than a general one. Even when there are multiple users, multiple personal subspaces are easily trained during the enrolment of each user, and a decision rule like OR can be applied for the final verification.

We have already described PCA and LDA in Section 2.2.2, with emphasis on the statement of problem and introduction of optimization criterion. In the following, we will adapt them to our verification problem, and concentrate on the mathematical solutions of the optimization problem.

Personal Subspace PCA

In this method, we derive two projection matrices with respect to the user class and the background class, separately and independently.

Suppose we have the training samples in the user class $s_1, ..., s_N$, $s_i \in \mathbb{R}^d$ where d is the dimensionality of s_i, the principal component analysis is conducted on the set of samples. The mean and covariance of the user class are obtained

$$\bar{s} = \sum_{i=1}^{M} s_i, \qquad \Sigma_{\text{user}} = \frac{1}{N-1}(s_i - \bar{s})(s_i - \bar{s})^{\mathrm{T}} \tag{3.23}$$

As introduced in Section 2.2.2, PCA solves an eigenvalue problem

$$\Sigma_{\text{user}} u = \lambda u \tag{3.24}$$

which can be solved by singular value decomposition (SVD) of Σ_{user}

$$\Sigma_{\text{user}} = U \Lambda_{\text{user}} U^{\mathrm{T}} \tag{3.25}$$

where $\Lambda_{\text{user}} = \mathrm{diag}(\lambda_1, ..., \lambda_N)$ is a diagonal matrix, and $U = [u_1, ..., u_N]$ is the orthogonal matrix satisfying $U^{\mathrm{T}} U = I$. Each eigenvalue λ_i characterizes the energy distributed along the eigenvector u_i, $i = 1, ..., N$. The eigenvalues are in decreasing order, $\lambda_1 > \lambda_2 > ... > \lambda_N$.

To reduce the dimensionality from the original d to k_1, we choose the first k_1 eigenvectors $P_{\text{user}} = [u_1, ..., u_{k_1}]^{\mathrm{T}}$ as the projection matrix. Any input feature vector x is reduced to a k_1-dimensional vector y

$$y = P_{\text{user}} x \tag{3.26}$$

The same PCA procedure is applied to the background space. Suppose we have the training samples in the background class $t_1, ..., t_N$, $t_i \in \mathbb{R}^d$. The mean and covariance of the user class are calculated in the same way as (3.23). The eigenvalue decomposition of the covariance Σ_{bg} yields

$$\Sigma_{\text{bg}} = V\Lambda_{\text{bg}}V^{\text{T}} \tag{3.27}$$

In the background space the first k_2 eigenvectors are used as the projection matrix. Any input feature vector x is then reduced to a k_2-dimensional vector y' via the projection matrix $P_{\text{bg}} = [v_1, ..., v_{k_2}]^{\text{T}}$

$$y' = P_{\text{bg}}x \tag{3.28}$$

The two PCA procedures works independently on two spaces and aim for compact representation for the two different spaces. Obviously, the user space is even more constrained than the background space. Therefore, the dimensionality of the user space k_1 is usually chosen smaller than that of the background space k_2. The criterion to determine the reduced dimensionality can be the energy preservation estimated by the sum of eigenvectors.

Now the probability density functions $p(y|\omega_{\text{user}})$ and $p(y'|\omega_{\text{bg}})$ can be estimated independently in the two reduced spaces with more ease. The likelihood ratio of the originally input vector x is rewritten as

$$L(x) = \frac{p(x|\omega_{\text{user}})}{p(x|\omega_{\text{bg}})} = \frac{p(y|\omega_{\text{user}})}{p(y'|\omega_{\text{bg}})} \tag{3.29}$$

Personal Subspace LDA

In this method, we derive one projection matrix with respect to the user class and the background class simultaneously. Different from the personal subspace PCA method in which two independent projection matrices P_{user} and P_{bg} are obtained for the two classes respectively, in the personal subspace LDA method, only one projection is derived for both classes.

The linear discriminant analysis aims to maximize the between-class scatter S_{b}, while at the same time minimize the within-class scatter S_{w} [52]. In the personal subspace LDA method, this is equivalent to the following criterion: find the projection P which satisfies

$$P = \arg\max_{P} \frac{|P^{\text{T}}\Sigma_{\text{bg}}P|}{|P^{\text{T}}\Sigma_{\text{user}}P|} \tag{3.30}$$

In the following, we will first give a simple proof of this equivalency, and then provide easy solutions for this generalized eigenvalue problem.

Proof. In the LDA method, given c classes $\omega_1, ..., \omega_c$, with N_i the number of samples in class ω_i, and N the total number $N = \sum_{i=1}^{c} N_i$. The within class scatter matrix is defined as

$$S_{\mathrm{w}} = \sum_{i=1}^{c} S_i$$

where

$$S_i = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^{\mathrm{T}}, \qquad \mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x$$

The between-class scatter matrix is defined as

$$S_{\mathrm{b}} = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^{\mathrm{T}}$$

where

$$\mu = \frac{1}{N} \sum_{x} x = \frac{1}{N} \sum_{i=1}^{c} N_i \mu_i$$

Summing S_{w} and S_{b}, we have

$$\begin{aligned} S_{\mathrm{w}} + S_{\mathrm{b}} &= \sum_{i=1}^{c} \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^{\mathrm{T}} + \sum_{i=1}^{c} \sum_{x \in \omega_i} (\mu_i - \mu)(\mu_i - \mu)^{\mathrm{T}} \\ &= \sum_{i=1}^{c} \sum_{x \in \omega_i} (x - \mu)(x - \mu)^{\mathrm{T}} \\ &= \sum_{x} (x - \mu)(x - \mu)^{\mathrm{T}} \\ &= S_T \end{aligned}$$

where S_T is the total scatter matrix. Therefore, the LDA criterion can be rewritten

$$\begin{aligned} P &= \arg\max_{P} \frac{|P^{\mathrm{T}} S_{\mathrm{b}} P|}{|P^{\mathrm{T}} S_{\mathrm{w}} P|} \\ &= \arg\max_{P} \frac{|P^{\mathrm{T}} S_{\mathrm{T}} P|}{|P^{\mathrm{T}} S_{\mathrm{w}} P|} \end{aligned} \qquad (3.31)$$

In the user verification case, by definition we have for the user class

$$\Sigma_{\text{user}} = N_{\text{user}} S_{\text{b}}$$

and for the background class

$$\Sigma_{\text{bg}} = N_{\text{bg}} S_{\text{bg}} = N_{\text{bg}} S_{\text{T}}$$

For the second equality, note that in the verification case, the background class is identical to the total class, as we assume that the full set includes infinitely many classes, and exclusion of a single class virtually does not alter its distribution (Section 3.3.1).

Referring to (3.31), the criterion of the finding the projection in personal space LDA is

$$\begin{aligned} P &= \arg\max_{P} \frac{|P^{\text{T}} S_{\text{T}} P|}{|P^{\text{T}} S_{\text{w}} P|} \\ &= \arg\max_{P} \frac{|P^{\text{T}} \Sigma_{\text{bg}} P|}{|P^{\text{T}} \Sigma_{\text{user}} P|} \end{aligned}$$

□

To solve the generalized eigenvalue problem in (3.30), we derive the projection matrix P in two steps: $P = P_2 P_1$, in which P_1 and P_2 are two orthogonal matrices, satisfying $P_1^{\text{T}} P_1 = I$, $P_2^{\text{T}} P_2 = I$. P_1 and P_2 diagonalize the covariance matrices Σ_{user} and Σ_{bg} simultaneously.

In the first step, P_1 whitens the Σ_{user} in the denominator. Eigenvalue decomposition of Σ_{user} yields

$$\Sigma_{\text{user}} = U \Lambda_{\text{user}} U^{\text{T}}$$

where Λ_{user} is a diagonal matrix, whose inverse can be easily calculated. The whitening matrix P_1 is

$$P_1 = \Lambda_{\text{user}}^{-\frac{1}{2}} U^{\text{T}} \tag{3.32}$$

which satisfies $P_1 \Sigma_{\text{user}} P_1^{\text{T}} = I$. The purpose of whitening is that the optimization problem can be now simplified. Suppose after applying the first projection, the whitened covariance matrices are Σ'_{user} and Σ'_{bg}

$$\Sigma'_{\text{user}} = P_1 \Sigma_{\text{user}} P_1^{\mathrm{T}} = I, \qquad \Sigma'_{\text{bg}} = P_1 \Sigma_{\text{bg}} P_1^{\mathrm{T}}$$

The optimization problem (3.30) is now reduced to

$$P = P_2 P_1$$

$$\begin{aligned}
P_2 &= \arg\max_{P_2} \frac{|P_2^{\mathrm{T}} \Sigma'_{\text{bg}} P_2|}{|P_2^{\mathrm{T}} \Sigma'_{\text{user}} P_2|} \\
&= \arg\max_{P_2} \frac{|P_2^{\mathrm{T}} \Sigma'_{\text{bg}} P_2|}{|P_2^{\mathrm{T}} I P_2|} \\
&= \arg\max_{P_2} \frac{|P_2^{\mathrm{T}} \Sigma'_{\text{bg}} P_2|}{|I|} \\
&= \arg\max_{P_2} |P_2^{\mathrm{T}} \Sigma'_{\text{bg}} P_2|
\end{aligned}$$

which can be simply solved by a SVD of Σ'_{bg}

$$\Sigma'_{\text{bg}} = V \Lambda_{\text{bg}} V^{\mathrm{T}}$$

Suppose the dimensionality is reduced from d to k, then the optimized projection matrix P_2 is constituted of the eigenvectors corresponding to the first k largest eigenvalues

$$P_2 = [v_1, ..., v_k]^{\mathrm{T}} = V_k^{\mathrm{T}} \tag{3.33}$$

where v_i is the ith column of V. Finally, we have projection matrix P for dimensionality reduction, which satisfies (3.30)

$$P = P_2 P_1 = V_k^{\mathrm{T}} \Lambda_{\text{user}}^{-\frac{1}{2}} U^{\mathrm{T}} \tag{3.34}$$

Another way of solving (3.30) is to solve an equivalent problem

$$P = \arg\min_{P} \frac{|P^{\mathrm{T}} \Sigma_{\text{user}} P|}{|P^{\mathrm{T}} \Sigma_{\text{bg}} P|}$$

in a similar way, by first whitening the background covariance matrix Σ_{bg}, and then calculating the eigenvectors of the whitened covariance matrix Σ'_{user} corresponding to the k smallest eigenvalues. Same solutions will be obtained.

Any input feature vector x is then reduced to a k-dimensional vector y via the projection matrix P

$$y = Px \tag{3.35}$$

The likelihood ratio of the originally input vector x is rewritten as

$$L(x) = \frac{p(x|\omega_{\text{user}})}{p(x|\omega_{\text{bg}})} = \frac{p(y|\omega_{\text{user}})}{p(y|\omega_{\text{bg}})} \tag{3.36}$$

in which $p(y|\omega_{\text{user}})$ and $p(y|\omega_{\text{bg}})$ are more easily estimated in a dimensionality-reduced feature space.

Personal Subspace KPCA

Kernel Principal Component Analysis (KPCA) [144] is a nonlinear generalization of the original PCA method. By introducing the same "kernel trick" as in support vector machines [175] [16] [30], the feature vectors are projected onto a higher dimensional nonlinear space. The basic idea of KPCA is illustrated in Fig. 3.9.

The kernel function is an inner product function. In the linear space, the kernel function is a simple dot product function

$$k(x_1, x_2) = (x_1 \cdot x_2) = x_1^{\mathrm{T}} x_2 \tag{3.37}$$

where x_1 and x_2 are two feature vectors in the linear space.

Suppose the feature vectors are projected to a nonlinear space by a nonlinear function $x_1' = \Phi(x_1)$, $x_2' = \Phi(x_2)$, then the inner product in the nonlinear space is

$$k(x_1, x_2) = (\Phi(x_1) \cdot \Phi(x_2)) \tag{3.38}$$

which is function with vector input and scalar output.

$k(x_1, x_2)$ can be any function that satisfies Mercer's theorem [175]: if K is the continuous kernel of an integral operator that is positive definite, we can construct a mapping into a space where K acts as a dot product. Common kernels includes polynomial kernel $k(x_1, x_2) = (x_1 \cdot x_2)^d$ and Gaussian radius function kernel $k(x_1, x_2) = \exp(-\frac{\|x_1 - x_2\|^2}{2\sigma^2})$ [30]. Consequently, if any algorithm can be written in the form of the dot product between the feature vectors, the kernel function $K(x_i, x_j)$ can be plugged in to substitute the dot product $(x_i \cdot x_j)$, and thus the nonlinear projection of feature vectors is implicitly done.

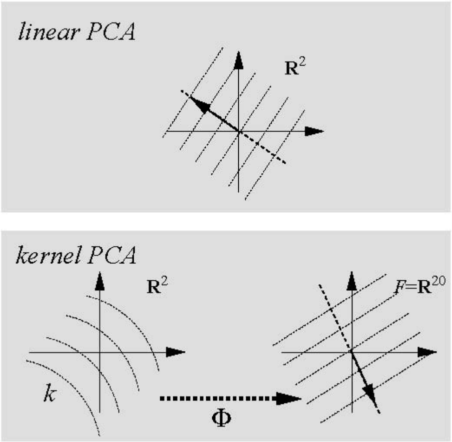


Figure 3.9: The basic idea of KPCA (from [144]): for a complex manifold as in bottom left, we can project it into a higher dimensional space, and the principal manifold is solved by performing linear PCA in this space, as in bottom right. Projecting back to the original space, the principal manifold is nonlinear.

Firstly we have to transform the PCA algorithm in such a form that the relationship between the feature vectors is expressed by dot product. For simplicity, we assume all the feature vectors x_i, $i = 1, ..., N$, are mean-removed[6]. The PCA problem in (3.24) can be rewritten

$$\left(\frac{1}{N} \sum_{i=1}^{N} x_i x_i^{\mathrm{T}} \right) u = \lambda u$$

Thus,

$$\begin{aligned} u &= \frac{1}{N\lambda} \sum_{i=1}^{N} x_i x_i^{\mathrm{T}} u \\ &= \frac{1}{N\lambda} \sum_{i=1}^{N} (x_i \cdot u) x_i \end{aligned} \tag{3.39}$$

which implies the eigenvector u is in the span of $x_1, ..., x_N$.

Define a new coefficient vector $\alpha = [t_1, ..., t_N]^{\mathrm{T}}$, in which $t_i = (x_i \cdot u)$. Represent the samples in a matrix $X = [x_1, ..., x_N] \in \mathbb{R}^{d \times N}$ with each column a sample feature vector, where d is the original dimensionality of the feature vectors, then α is

$$\alpha = X^{\mathrm{T}} u \tag{3.40}$$

Rewrite (3.39) in matrix form

$$u = \frac{1}{N\lambda} X \alpha \tag{3.41}$$

Substitute (3.41) into (3.40), we have

[6] For mean-centered samples, the computation of the kernel matrix K as in (3.44) becomes more complicated

$$\tilde{k}(x_i, x_j) = k(x_i, x_j) - \frac{1}{N} \sum_{p=1}^{N} k(x_i, x_p) - \frac{1}{N} \sum_{q=1}^{N} k(x_q, x_j) + \frac{1}{N^2} \sum_{p=1}^{N} \sum_{q=1}^{N} k(x_p, x_q)$$

or, equivalently,

$$\tilde{K} = K - 1_N K - K 1_N + 1_N K 1_N$$

where $(1_N)_{ij} = \frac{1}{N}$, for $i = 1, ..., N$, $j = 1, ..., N$. See [144] Appendix A.

$$\alpha = \frac{1}{N\lambda} X^{\mathrm{T}} X \alpha \tag{3.42}$$

Further transformation yields

$$\lambda\alpha = \frac{X^{\mathrm{T}} X}{N} \alpha \tag{3.43}$$

This is an eigenvalue problem, in which α the eigenvector of $\frac{X^{\mathrm{T}} X}{N}$, and λ the corresponding eigenvalue. $X^{\mathrm{T}} X$ can be calculated through dot product between the feature vectors

$$X^{\mathrm{T}} X = \begin{pmatrix} (x_1 \cdot x_1) & (x_1 \cdot x_2) & \dots & (x_1 \cdot x_N) \\ (x_2 \cdot x_1) & (x_2 \cdot x_2) & \dots & (x_2 \cdot x_N) \\ \vdots & \vdots & & \vdots \\ (x_N \cdot x_1) & (x_N \cdot x_2) & \dots & (x_N \cdot x_N) \end{pmatrix}$$

which can be calculated by kernel function only. In the projected nonlinear space, $(x_i \cdot x_j)$ is simply substituted by $k(x_i, x_j)$. Define

$$K = \frac{X^{\mathrm{T}} X}{N} = \frac{1}{N} \begin{pmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_N) \\ \vdots & \vdots & & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & \dots & k(x_N, x_N) \end{pmatrix} \tag{3.44}$$

The coefficient vector α_i and the associated λ_i is obtained by taking the eigenvector and eigenvalue of K, $i = 1, ..., m$, where m is the reduced dimensionality. Referring to (3.41) which represents one vector, the projection matrix U is

$$U = X \begin{pmatrix} \frac{\alpha_1}{N\lambda_1} & \frac{\alpha_2}{N\lambda_2} & \dots & \frac{\alpha_m}{N\lambda_m} \end{pmatrix} = XA \tag{3.45}$$

in which $A = \begin{pmatrix} \frac{\alpha_1}{N\lambda_1} & \frac{\alpha_2}{N\lambda_2} & \dots & \frac{\alpha_m}{N\lambda_m} \end{pmatrix} \in \mathbb{R}^{N \times m}$, $U \in \mathbb{R}^{d \times m}$. To project the original input feature vector x onto the principal manifolds, we have

$$y = U^{\mathrm{T}}x = A^{\mathrm{T}}X^{\mathrm{T}}x = A^{\mathrm{T}} \left(\begin{array}{c} (x_1 \cdot x) \\ (x_2 \cdot x) \\ \vdots \\ (x_N \cdot x) \end{array} \right) = A^{\mathrm{T}} \left(\begin{array}{c} k(x_1, x) \\ k(x_2, x) \\ \vdots \\ k(x_N, x) \end{array} \right) \tag{3.46}$$

where $y \in \mathbb{R}^m$ is the projected result. In this projection, the calculation only involves the dot products between the input feature vector x and the training feature vectors x_i, $i = 1, ..., N$. By far we have obtained the desired dimensionality-reduced feature vector y.

The personal space KPCA method carries out the above nonlinear dimensionality reduction procedure in both the user and the background space. The final likelihood ratio of the reduced feature vectors is calculated in the same way as in (3.29).

Personal Subspace KLDA

Given the mathematical derivations in personal subspace KPCA and personal subspace LDA, we can generalize the personal subspace LDA to its kernel version. The feasibility lies in the way we solve LDA, as in (3.32), (3.33), and (3.34): the final LDA projection matrix P is divided into two PCA-like procedures, which can be generalized by using the KPCA strategies.

In (3.32), The first whitening matrix P_1 is

$$P_1 = \Lambda_{\text{user}}^{-\frac{1}{2}} U^{\mathrm{T}}$$

Referring to (3.45), we have

$$P_1 = \Lambda_{\text{user}}^{-\frac{1}{2}} A^{\mathrm{T}} X_{\text{user}}^{\mathrm{T}} \tag{3.47}$$

where $A = \begin{pmatrix} \frac{\alpha_1}{N_1\lambda_1} & \frac{\alpha_2}{N_1\lambda_2} & \cdots & \frac{\alpha_{N_1}}{N_1\lambda_{N_1}} \end{pmatrix} \in \mathbb{R}^{N_1 \times N_1}$, $P_1 \in \mathbb{R}^{N_1 \times d}$. The pairs $\{\alpha_i, \lambda_i\}$, $i = 1, ..., N_1$, are the eigenvector and eigenvalues of the kernel matrix K_{user}, defined in the same way as in (3.44).

Given P_1, the whitened background samples are calculated as in (3.46). Let X_{bg} be the sample matrix of the background class, then the whitened samples and their kernel matrix are

$$X'_{\mathrm{bg}} = P_1 X_{\mathrm{bg}}, \qquad K'_{\mathrm{bg}} = \frac{(X'_{\mathrm{bg}})^{\mathrm{T}} X'_{\mathrm{bg}}}{N_2}$$

In (3.33), the second projection matrix P_2 is

$$P_2 = V_k^{\mathrm{T}}$$

Referring to (3.45), we have

$$P_2 = B_m^{\mathrm{T}} (X'_{\mathrm{bg}})^{\mathrm{T}} \tag{3.48}$$

where $B_m = \begin{pmatrix} \frac{\alpha_1}{N_2\lambda_1} & \frac{\alpha_2}{N_2\lambda_2} & \cdots & \frac{\alpha_m}{N_2\lambda_m} \end{pmatrix} \in \mathbb{R}^{N_2 \times m}$, $P_2 \in \mathbb{R}^{m \times N_1}$. The pairs $\{\alpha_i, \lambda_i\}$, $i = 1, ..., m$, are the eigenvector and eigenvalues of the kernel matrix K'_{bg}. The final KLDA projection matrix is

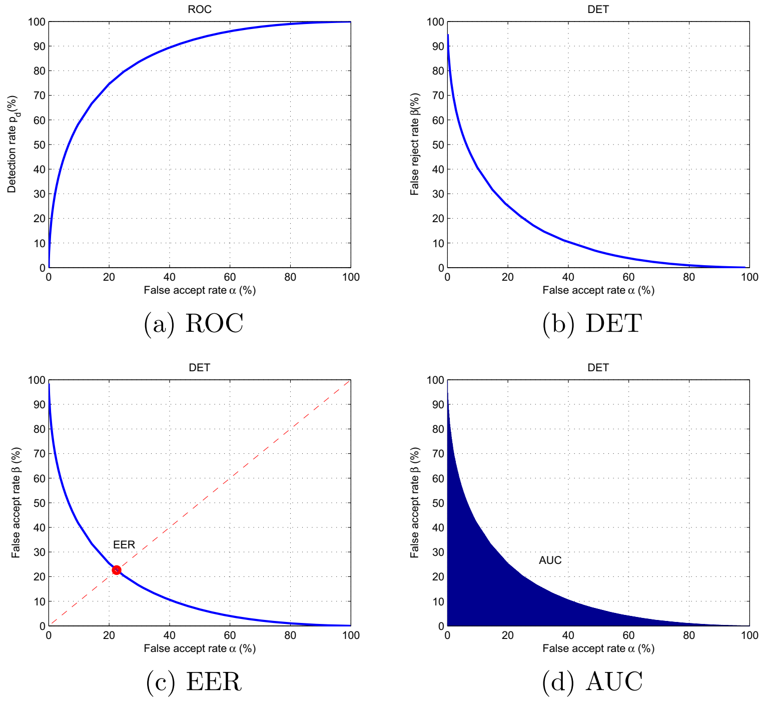
$$P = P_2 P_1 = B_m^{\mathrm{T}} (X'_{\mathrm{bg}})^{\mathrm{T}} \Lambda_{\mathrm{user}}^{-\frac{1}{2}} A^{\mathrm{T}} X_{\mathrm{user}}^{\mathrm{T}} \tag{3.49}$$

The projection matrix $P \in \mathbb{R}^{m \times d}$, and the projection can be calculated via kernel functions between the training user sample data $x_1, ..., x_{N_1}$ with any input sample x

$$y = Px = B_m^{\mathrm{T}} (X'_{\mathrm{bg}})^{\mathrm{T}} \Lambda_{\mathrm{user}}^{-\frac{1}{2}} A^{\mathrm{T}} \begin{pmatrix} k(x_1, x) \\ k(x_2, x) \\ \vdots \\ k(x_{N_1}, x) \end{pmatrix} \tag{3.50}$$

Finally, the likelihood ratio of the dimensionality reduced feature vectors y can be then calculated as in (3.36).

Theoretically, we have had the complete solution to the KLDA problem, but the computation involved can be very expensive, especially when the number of training samples is large. In such cases, the kernel matrix K is also large, which makes itself slow to calculate, and renders the eigenvalue decomposition of it very slow, too. Meanwhile, the projection in (3.50) requires calculating the kernel function between the input x and all the training samples, and this is again very expensive. A close look at the projection functions for KPCA



(a) ROC

(b) DET

(c) EER

(d) AUC

Figure 3.10: Illustration of the performance measures: (a) ROC, (b) DET, (c) EER, (d) AUC.

(3.46) and KLDA (3.50) reveals that the form is alike to that of the neural network [12], in the sense that the output (each element of y) is a weighted sum of the kernel functions between the training samples and the input pattern. The weights, however, are directly derived in KPCA or KLDA in closed form, instead of iteratively determined as in neural network.

3.5 Experiments and Results

3.5.1 Performance Measures

The verification is basically comes down to thresholding the calculated likelihood ratio. The acceptance region R_a and the rejection region R_r at a certain threshold t are defined as

$$R_\mathrm{a}(t) = \{x | L(x) \geq t\} \tag{3.51}$$

$$R_\mathrm{r}(t) = \{x | L(x) < t\} \tag{3.52}$$

where $L(\cdot)$ is the calculated likelihood ratio.

To measure the performance, there are two important quantities: the FAR, denoted by α, and the FRR, denoted by β. Although mentioned many times before, here strict mathematic definitions are given

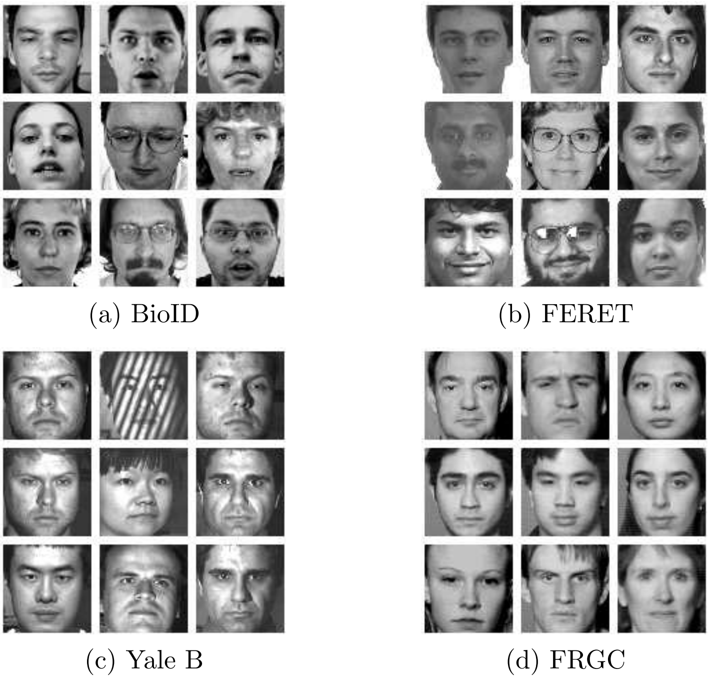
$$\alpha(t) = P(x \in R_\mathrm{a}(t) | x \notin \omega) = \int_{R_\mathrm{a}(t)} p(x|\bar{\omega})dx \tag{3.53}$$

$$\beta(t) = P(x \in R_\mathrm{r}(t) | x \in \omega) = \int_{R_\mathrm{r}(t)} p(x|\omega)dx \tag{3.54}$$

The dependency of the above quantities on the threshold t leads to two equivalent performance measures that are widely used. When t varies, the FAR can be seen as a function of the FRR, $\beta(\alpha)$, known as the detection error trade-off characteristic (DET) [103]. Another popular measure is the receiver operating characteristic (ROC), in which the detection rate is expressed as a function of FAR, $p_\mathrm{d}(\alpha)$ [49]. Both DET and ROC are monotonic curves, illustrating the overall performance across the whole range of thresholds. Sometimes the equal error rate (EER) and the area under curve (AUC) are used as reduced measures of performance. Fig. 6.1 illustrates the performance measures that were discussed above.

3.5.2 Data Collection

To learn the probability density functions of the user class $p(x|\omega_\mathrm{user})$ and the background class $p(x|\omega_\mathrm{bg})$, a large number of samples are required. This is especially true in case of estimating the GMM models. For accurate and robust estimation, we expect that the number of samples is high compared to the dimensionality of feature vectors, and the number of unknown parameters is


(a) BioID

(b) FERET

(c) Yale B

(d) FRGC

Figure 3.11: Examples of the background faces from BioID, FERET, Yale B, and FRGC, detected and registrated using the methods in Chapter 2.

low. Otherwise overtraining is likely to occur and leads to poor generalization ability of the verification system.

The Background Class

The background sample set can be taken from the public face databases. In the experiments we adopt four databases, namely the BioID database [171], FERET database [172], Yale B database [56], and FRGC database [173]. The faces are detected and registrated using the methods proposed in Chapter 2. Fig. 3.11 gives some examples of the faces in the databases. The databases in total result in more than 10,000 samples for training in the background class.

The User Class

The user sample set are obtained by taking face images from the MPD. We used the Eten M600 PDA as the mobile device. Practically the user set is easy to collect: at a frame rate of 15 fps, one-minute video ends up with 900 frames of face images. In total the data of 20 users has been collected from volunteers, with 4 independent sessions for each subject, taken at different time and under different illuminations. Fig. 3.12 gives four examples of the collected user faces, each example are shown in two sessions. Besides, the training set is extended by creating slightly shifted, rotated, and scaled versions of the face image. This step increases the within-class variation only slightly, as the previous step, face registration, also causes some variations on this level.

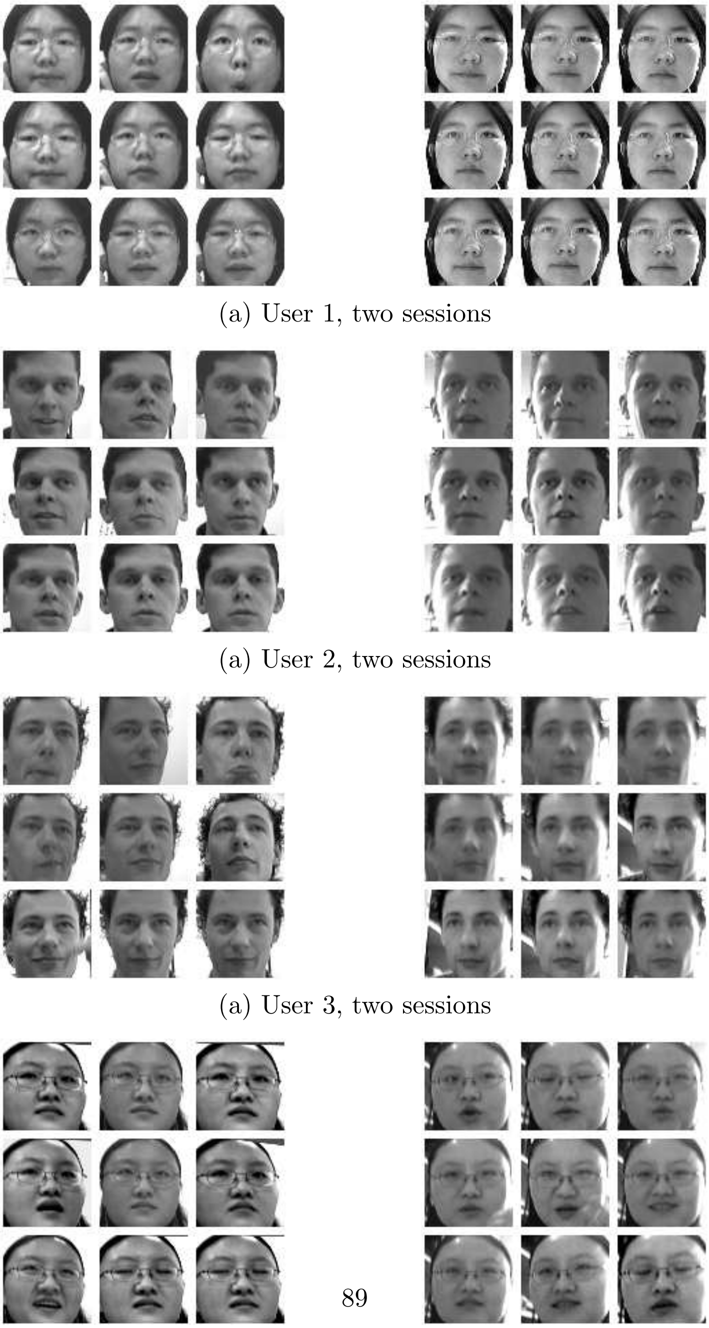
Test Protocol

In the following, we will carry out a series of experiments. To evaluate the verification performances using different methods and parameters, it is necessary to first specify the test protocol.

For one specific user, we learn the density of the user class from the user data of one session, and compute the testing genuine scores, i.e., the likelihood ratios, from the user data of the other three independent sessions. To remove the cross-session variations, we used simple preprocessing methods, like the zero-mean and unit-variance normalization and the histogram equalization on the images. The main purpose of the experiments is to find effective and simple ways of calculating the likelihood ratio, while more complicated illumination normalization problem will be introduced in the next chapter.

The density of the background class is learned from the public database, and the impostor scores are computed from our collected data of the other 19 subjects. Note that the training and testing data of the background class are different in the setting. Using the public databases as the training data is convenient for the MPD implementation, as the background parameters need to be trained only once, and stored for all the users. On the other hand, obtaining the impostor scores from our own database is of more interest than obtaining them from the public database, because the face images in our own database are collected under more or less the same situation, and thus more meaningful and critical for testing the verification performance.

Given the genuine and the impostor scores, the ROC can be obtained from them to show the performance of the system. The EER can be derived from the ROC as a performance measure.



(a) User 1, two sessions

(a) User 2, two sessions

(a) User 3, two sessions

(a) User 4, two sessions

Figure 3.12: Examples of user faces in two independent sessions, detected and registrated using the methods in Chapter 2.

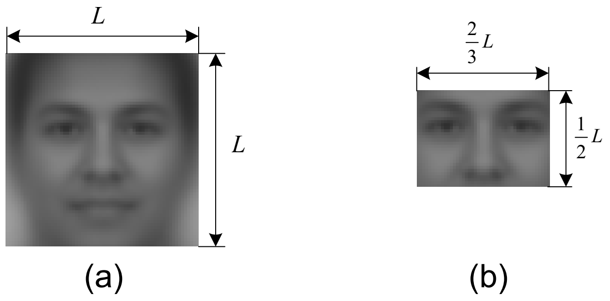


Figure 3.13: (a) The originally registered face square, (b) the ROI taken.

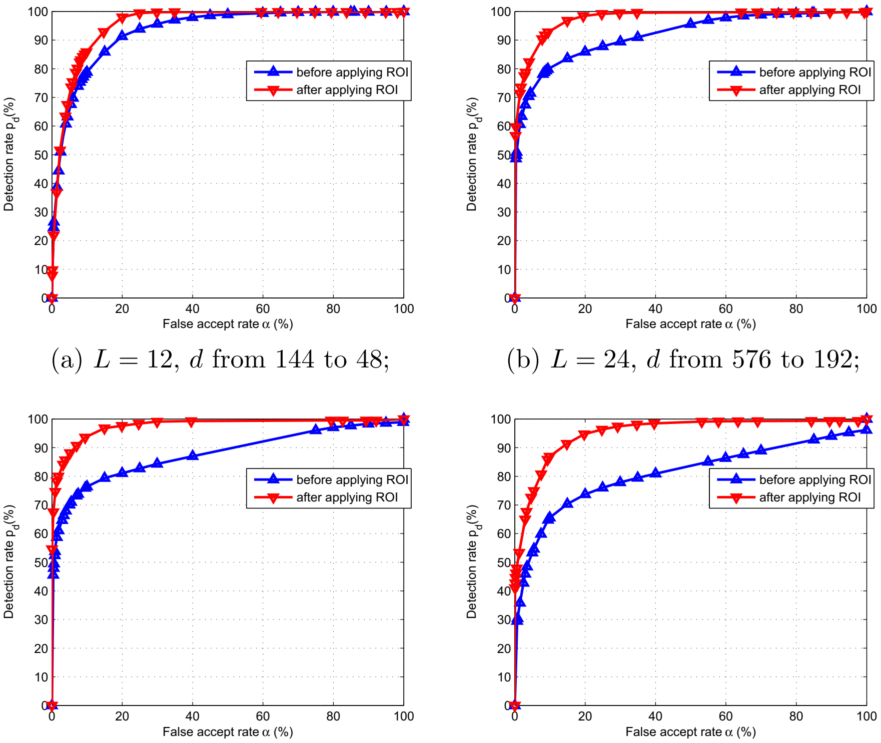
3.5.3 Experimental Results

We present the experimental results of face verification using the likelihood ratio classifier. To calculate the likelihood ratio, the probability densities of the user class ω_{user} and the background class ω_{bg} have to be estimated. Prior to the density estimation, a dimensionality reduction procedure can be applied to the feature vectors. To find the best and easiest way of obtaining the likelihood ratio, we carried out experiments with different set-up:

- For dimensionality reduction:
  - Rescaling the image, as introduced in Section 3.4.1;
  - Feature selection, as introduced in Section 3.4.2.
  - Personal subspace PCA, LDA, KPCA, and KLDA, as introduced in Section 3.4.3;
- For density estimation:
  - Single Gaussian model, as introduced in Section 3.3.2;
  - Gaussian mixture model, as introduced in Section 3.3.3.

Scale and ROI

We first investigate the influence of the region of interest (ROI) and the face scale on the verification performances. To evaluate this parameter, for simplicity, we used the single Gaussian model, and used the feature vector of the



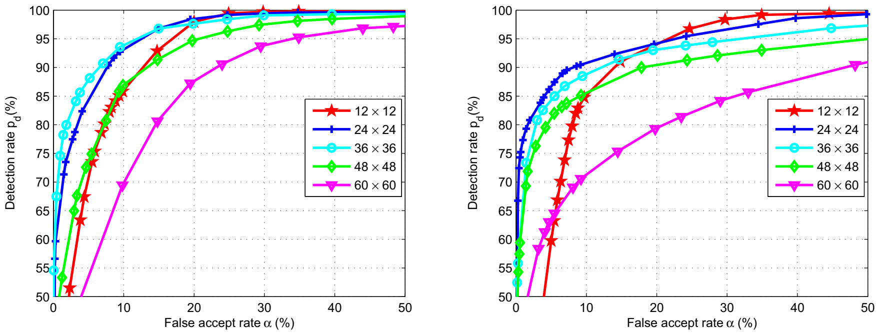
(a) $L = 12$, d from 144 to 48;

(b) $L = 24$, d from 576 to 192;

(c) $L = 36$, d from 1,296 to 432;

(d) $L = 48$, d from 2,304 to 768.

Figure 3.14: Comparison of the verification performance before and after applying the ROI. Four different scales are tested, d is the dimensionality of the feature vector.



(a) (b)

Figure 3.15: Comparison of the ROCs, using face image of different scales. Two preprocessing methods are used: (a)zero-mean and unit-variance normalization, (b) histogram equalization.

full dimensionality. The simple zero-mean and unit-variance normalization is applied on the cross-session images as a preprocessing.

The selected ROI should contain the major and consistent facial features in the face. Under such criterion, the selected ROI includes the eyebrows, eyes, nose, but excludes the forehead, mouth, and temple, as shown in Fig. 3.13. The face region within this ROI are less influenced by the expressions of the user, as mouth is most sensitive to the expressions. Moreover, the ROI selection as indicated in Fig. 3.13 reduces the length of the feature vectors by $\frac{2}{3}$. This makes the training much easier. In this way, the ROI largely reduces the dimensionality of the feature vector, and at the same time throws away the uninformative and noisy components of the face image. The specified ROI in Fig. 3.13 is empirically chosen.

To validate the ROI selection, we also show the performance comparison of the face verification before and after the ROI selection. We tested 4 scales: 12, 24, 36, 48, of the originally registered face image. Fig. 3.14 shows the verification performance before and after applying the ROI on the originally registered image square. Obvious improvement can be observed in all the four cases. Moreover, the improvement increases with the increase of the dimensionality. This indicates that a selection of the ROI is necessary and beneficial for the face verification.

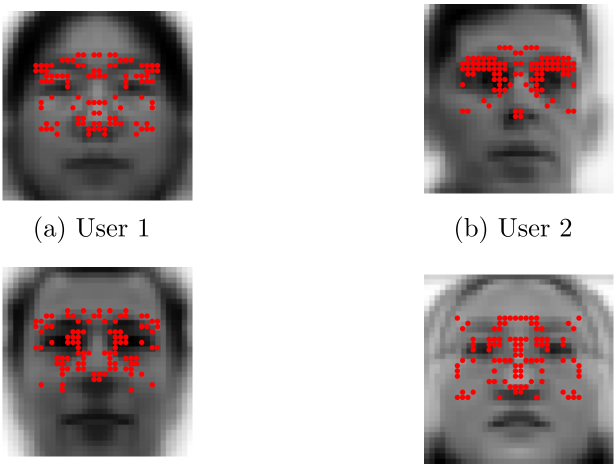
To investigate the impact of the image sizes on the verification performance, we further calculated the ROCs of 5 different scales, 12 × 12, 24 × 24, 36 × 36, 48 × 48, and 60 × 60. Note these are the original face sizes, and the image in the ROI is taken as the feature vector. For this purpose, we applied two preprocessing methods to remove the cross-session variations. Similar trend of the scale influence on the verification performance can be observed from Fig. 3.15 (a), in which zero-mean and unit-variance normalization is applied, and from Fig. 3.15 (b), in which histogram equalization is applied. In both cases, with the increase of the scale from 12 to 60, the performance firstly improves and then drops. This can be explained by the fact that images on a scale that is too small do not contain enough discriminative information for verification, while images on a scale that is too large put forwards high requirements on the training, which is difficult to be satisfied. In both image, it is suggested that the scale in the range of 24 and 36 should be a good choice. This coincides with the observation in [14] which suggests an optimal scale of 32. It can be further observed that the cross session tests do not have satisfactory performance in general, with the EER higher than 5%, even in good cases. In Fig. 3.15 (a), the zero-mean and unit-variance normalization yields better performance than the histogram equalization method in Fig. 3.15 (b), but still not good enough. This implies that the cross-session variation, which is primarily caused by the illumination differences, has a large influence on the verification performances.

Feature Selection

The dimensionality can be further reduced by feature selection, i.e., select another discrete ROI by the information theoretic criterion, i.e., mutual information, as introduced in Section 3.4.2. The feature selection is within the ROI defined in Fig. 3.13. The feature selection method is applied for each user, and results in user-specific feature selections. Fig. 3.16 shows the selected first 100 feature locations for the four different users, projected on the user average face image, forming a mask-like covering.

It can be observed that for different users, the selected feature locations are different. This is explained by the fact that different users have different face textures, which means that the distributions of the pixel values, if seen as random variables, in the face are user-specific. For example, for the user 2 and 3, in Fig. 3.16 (b) and (c), the important features are distributed around the eye region, while for the user 1 and 4, in Fig. 3.16 (a) and (d), the important features are more uniformly distributed in the ROI.

Unlike many other optimization methods that use the criterion of classifica-



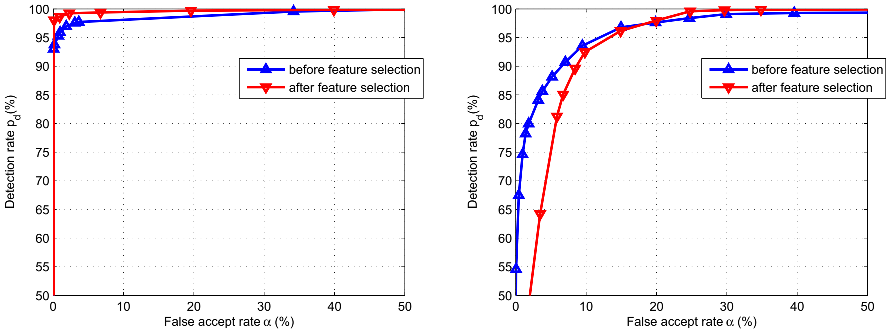
(a) User 1

(b) User 2

(c) User 3

(d) User 4

Figure 3.16: The first 100 selected feature locations, projected on the user average face image. The face size is 36 × 36.



(a)

(b)

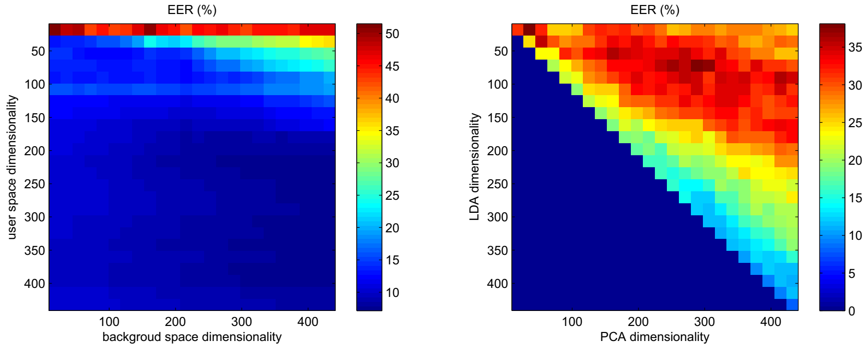
Figure 3.17: Comparison of the ROCs before and after the feature selection: (a) within-session test, (b) cross-session test.

tion performance or errors, the criterion of our feature selection is the mutual information, so it is interesting to check whether the performance is indeed improved after feature selection. Fig. 3.17 shows the verification performance before and after applying the feature selection. We used the zero-mean and unit-variance normalization for the preprocessing. Fig. 3.17 (a) shows the ROCs in the within-session test, and Fig. 3.17 (b) shows the ROCs in the cross-session test. In the within-session test, the performance is improved after feature selection, however in the cross-session test, the performance is degraded to some extent. This can be explained by the overtraining effect of the feature selection, which means that the selected features are not only user-specific, but also session-specific. In other words, the difference of the cross-session data is such that it changes the distributions of the random variables, and makes the feature selection based on the distribution information overfitted to the data of the training session. Besides, the feature selection procedure is dependent on the distribution estimation, which is by itself sensitive to the training.

Subspace Method

We also evaluated the personal subspace methods for dimensionality reduction as discussed in Section 3.4.3. To find the best dimensionality, we firstly applied the PCA and LDA in its linear version, i.e. without applying a non-linear kernel. To give a comprehensive view of the influence of the dimensionality on the verification performances, we calculate the error map with respect to different combination of dimensionalities. The performance measure we used here is the EER. In Fig. 3.18 (a) we show the error map of the personal subspace PCA, in which different dimensionalities are applied on the user space and the background space separately and independently. To obtain this map with respect to varying dimensionalities, we used a dimensionality step of 18. From dimensionality of 18 to 432, we calculated the EER of $24 \times 24 = 576$ combinations of different user space and background space dimensionalities. It can be observed that in a large range of dimensionality combinations, the verification has a performance on more or less constant level. The best performance, however, can be found when the user space is represented in a more compact way than the background space, e.g., the dimensionality of the user space is 200, and the dimensionality of the background space is 400.

In Fig. 3.18 (b) we show the error map of the personal subspace LDA, in which different dimensionalities are applied for the two diagonalizaion steps, as called PCA dimensionality, d_{PCA}, and LDA dimensionality, d_{LDA}, $d_{\mathrm{LDA}} \leq d_{\mathrm{PCA}}$, in the map. Note the LDA dimensionality is always smaller than the PCA



(a) (b)

Figure 3.18: (a) The error map with respect to different user space dimensionality and background space dimensionality. (b)The error map with respect to different PCA and LDA dimensions.

dimensionality, therefore only the upper right triangle of the map contains valid information, with in total 300 different combinations of the two dimensionalities. At each d_{PCA}, it is observed that the verification performance improves with the increase of d_{LDA}. As the error map is with respect to the cross-session test, this implies that the LDA dimensionality reduction based on simultaneously manipulating the user training data and the background data is sensitive to the training session. Furthermore, the verification performance globally improves with the increase of d_{PCA}, which implies that the higher dimensionalities are beneficial for representation of the two classes.

We have also implemented the personal subspace PCA and LDA in their kernel version. Unfortunately, the nonlinear subspace methods cannot improve the verification performance further under our test protocol. A benefit of nonlinear subspace analysis, nevertheless, is that with relatively a small number of samples, e.g. 100 user training samples and 1,000 background training samples, it can achieve better performance than the linear subspace analysis at the same situation. However, the performance still cannot compare to that of the linear subspace methods using sufficient samples as discussed above. Besides, with the increase of the training samples, the calculation becomes unaffordable, as every sample implies one time calculation of the kernel function between the input vector and itself. For example, to learn a good background class, we have

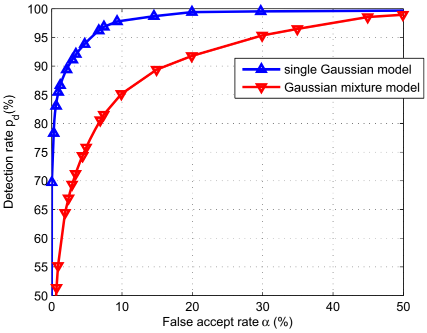


Figure 3.19: The ROC curves of the single Gaussian model and the Gaussian mixture model for calculating the LLR.

used the public database with more than 10,000 samples, and this will make the implementation of the kernel methods on the MPD impossible.

Probability Density Estimation

We also evaluated two probability density models: the single Gaussian model and the Gaussian mixture model, as introduced in Section 3.3.2 and Section 3.3.3. We used the Gaussian mixture algorithm in [51]. The Gaussian mixture model gives a more detailed description of the probability density, and is able to achieve lower classification error in the within-session test. However, the cross-session tests are of more interest in our application, as the enrollment phase and the testing phase are clearly separated in reality. Fig. 3.19 shows the ROC of the cross-session test, as described in the test protocol in Section 3.5.2. For this calculation we used the zero-mean and unit-variance normalization method, and the full dimensionality of the feature vector. A sharp decrease of the performance can be observed in the cross-session test.

So far it has been noticed that due to the large variability of the cross-session data, we must be very cautious with the dimensionality reduction, as well as with the density estimation. If the model in either step is too complicated, overtraining effects will occur. Normalization of the cross-session variability, which is largely caused by illumination, will help to reduce the speciality of the training data. Nevertheless, it is still safer to use a simpler model in order to increase the generalization capability of the system, as long as the discrimination capability of the system is sufficient. For this concern, as well as for the concern of computational complexity, we will use the single Gaussian model for the density estimation. The discrimination capability of the face verification system will be discussed in the next chapter.

3.6 Summary

This chapter discusses the face verification problem. Although similar to the face detection problem in the sense that both are two-class classification problems, the verification is more difficult a problem to solve, as it has larger intra-class variation and less inter-class variation. In the detection problem, the face class and the non-face class are two classes distributed with considerable margins, i.e., more separate, therefore, a detector with good performance is theoretically achievable. The emphasis of detection, instead, is to a large extent on the efficiency and the speed of the detection algorithm, for detection means enormous candidates to be classified. In the verification problem, in contrast, the user face class and the non-user class are much closer in distribution. A margin based classifier that works well enough for the detection, like the SVM which relies explicitly on the support vectors, or the Viola-Jones Adaboost method which relies implicitly on the highly-weighted samples, is not adequate for verification. Instead, the two classes are more accurately modeled as two overlapping classes with one encompassing the other, as shown in Fig. 3.5. To achieve the theoretically best performance, we used the likelihood ratio, an optimal measure in the Neyman-Pearson sense, as the verification rule.

Dimensionality reduction is an important step before estimating the likelihood ratio. It largely reduces the risk of curse of dimensionality. We have studied various methods for dimensionality reduction, including rescaling of the image, taking the ROI, feature selection based on information theoretic measure, and linear and nonlinear subspace methods. With our experiments on the cross-session MPD data, we have observed that the scale and ROI of the face image have large influence on the verification performances, while feature selection and subspace methods are less influential, sometimes even leading to overtraining of the system, i.e., the feature reduction trained on one session data does not work well enough for the data of another independent session.

Another relevant problem of the likelihood ratio based verification is the estimation method of the likelihood ratios. We have evaluated the single Gaussian model and Gaussian mixture model for both of the two classes. We showed that the single Gaussian model is not only much faster, but also more robust and reliable than the Gaussian mixture models. With this model, the likelihood ratio can be simply reduced to the difference between the Mahalanobis distances in the two classes.

In contrast to many other face applications which involves multiple users and have more or less even distribution of classes, one characteristic of the verification problem is that the two classes are largely unbalanced: a small user-specific class against a large background class. This brings high requirements on the generalization between the face data collected under different situations. This characteristic will be revisited in the following chapters, and guide our study on the illumination normalization problem in Chapter 5, and the information fusion problem in Chapter 6.

Chapter 4

Illumination Normalization

4.1 Introduction

[1]The variability on the face images brought by illumination changes is one of the biggest obstacles for reliable and robust face verification. It has been suggested that the variability caused by illumination changes easily exceeds the variability caused by identity changes [111]. As an example, Fig. 4.1 shows two face images of the same subject under two distinct illuminations. This observation has been widely acknowledged in the face recognition society. Basically, illumination easily alters the the distributions and amplitudes of the pixel values on a face image, which in turn changes the extracted feature vectors. Consequently, the face classifiers working on feature vectors are highly sensitive to the illumination. Illumination normalization, therefore, is a very important component of a face recognition system.

An illumination invariant representation of the face is most desirable for any face interpretation task. However, such a representation is difficult to achieve. There has been intensive study on this topic in literature, which can be categorized into two groups. The first category of methods try to study the illumination problem in a fundamental way, by building up the physics imaging model and the three-dimensional face surface model. This category we call *three-dimensional methods*. The second category of methods, however, do not rely on recovering the full three-dimensional information, instead, they work directly on the two-dimensional image pixel values. This category we call *two-dimensional*

[1] This Chapter is based on the publication [159], [164].

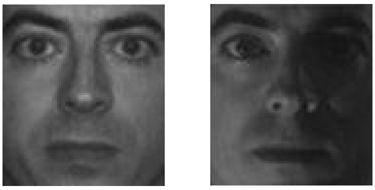


Figure 4.1: Face images of the same subject under two distinct illuminations. Examples are taken from the Yale B database [56].

methods. In this chapter, we will firstly present a review of two categories of methods, analyze their advantages and disadvantages, and propose the solutions that are most suitable for our specific face verification problem on a mobile personal device.

The remainder of this chapter is organized as follows. Section 4.2 reviews the illumination normalization methods in the two aforementioned categories. Section 4.3 and Section 4.4 proposes two solutions to our specific problem on the MPD, namely, directional sensitive horizontal Gaussian derivative filters and directional insensitive local binary patterns. Section 4.5 examines our proposed solutions under the verification framework introduced in Chapter 3. Section 4.6 presents the experimental results of the proposed illumination normalization methods. Section 4.7 summarizes this chapter.

4.2 Review of the Illumination Normalization Methods

4.2.1 Three-Dimensional Methods

Illumination on faces is essentially a three-dimensional problem. Three-dimensional illumination normalization methods aim to solve the problem on the two-dimensional images from the three-dimensional point of view. Most of these methods share the same basic physical model, as in [146], [147] [8], [193], [5], [149] etc, assuming Lambertian reflectance on the object surface. In general, inhomogeneous surfaces are dominantly Lambertian, except for isolated regions that are specularly reflecting light [146]. The Lambertian reflectance model is expressed as

$$I(x,y) = \rho(x,y)\, n^{\mathrm{T}}(x,y)\, s, \tag{4.1}$$

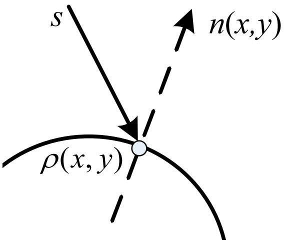


Figure 4.2: The Lambertian reflectance model, where $\rho(x, y)$ is the albedo at the point (x, y), $n(x, y)$ is the surface normal, and s is the light source.

In case of shadow caused by angles larger than 90°,

$$I(x, y) = \max\left\{\rho(x, y)\, n^{\mathrm{T}}(x, y)\, s, \ \ 0\right\}. \tag{4.2}$$

In both equations, (x, y) are the coordinates of the image point, $I(x, y)$ is the corresponding image pixel value, $\rho(x, y) \in \mathbb{R}$ is the albedo at this point, $n(x, y) \in \mathbb{R}^{3\times1}$ is the surface normal, and $s \in \mathbb{R}^{3\times1}$ is the light source, containing both the direction and intensity information. Fig. 4.2 gives an illustration of this reflectance model. By approaching the problem in the three-dimensional domain, it is assumed that the effects of s can be decoupled in either an explicit or inexplicit manner.

Linear Subspace

In the early work of Shashua [146], it was proposed that the images of a stationary object lie in a three-dimensional Euclidean space, and can be represented as linear combinations of a set of 3 images of the object. A simple proof is given as follows.

Proof. Let $m(x, y) = \rho(x, y)n^{\mathrm{T}} \in \mathbb{R}^{1\times3}$. For simplicity, we will omit the coordinate (x, y) in the following proof. The image I obtained under some illumination s is, according to (4.1)

$$I = ms$$

Any 3-dimensional light vector s can be decomposed into 3 linear independent basis, satisfying the condition that the rank of matrix $(s_1\ s_2\ s_3)$ is 3,

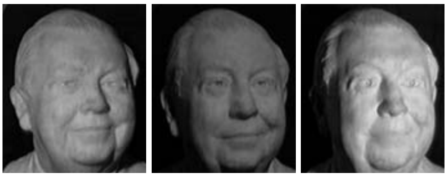


Figure 4.3: The 3 basis images of an object, taken under 3 different illuminations, from [146].

expressed as

$$s = \alpha_1 s_1 + \alpha_2 s_2 + \alpha_3 s_3$$

Therefore,

$$\begin{aligned} I &= m(\alpha_1 s_1 + \alpha_2 s_2 + \alpha_3 s_3) \\ &= \alpha_1 (m s_1) + \alpha_2 (m s_2) + \alpha_3 (m s_3) \\ &= \alpha_1 I_1 + \alpha_2 I_2 + \alpha_3 I_3 \end{aligned}$$

which means that images taken under an arbitrary illumination can be decomposed to the weighted sum of the other 3 basis images, taken under 3 basis illuminations. □

Given three basis images of the subject taken under three linearly independent illuminations, see Fig. 4.3 for an example, the coefficients α_1, α_2, and α_3 can be solved for any input image I by formulating a least square problem

$$\alpha = \arg\min_{\alpha} |S\alpha - I|^2$$

where $\alpha = [\alpha_1\ \alpha_2\ \alpha_3]^{\mathrm{T}}$, $S = [s_1\ s_2\ s_3]$. Consequently, a distance from the input image to the spanned space by the basis images can be calculated, and used for classification of the input image, in the same way as the "faceness" measure in [167].

Illumination Cone

Illumination cone is an extension of the linear subspace of the Lambertian objects [8]. It is shown that the set of images of an object seen under arbitrary lighting conditions is a *convex cone* that lies in a low-dimensional subspace of

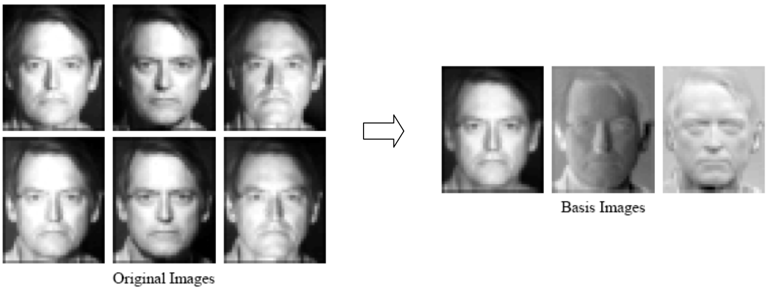


Figure 4.4: Left: the original images used to construct the illumination subspace of the subject. Right: the three basis images that span the illumination subspace of the subject [8].

the image space. Using the Lambertian model in (4.2), the set of all possible images of a subject, created by varying the light sources, is constructed

$$\mathcal{I} = \left\{ I : \; I = \sum_{i=1}^{k} \max(Ms_i, 0) \right\}$$

where $M \in \mathbb{R}^{N \times 3}$ is a matrix containing both the surface normal and albedo information of all the pixels in the face image (N is the number of pixels in image). When $Ms_i < 0$, the value 0 is taken as the shadow area.

For the face recognition purpose, an illumination cone is built for each subject. The illumination cone is derived using a photometric stereo algorithm. The intrinsic information of the subject's three-dimensional mode, M, is estimated by recursively minimizing $|\mathbb{I} - M^* S|$, where $\mathbb{I}$ is a matrix representing the training images, and S is the matrix representing different lighting conditions. Given M, a dimensionality reduction through SVD is performed so as to represent the illumination cone in a more compact way.

The central argument of [8] is that the set of images of an object under all possible illumination conditions formed a convex cone in the image space, and that this illumination cone can be constructed from as few as 3 images, see Fig. 4.4 for an example. The distance between the input image and the same image represented by the illumination cone is calculated for the final classification.

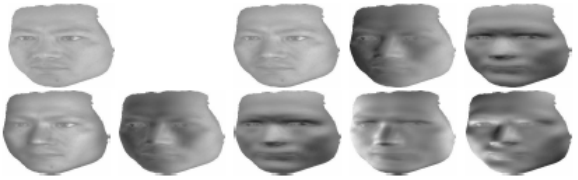


Figure 4.5: The calculated first 9 harmonic images for a subject, from [5].

Spherical Harmonics

The spherical harmonics method introduces a sphere with unit albedo for modeling the reflectance functions [5]. It is proved that the set of all reflectance functions, i.e., the mapping from surface normals to intensities, produced by Lambertian objects lies close to a 9-dimensional linear subspace. This implies that the set of images of a Lambertian object obtained under a wide range of lighting conditions can be approximated by a low-dimensional subspace.

In this work, the authors modeled the reflectance function of a sphere with unit albedo under isotropic lighting conditions using spherical harmonics representations. It is observed that 99% of the reflectance function energy resides in the first few harmonics. The motivation to model the reflectance function of a sphere is, each point on the face object can inherit its intensity from the point on the sphere which share the same normal n. Given the reflectance function $r(x, y, z)$, the image pixel at location p, corresponding to the point on the face object with the albedo ρ and surface normal $n = [n_x, n_y, n_z]^{\mathrm{T}}$, has the value

$$I(p) = \rho\, r(n_x, n_y, n_z)$$

Using the first 9 spherical harmonics as the base for the reflectance function $r(x, y, z)$, the set of images of an object can be approximated by a linear subspace, using the harmonics images, which are images obtained using the basis of the reflectance functions, as shown by Fig. 4.5. Face recognition is done in the same way as the previous method, by calculating the distances between the input image and the same image reconstructed in the harmonics subspace.

Quotient Image

In [147], Shashua et al. proposed a *quotient image* method to deal with the illuminations problem. The authors show that the set of all images, generated

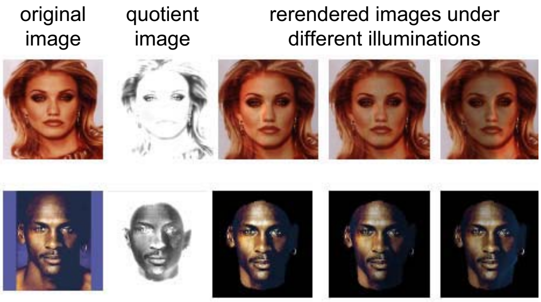


Figure 4.6: Examples of the quotient images, from [147]. The first column is the original face image, the second row is the illumination-invariant quotient image, and the last 3 rows are the rerendered image under different illuminations via the quotient image.

by varying the lighting conditions on a collection of Lambertian objects in a *class*, defined as the objects of similar shape (e.g. human faces are taken as a class), can be characterized analytically using images of a prototype object in this class, and an illumination-invariant quotient image of the object in this class. The obtained quotient image, or the image rerendered under the uniform frontal illumination, can be used as the illumination-normalized output for the subsequent face recognition.

It is important to note that the authors defined the *class* as the collection of subjects, which share the same three-dimensional shape, i.e., the surface normals n in (4.2), but differ in the surface albedos ρ. The quotient image is defined as

$$Q_x(p) = \frac{\rho_x(p)}{\rho_a(p)}$$

where Q_x is the quotient image of subject x, p is the range of the image, ρ_x is the albedos of the subject x, and ρ_a is the albedos of the training subject a in the same class. Under the constant shape assumption, the quotient image is derived by taking the ratio between the input image and the image of the training subject a, taken under the same illumination

$$Q_x(p) = \frac{\rho_x(p)}{\rho_a(p)} = \frac{\rho_x(p) n_x(p) s}{\rho_a(p) n_a(p) s} = \frac{I_x(p)}{I_a(p)} = \frac{I_x(p)}{\sum_{i=1}^{k} \alpha_i I_{a,i}(p)} \tag{4.3}$$

In this equation, given the input image $I_x(p)$, and the training images of subject a taken under a set of different illuminations $I_{a,i}(p)$, $i = 1, ..., k$, the task is to find the coefficients $\alpha_1, ..., \alpha_k$ that is able to construct the image of subject a taken under the same illumination as I_x. This is feasible given the linear subspace theory of [146], [8], and [5]. The coefficients are solved in closed form by the least square method. Details are found in [147].

Once the quotient image of the subject x is obtained, the images of the same subject taken under different illuminations can be rendered

$$I'_x(p) = \left(\sum_i \beta_i I_a(p) \right) \otimes Q_x(p) \tag{4.4}$$

where β_i's are some arbitrary coefficients, $\otimes$ denotes the Cartesian product, i.e., per pixel multiplication. Fig. 4.6 shows two examples of the quotient image, and the rerendered images under different illuminations.

For face recognition purpose, the illumination-invariant quotient image, or the image rerendered under the uniform frontal illumination, can be used as the illumination-normalized face pattern.

Shape from Shading

In [149], a novel method is proposed for solving the *shape from shading* problem [66] within the restricted class of human faces. Different from the quotient image model which assumes constant surface normals of the face class, the method does not have such restrictions, instead, it estimates the surface normals from a statistical point of view. Besides, the method uses an augmented Lambertian model

$$I = \rho\, n^{\mathrm{T}}\, s + e = ms + e \tag{4.5}$$

where e is the error term to model shadows and specular reflections that are neglected in the original model. From a bootstrap set, in [149] the Yale B dataset [56], which contains images of different subjects taken under predefined illumination settings $s_1, ..., s_N$, the mean and covariance of the term e at the predefined illuminations can be estimated by calculating the least-square solutions of m and hence $e(s_i)$ at each s_i, $i = 1, ..., N$. Note Gaussian models are used for the error term s.

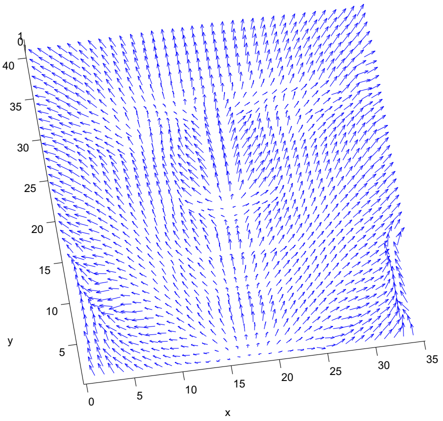


Figure 4.7: The needle map that shows the estimated surface normals as little arrows, from [149].

Given an input image with unknown illuminations, albedos, and surface normals, the illumination s is firstly estimated as a shape-from-shading problem, using the kernel regression method [149]. Once s is obtained, the mean and covariance of the error term $e(s)$ is estimated from $m(s_i)$ and $e(s_i)$, $i = 1, ..., N.$, again using the kernel regression method. Consequently, the term m containing the albedo and surface normal information can be recovered by the maximum a posteriori estimation $\hat{m} = \arg\max_m p\,(m|I)$, and the surface normal is obtained as $n = \frac{m}{|m|}$. Fig. 4.7 shows an example of the recovered n. Finally, a illumination-normalized image is created by rendering uniform frontal lighting on m.

Summary

We have reviewed five representative three-dimensional methods dealing with the illumination problem on face. The first three methods, namely, linear subspace, illumination cone, and spherical harmonics, share the common idea that, the images of an object taken under different illuminations lie within a low-dimensional space, which is spanned by some basis images, as shown in Fig. 4.3, Fig. 4.4, and Fig. 4.5. Another common characteristic is that the three methods result in subject-specific linear subspaces, which require the images of

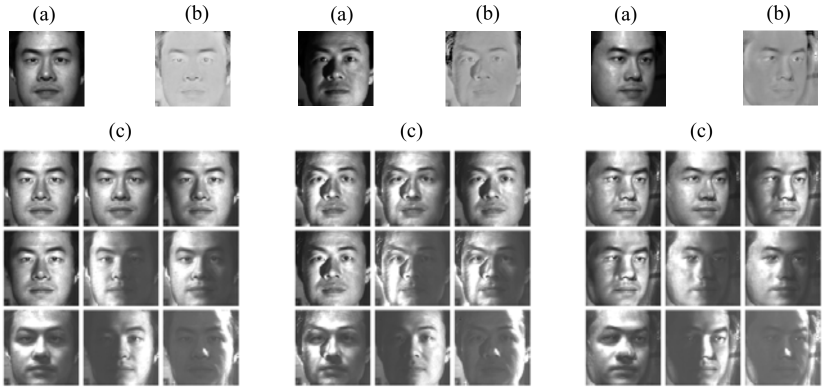


Figure 4.8: Examples of quotient images: left - good example when the shadow-free assumption and constant-shape assumption are well satisfied, middle - example when strong shadow exists, right - example when the shape are not well aligned. In all cases, (a) is the original image, (b) is the quotient image, (c) is the rerendering of the original image under different illuminations, indicating the accuracy of the quotient image.

the same subject taken under different illuminations as the training set.

The drawback of these methods is the over-simplified imaging model. First of all, the objects are assumed to be stationary, but in reality, the faces are subject to changes with respect to poses and expressions. In other words, the surface normals are not constant. Moreover, the methods do not explicitly deal with shadows and reflections. As a result, there are considerable image variations that cannot be accounted by the low-dimensional subspaces. Furthermore, the training set of the user under different illuminations are difficult to obtain, putting forward high requirements on the hardware device.

Instead of deriving a user-specific illumination space, the quotient image method and the shape from shading method aim to produce the illumination-normalized output, as a preprocessing method. Furthermore, the methods do not require the user face image to be acquired under different illuminations, instead, they rely on a bootstrap set that is already available. The drawback of them, nevertheless, is still the over-simplified physical models, for example, in quotient image method the fixed three-dimensional shape of the face class, and in shape from shading method the Gaussian model of the error term, and the kernel regression estimation of illumination-dependent parameters.

To summarize, the three-dimensional methods aim to recover the three-dimensional information, which is fundamental of a face, therefore, they can be expected to achieve good performance. However, as converting the three-dimensional objects to the two-dimensional images is a process with loss of information, the reverse process will unavoidably have restrictions, such as fixed surface normals, absence of shadow or specular reflections. In reality, however, such assumptions are often not true. The shadow-free face images are only available under frontal or near-frontal lighting conditions. The constant shape assumption is easily violated by slight pose changes or expressions. In [149], where the surface normals $\vec{n}$ are estimated in a MAP (*maximum a posteriori*) manner without constant shape assumptions, it is also reported that the algorithm can only achieve good performance under near-frontal illuminations.

As an example of the three-dimensional methods, we implemented the quotient image method [147], as illustrated in Fig. 4.8, showing the quotient images [147] under three situations. In each situation, (a) is the original image, (b) is the quotient image, and (c) presents the rerendered images from the quotient image as in (4.4). Fig. 4.8 gives some feeling how the shadows and surface normal variations harm the quotient image performance. It can be seen that shadows and mis-alignment of the surface normals cause significant artifacts, which can be more easily observed from the rerendered images in (c). Although the results are only shown for the quotient image method, such drawbacks exist in general for Lambertian model based three-dimensional methods which cannot effectively deal with shadows and shape variations.

4.2.2 Two-Dimensional Methods

Two-dimensional illumination normalization methods do not rely on recovering the fundamental three-dimensional information, instead, they work directly on the two-dimensional image pixel values. Basically, they are image preprocessing methods designed specially to remove the illumination effects. A simple example is the offset correction method, which tunes the dynamic range of the pixel values in an image to the predefined scope, or the zero-mean and unit-variation normalization of the values. In the following, we will review some popular two-dimensional illumination normalization methods.

Histogram Equalization

Histogram equalization transforms the distributions of the pixel values in such a way that a more or less uniformly distributed histogram is realized. This allows for areas of lower local contrast to gain a higher contrast without affecting the global contrast. Histogram equalization has been used in many face recognition systems [179] [88] [155] for simple preprocessing purposes.

Linear Filters and Homomorphic Filters

Linear high-pass filter is a direct illumination filter, based on the observation that illuminations often appear as low-pass effects on an image (in [5], a theoretical proof of this observation is also given), while the facial feature edges are intrinsically high-frequency. The high-pass filtering thus removes the extrinsic illumination effects while retaining the intrinsic image information.

The homographic filter [63] is a technique that acts as high-pass filtering on the transformed domain of the image. Instead of assuming the illumination effects as being addictive, like in the normal high-pass filters, the method assumes the illumination effects to be multiplicative. Therefore, the image is firstly transformed by applying the logarithmic operation, then the logarithmic image is high-pass filtered. The final output image is obtained by taking the exponentials of the homomorphically filtered image.

Retinex Method

The Retinex theory [94], from the term retina and cortex, aims to describe how the human visual system perceives the color and lightness of a natural scene. The general idea of the Retinex theory is that the perceived sensation is related to the relative brightness of the light, denoted by L, and the surface reflectance, denoted by R, invariant to the illumination. The model is expressed as

$$I(x, y) = L(x, y)\, R(x, y) \tag{4.6}$$

Note that different from the three-dimensional Lambertian model with the three-dimensional variables s and n as in (4.1), both L and R are scalar values at the location (x, y).

The retinex model assumes that the illumination effects is multiplicative on the object surface reflectance. In [79], a single scale retinex algorithm is proposed, in which the reflectance $L(x, y)$ is estimated by the ratio of the pixel value at (x, y) and the weighted average of the intensities in the neighborhood. For simplicity, the logarithm of R is taken

$$R(x,y) = \log \frac{I(x,y)}{I(x,y) * G(x,y;s)} \tag{4.7}$$

where $G(x,y;s)$ is the weighting function in the neighborhood with scale s. An extension of this single scale retinex algorithm is to take into consideration multiple scale retinex [78], by

$$R(x,y) = \sum_{s=s_{\min}}^{s_{\max}} \log \frac{I(x,y)}{I(x,y) * G(x,y;s)} \tag{4.8}$$

which is reported to give a more robust estimate of the reflectance.

Diffusion Methods

The diffusion processes is a dynamic process, originally observed from the physical processes, e.g. heat conduction. The diffusion relies on partial differential equations [23]. With the retinex model, the diffusion methods can be used to estimate the lighting field L. This is achieved by using the linear diffusion equation, also known as the heat conduction equation, with the original image as the initial conditions. The diffusion process is written as a function of the time index t

$$L_{t+1}(x,y) = I_t(x,y) + \frac{1}{N} \sum_{\omega=\Omega_1}^{\Omega_N} \nabla I_{\omega,t}(x,y) \tag{4.9}$$

where Ω_i, $i = 1, ..., N$, is the set of directions in which the diffusion is computed, $\nabla I_{\omega,t}(x,y)$ is the directional derivative in the direction ω at time t at location (x,y). Obviously, the diffusion step gradually blurs the image in the predefined directions.

Diffusion in all directions has the risk of removing meaningful edges. It is desired that the useful edge information is still preserved after the diffusion. For this purpose, the anisotropic diffusion is proposed [123], adding spatially-varying diffusion coefficients that treat different image contents in different ways. The diffusion coefficients are functions of the gradient in the image. The anisotropic diffusion is done in such a way that when the gradient is small, i.e., no prominent edges, the diffusion process in the respective direction is carried out, otherwise the diffusion process is attenuated to keep the edges. In this manner, the image noises are removed while the edges are preserved.

Local Binary Patterns

The Local binary patterns method was proposed in [117], and have proved to be useful in a variety of texture recognition tasks. The fundamental idea is simple: each 3×3 neighborhood block in the image is thresholded by the value of its center pixel. The eight thresholding results form a 8-bit binary sequence, representing the pattern at the center point. The thresholding is insensitive to illuminations, as it is only a relative measure in the local area. A decimal representation is obtained by taking the binary sequence as a decimal number between 0 and 255. Subsequently, a histogram of the LBPs on every image point is calculated, representing the distribution of 256 patterns in the face image. This is thus very suitable for texture recognition purposes, as the texture images have uniform patterns across the image.

For face recognition purposes, it is proposed in [2] that the face image is firstly partitioned into a set of subimages, and from each subimage a LBP histogram is obtained, representing the texture distributions at different locations of the face. The concatenated histogram is then used as the illumination-insensitive feature vectors for the classification purpose.

Summary

A close examination of the reviewed methods reveals that most of the two-dimensional illumination normalization methods are essentially linear or non-linear *high-pass* filters, emphasizing edges in the original face image. This makes sense as illumination effect often appear as low-frequency components in the face image, like the overall brightness changes, while the intrinsic facial features, such as eyebrows, lips, appear as high-frequency components of the image.

According to the strict physics model, however, the illumination cannot be simply seen as the low frequency component of the image, as it is a three-dimensional quantity, and has been modulated by the three-dimensional surface normals in a complicated way. Illumination also causes high frequency edges on the two-dimensional face image, most frequently around the nose area, as well as in other areas, see Fig. 4.1 or Fig. 4.8 for example. The edges caused by illumination can be very strong, as shadows and reflections often cause significant high-frequency components in the image. The biggest problem for the two-dimensional illumination normalization methods, therefore, is that the high frequency edges caused by illumination cannot be easily discriminated from the facial feature edges intrinsic to the face. If local methods are used, only local views are provided for the filters and all the edges are deemed equivalent. If global methods are used, a model must be built up to discriminate the two types of edges in the first place. Introducing such a model has the risk of bringing errors to the illumination-normalized result, as in the case of the three-dimensional methods.

Invariance to illumination is very desirable but cannot be easily achieved. For the three-dimensional methods it is in theory possible by recovering the lost information through extensive training and complicated calculation, but the cost is high. For the two-dimensional methods it is theoretically not possible, as stated in [25]: *for an object with Lambertian reflectance there are no discriminative functions that are invariant to illumination.*

In this thesis, therefore, we aim for simple and efficient two-dimensional methods that are *insensitive* to illumination. Without rigid restrictions on the input image as in the three-dimensional methods, the two-dimensional methods put forward lower requirements on image acquisition process and hardware devices. We will propose two methods, and show how insensitivity is achieved through the implementation of them [159] [164]. Furthermore, we will analyze the generalization capability and discrimination capability of the subsequent verification after applying the proposed illumination normalization methods, under the verification framework described in Chapter 3.

4.3 Illumination-Insensitive Filter I: Horizontal Gaussian Derivative Filters

4.3.1 Image Filters

Gabor, Gaussian, Laplacian, and Gaussian derivative filters are typical two-dimensional filters widely used in image processing and computer vision [176]. Each of them can emphasize certain type of image textures, and has different sensitivity to noise and illumination.

Gabor filter is a linear filter whose impulse response is defined by a harmonic function multiplied by a Gaussian function. It has been widely used due to its biological resemblance to the response of the human retina [36] [37] [38] [45] [17]. Gabor filter is expressed as

$$F_{\text{Gabor}}(x, y) = \cos\left(2\pi\frac{x'}{\lambda} + \psi\right)\exp\left(-\frac{x'^2 + \gamma y'^2}{2\sigma^2}\right) \tag{4.10}$$

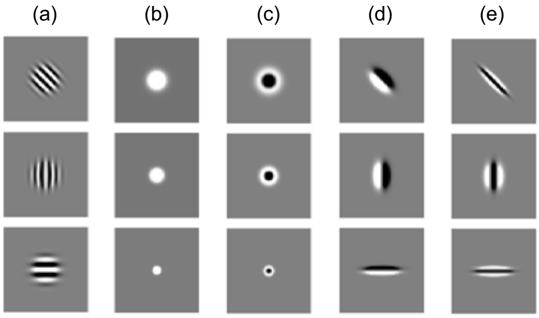


Figure 4.9: Examples of different two-dimensional image filters: (a) Gabor filters, (b) Gaussian filters, (c) Laplacian filters, (d) first-order Gaussian derivative filters, (e) second-order Gaussian derivative filters. Each colum shows the filters of the same category but with different parameters.

where

$$x' = x\cos\theta + y\sin\theta, \qquad y' = -x\sin\theta + y\cos\theta \tag{4.11}$$

In this expression, x and y are the self-variables in the two-dimensional space, and the other five parameters characterize the Gabor filter: θ - the direction of the filter, σ - the width of the filter, γ - the aspect ratio of the x and y axis, λ - the frequency of the filter, and ψ - the phase of the filter. The Gabor filter, therefore, is sensitive to the image textures of certain direction, width, aspect ratio, and frequency, as shown by Fig. 4.9 (a).

The two-dimensional Gaussian filter, as shown by Fig. 4.9 (b), is expressed as

$$F_{\text{Gaussian}}(x,y) = A\exp\left(-\frac{(x-x_0)^2}{2\sigma_x^2} - \frac{(y-y_0)^2}{2\sigma_y^2}\right) \tag{4.12}$$

where A is the amplitude, x_0 and y_0 is the center of the filter, σ_x and σ_y are the width of the x and y directions, respectively. The parameters σ_x and σ_y actually define the aspect ratio of the filters. The Gaussian filter is mostly used for smooth purposes in image processing [21], especially before applying high-pass filters, to given more robustness to image noise and to produce more reliable result.

The Laplacian filter is the Laplace operation of the Gaussian filter, expressed as

$$\begin{aligned} F_{\text{Laplacian}}(x, y) &= \nabla^2 F_{\text{Gaussian}}(x, y) \\ &= \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) F_{\text{Gaussian}}(x, y) \qquad (4.13) \end{aligned}$$

which is rotational symmetric, and sensitive to isolated dots of different sizes, as can be observed from Fig. 4.9 (c).

The first and second order Gaussian derivative filters are expressed as

$$F_{\text{d}_1}(x, y) = \frac{\partial}{\partial x'} F_{\text{Gaussian}}(x, y) \qquad (4.14)$$

$$F_{\text{d}_2}(x, y) = \frac{\partial^2}{\partial x'^2} F_{\text{Gaussian}}(x, y) \qquad (4.15)$$

in which

$$x' = x \cos\theta + y \sin\theta \qquad (4.16)$$

with θ indicating different directions of the derivative.

Of the Gaussian derivative filters, the first-order and second-order Gaussian derivative filters are most interesting for image processing, as shown in Fig. 4.9 (d) and (e), discriminative of image edges and bars in different directions and of different sizes.

In literature, very often the image filters are used in a collective way, forming filter bank that is capable of extracting different facial patterns. The concatenated response of the filters are used as the feature vector for the recognition purpose. A well-known example is the elastic bunch graph method [184], in which a set of 40 Gabor wavelets (5 frequencies × 8 orientations) are used to produce the jet output at the facial feature locations. In [18], the Gabor filter bank is also used to extract an augmented Gabor face feature vector for recognition. The risk of using the filter bank, however, is that it increases the dimensionality of the feature, and possibly introduces the curse of dimensionality problem in the training. Besides, some filters in the bank may not be useful, or even potentially introduce sensitivities to the illumination, due to the specific textures on which they are sensitive.

4.3.2 Directional Gaussian Derivative Filters

To investigate the sensitivity of the two-dimensional image filters to the illumination, we show the face image under the side light passing through a bank of

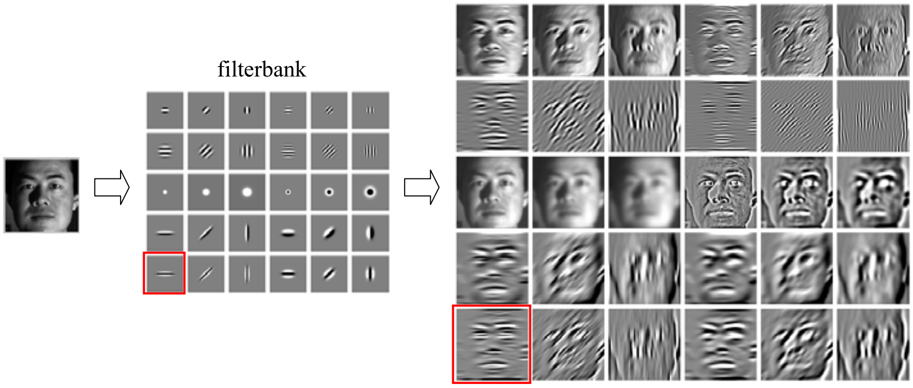


Figure 4.10: The original face image under the side light, filter bank, including Gabor filters (the first two rows), Gaussian filters (the first half of the third row), Laplacian filters (the last half of the third row), first-order Gaussian derivative filters (the fourth row), and second-order Gaussian derivative filters (the fifth row), and filtered face images.

different filters. As shown in Fig. 4.10, the filter bank contains the Gabor filters, Gaussian filters, Laplacian filters, and the first and second order Gaussian derivative filters, with different scales, orientations, and aspect ratios.

It can be observed from Fig. 4.10 that the Gabor filters emphasize textures of certain orientation, scale, and frequency; Gaussian and Laplacian filters are not directional, but are selective on the sizes of dots and circles; the first-order Gaussian derivative filters concentrate on edges of different sizes and directions; and the second-order Gaussian derivative filters concentrate on bars of different sizes and directions. As in the original image, the illumination creates edges mostly in the vertical direction, it can be seen that the illumination effects are less obvious in images filtered by the horizontal directional filters. For example, in the filtered image on the lower left corner, almost no indication of side lighting can be observed. In contrast, in the filtered image by the vertical filters, the high frequency components caused by the side lighting are more or less emphasized.

Interestingly enough, most of the important face textures, like eyebrows, eyes, mouth, except nose, are more in horizontal directions than in vertical directions. The nose is informative in the 3D sense, but in the 2D images, it is often sensitive to illuminations due to the directions of its surface normals. Moreover, the nose often causes shadows along its center line. Generally speak-

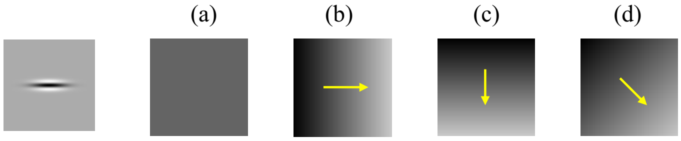


Figure 4.11: Four simple illumination patterns, (a) uniform intensity, (b)(c)(d) linearly increasing intensity, direction indicated by the arrowhead. The convolution result of the filter with these four simple illumination patterns are zero.

ing, the edges caused by illumination are more often in vertical directions than horizontal. This has inspired us to use the horizontal filters to make the image insensitive to illumination.

We select the second-order Gaussian derivative filter in the horizontal direction, as marked by the rectangle in Fig. 4.10, which is sensitive to bar textures with certain length and width. The reason of selecting the horizontal direction has been stated above, and the reason of selecting the specific shape is that the bar, with approximately the same size with the facial features, is more informative and robust than edges in the face image. Another good property of the selected filter is that all the columns of the two-dimensional patch are symmetric and sum up to zero, which make it invariant to certain types of illumination patterns. Fig. 4.11 shows four examples of the common illumination patterns. In other words, if in the imaging model, these illumination patterns are addictive, the linear property of two-dimensional linear filters can guarantee strict invariance to these patterns. For the illumination normalization purpose, this is a desirable property. If these illumination patterns are modeled as multiplicative, the homomorphic filtering in the logarithmic domain can be applied.

The selected horizontal Gaussian derivative filters is expressed as

$$F_{\mathrm{d}_2}(x,y) \quad = \quad \frac{\partial^2}{\partial y^2} F_{\mathrm{Gaussian}}(x,y) \tag{4.17}$$

The null space of $F_{\mathrm{d}_2}(x,y)$ can be expressed in a general form

$$P(x,y) = af(x) + by + c \tag{4.18}$$

where a, b, and c are arbitrary coefficients, and $f(x)$ is an arbitrary function of x. We show simple proof in the following.

Proof. Filter the pattern $P(x,y)$ with the proposed horizontal Gaussian derivative filter $F_{\mathrm{d}_2}(x,y)$, we have

$$\begin{aligned}
F_{\mathrm{d}_2}(x,y) * P(x,y) &= \frac{\partial^2}{\partial y^2} F_{\mathrm{Gaussian}}(x,y) * P(x,y) \\
&= F_{\mathrm{Gaussian}}(x,y) * \frac{\partial^2}{\partial y^2} P(x,y) \\
&= F_{\mathrm{Gaussian}}(x,y) * \frac{\partial^2}{\partial y^2} \left(af(x) + by + c\right) \\
&= F_{\mathrm{Gaussian}}(x,y) * 0 \\
&= 0
\end{aligned}$$

Suppose $I(x,y)$ is any input face image, then

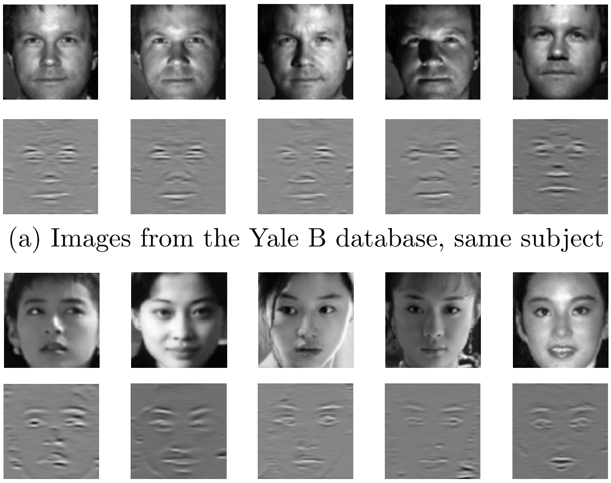
$$\begin{aligned}
F_{\mathrm{d}_2}(x,y) * (I(x,y) + P(x,y)) &= F_{\mathrm{d}_2}(x,y) * I(x,y) + F_{\mathrm{d}_2}(x,y) * P(x,y) \\
&= F_{\mathrm{d}_2}(x,y) * I(x,y)
\end{aligned}$$

meaning that the filter $F_{\mathrm{d}_2}(x,y)$ is invariant to any additive patterns $P(x,y)$. □

The proof also explains why the second-order derivative is more interesting than the first order derivative, as the second-order derivative allows a linear term in the null space of the filter, which can be modeled as the linearly increasing lighting, shown in Fig. 4.11.

Effective feature extraction is dependent on the size, i.e., σ_x and σ_y, of the Gaussian derivative filters. The size of the filter is selected in such a way that it can extract important facial texture information, but meanwhile filter out vertical edges and small-size noises. We estimate the average length and width of the following three important facial features: eyebrows, eyes, and mouth from the landmarked BioID database [171], and use them as the parameters of the Gaussian derivative function.

To illustrate the insensitivity of the proposed filters to illumination, we further show more examples in Fig. 4.12. In Fig. 4.12 (a), we take the face images of the same subject in the Yale B database [56], under different illuminations. Note that in these images, the lights are not only from the side direction, but also from the up and down directions. The filtered face images exhibit the horizontal face textures of the subject, which are insensitive to the different illuminations in the original images. In Fig. 4.12 (b), we take the Internet face images of



(a) Images from the Yale B database, same subject

(b) Real life images, different subjects

Figure 4.12: Examples of face images under different illumination and the filtered images. The filtered images are more insensitive to illuminations.

different subjects, under diverse illuminations. It can be observed again that the strong illumination influences in the original image is not prominent in the filtered images, while the different image textures of different subjects are still preserved.

4.4 Illumination-Insensitive Filter II: Simplified Local Binary Pattern

In the previous section, we have introduced the horizontal Gaussian derivative filter, which is insensitive to changes of the image textures caused by changes of the illuminations. Besides the image textures, the image intensities are also sensitive to illuminations. Strong illumination changes alter the image textures, while ordinary illumination changes mostly alter the image intensities. In order to achieve insensitivity to intensities, we propose to use the local binary patterns (LBP) as a nonlinear filter on the image values.

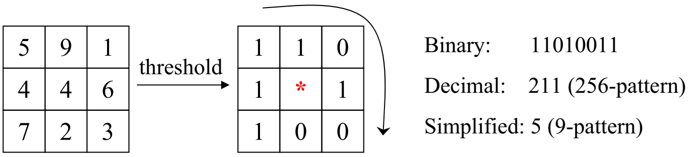


Figure 4.13: The LBP operator: the binary values, decimal value, and the simplified value.

4.4.1 Non-directional Local Binary Pattern

As has been introduced in Section 4.2.2, the LBP is firstly used for texture recognition [117]. The basic idea is illustrated in Fig. 4.13: each 3 × 3 neighborhood block in the image is thresholded by the value of its center pixel. The eight thresholding results form a binary sequence, representing the pattern at the center point. A decimal value is obtained by taking the binary sequence as a decimal number between 0 and 255, representing one of the 256 possible relative patterns. The distribution/histgram of the LBP patterns in the image is then used as the feature of the image.

To use the LBP histogram for face recognition, however, is not as meaningful as for texture recognition. The distribution of LBPs can be used as a good representation for images with more or less uniform textures, but for the face images it is not suitable. A distribution loses connection between the patterns and their relative positions in the face. To take advantage of both the local patterns and the positional information, LBP can be instead used as a filter on the image values. It has been proposed in [64] that the LBP is used as a preprocessing methods, using the LBP decimal values of the preprocessed pixel values.

The advantage of LBP is twofold. Firstly, it is a local measure, so the LBPs in a small region are not affected by the illumination conditions in other regions. Secondly it is a relative measure, and is therefore invariant to *any* monotonic transformation, such as shifting, scaling, or taking the logarithm, of the pixel values. For a pixel, LBP only accounts for its relative relationship with its neighbors, while discarding the information of amplitude. Using the LBP as a preprocessing method not only preserves the good property of the patterns, but also keeps the positional information of them on the face image.

Essentially LBP preprocessing acts as a nonlinear high-pass filter on the

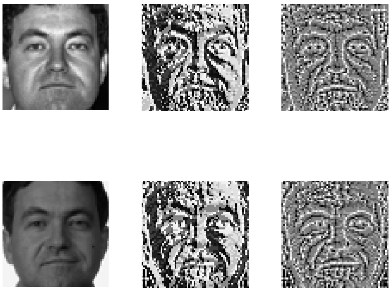


Figure 4.14: The effects of LBP preprocessing: first column - the original images under different illumination intensities; second column - the original LBP preprocessing; third column - the simplified LBP preprocessing. The face size is 64 by 64.

image values, according to the fact that it consists of the thresholding operation similar to differentiation. As a result, it emphasizes the edges in the image which contains significant change of the values, however, at the same time, it also emphasizes the noises in the image which contains negligible change of the values. Because noise occurs in a random manner as far as the direction is concerned, the exponential weights on the neighbors, from which the decimal value is calculated, subject the LBP values to considerable variabilities. To make the patterns more robust and insensitive to the noises occurring in random directions, we propose to assign equal weights to each of the 8 neighbors. The simplified LBP value is calculated by adding up all the 1's in the neighborhood, as shown in Fig. 4.13. In total the simplified LBP only has 9 possible values.

Fig. 4.14 shows the filtering effects of the original LBP and simplified LBP on two images with different illumination intensities. It is observed that the original LBP filtered image exhibits the directional sensitivity of the patterns, while the simplified LBP filtered image exhibits more robustness to such influence.

We further show more examples from the Yale B database [56] in Fig. 4.15. We compare the proposed simplified LBP filtering with the original LBP filtering. It is observed that the simplified LBP filtering produce more stable patterns under diverse illuminations, including extreme illuminations.

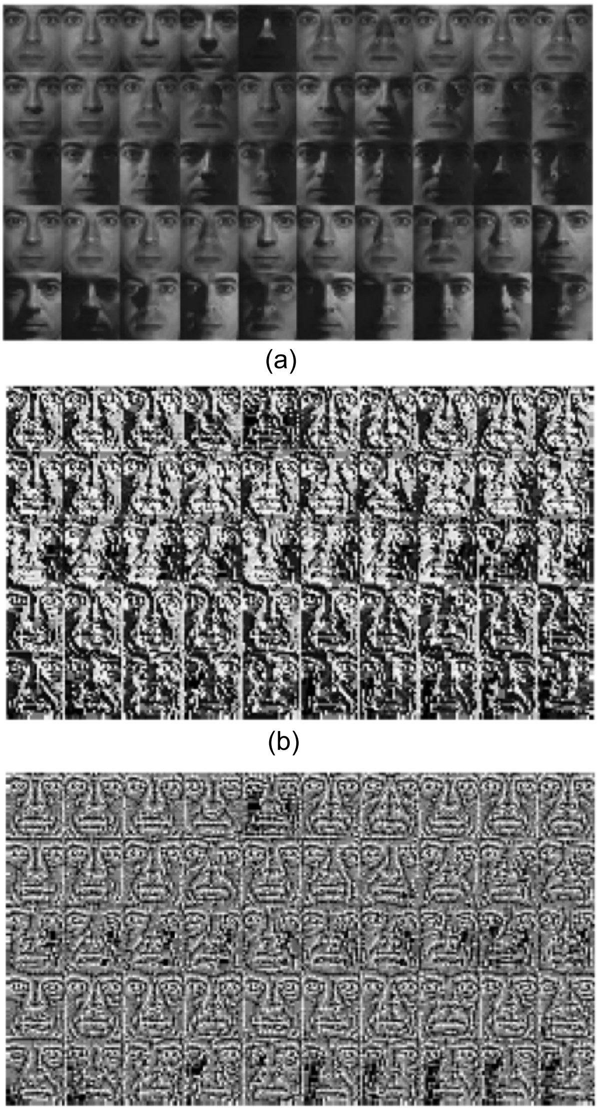


Figure 4.15: Examples in Yale B database: (a) the original face images, (b) preprocessed by the original LBP, (c) preprocessed by the simplified LBP.

4.4.2 Interpretation from a Lambertian Point of View

This simplified LBP preprocessing method can be physically interpreted if the Lambertian imaging model applies and if the lighting conditions are not extreme. According to the Lambertian model $I(x,y) = \rho(x,y)n(x,y)s$, we can express the difference between two neighboring pixels

$$\begin{aligned}\delta_I(x,y) &= [\rho(x+\delta_x, y+\delta_y)n(x+\delta_x, y+\delta_y) - \rho(x,y)n(x,y)]\, s \\ &= \Delta(x,y)s \qquad (4.19)\end{aligned}$$

where $\delta_x, \delta_y \in \{-1, 0, 1\}$, $(\delta_x, \delta_y) \neq (0,0)$ represent the position of the pixel to be thresholded with respect to the center point at (x,y), and $\Delta(x,y)$ is a three-dimensional vector related to the physical properties of the face at this point. According to the calculation of the LBP, only the sign of $\delta_I(x,y)$ has an influence on the final result, while its amplitude does not matter. Therefore, the question is essentially whether a change of s will alter the sign of $\delta_I(x,y)$.

We assume that the direction of the light source is more or less frontal, i.e., from above the image plane, as shown in Figure 4.16. This assumption avoids situations like strong back lighting or extreme side lighting, which will also be avoided in realistic scenarios. In the following, we will discuss (4.19) in three situations according to different characteristics of a face region. The term *invariance range* is defined as the range of all possible lighting directions of s, under which the sign of (4.19), i.e., the dot product of s and $\Delta(x,y)$, is the same.

- Constant ρ and n

  Examples of such area are the cheek, forehead, and chin regions. In this case, $\Delta(x,y)$ cannot be strictly 0, so its direction cannot be determined within a confined scope, due to the simultaneous small changes of both albedo and surface normal.

  When $\Delta(x,y)$ is orthogonal to the image plane, as shown by $n_1(x,y)$ in Fig. 4.16, the sign of (4.19) cannot be altered as the invariant range for $\Delta_1(x,y)$ covers the whole range of the front lighting. When $\Delta(x,y)$ changes from $\Delta_1(x,y)$ to $\Delta_2(x,y)$, the invariant range of lighting reduces, gradually excluding extreme lightings. Therefore, under non-extreme lightings, the sign of (4.19) is still unchanged. When $\Delta(x,y)$ is moving to be parallel to the image plane, the invariant range is gradually reduced to a half sphere, which is the minimum possible invariance range. In that case, the sign

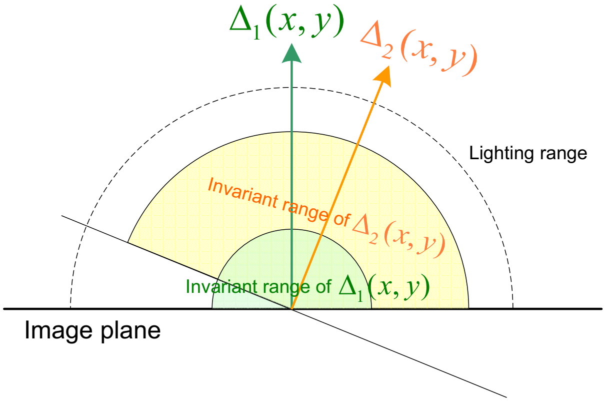

$\Delta_1(x,y)$
$\Delta_2(x,y)$
Lighting range
Invariant range of $\Delta_2(x,y)$
Invariant range of $\Delta_1(x,y)$
Image plane

Figure 4.16: Illustration of the invariant ranges for $\delta_I(x,y)$ of different directions $\Delta_1(x,y)$ and $\Delta_2(x,y)$. For lighting directions in this range, the sign of $\delta_I(x,y)$ is not altered. $\Delta_2(x,y)$ has a smaller invariant range than $\Delta_1(x,y)$, but as long as it is close to $n_1(x,y)$, the invariant range does not include extreme lighting directions.

of (4.19) will be altered by a change of direction of illumination s that crosses the half sphere, e.g. from the left-sided to right-sided, or from the up-sided to down-sided.

Such situation results in noisy effects in $\delta_I(x,y)$ with respect to one neighbor. Nevertheless, in smooth regions the 8 surrounding neighbors are constituted of 4 pairs of $\Delta(x,y)$ whose directions are nearly opposite. By adding them together, the noisy effect will be counteracted to an extent, and the result pattern will be still be relatively consistent even under this unfavorable situation.

- Constant ρ, changing n
he typical example is the nose region, where $\Delta(x,y)$ is approximated as

$$\Delta(x,y) = \rho(x,y)\,[n(x+\delta_x, y+\delta_y) - n(x,y)]$$

In this situation, only the change of normal direction matters. Note that in general, the change of the normal direction $n(x+\delta_x, y+\delta_y) - n(x,y)$ around the position (x,y) in a small neighborhood is approximately parallel to the surface at this position. As already discussed in the first situation, when the change of normal direction is nearly orthogonal to the image plane, for example, at the two side facade of the nose, the resulting pattern is robust. At the front facade of the nose the direction of $\Delta(x, y)$ is approximately parallel to the image plane, possibly introducing inconsistency of the patterns, under the changing illuminations that cross the half sphere. This explains the unsatisfactory performance of the LBP preprocessing at the center nose regions in Fig. 4.15. Moreover, in such regions, the simple Lambertian model that is used to derive $\delta_I(x, y)$ does not strictly apply due to the existence of shadows, which cause more instability in the results.

- Changing ρ, constant n
  Examples of such area are the eye, eyebrow, or mouth regions. In this case, $\Delta(x, y)$ is approximated as

  $$\Delta(x, y) = [\rho(x + \delta_x, y + \delta_y) - \rho(x, y)]\, n(x, y)$$

  This is the easy situation for the first term is a scalar constant, and only the direction of surface $n(x, y)$ has an effect on the sign. In these regions, as can be observed, the normals lie mostly within a small range of $n_(x, y)$. As shown in Fig. 4.16, for such normal directions, the lighting s has a large invariant range. Therefore, the resulting sign will be constant as long as the lighting is changing in this range.

  This type of the face region is the most informative part for face recognition. It can be observed from Fig. 4.15 that the LBP filtered image preserves such information in a consistent way under different illuminations.

In the above discussion, almost all the regions in the face are covered. From Fig. 4.15, it is observed that the a large part of the face textures, i.e., eyes, mouth, and nose edge, are extracted by the filter in a rather consistent way. Due to the high-pass filtering characteristic of the thresholding operation, however, the filtered images also exhibit some noisy phenomenon in other regions, like forehead, cheeks, and nose center. The subsequent likelihood-ratio based classifier will further act to reduce the effects of these random noises, as noises are easily modeled by the probabilistic model.

4.5 Illumination Normalization in Face Verification

The proposed horizontal Gaussian derivative filter and the simplified local binary pattern filter share a same characteristic: they discard part of the image information, which is sensitive to illumination. For the horizontal Gaussian filters, the texture information in the vertical direction are filtered out, and for the simplified LBP filters, not only the amplitude information, but also the directional information of the binary patterns, are thrown away. This leads to the suspicion that due to the loss of such information, the subsequent verification performance will possibly suffer. In this section, we will subject the two methods to the verification framework, and show that the proposed methods are able to gain stronger generalization capability, and at the same time keep the discrimination capability.

As has been introduced in Chapter 3, our face verification system is based on the likelihood ratio, calculated by

$$L(x) = \frac{p_{\text{user}}(x)}{p_{\text{bg}}(x)} \tag{4.20}$$

where x is the preprocessed holistic face feature, p_{user} is the user data distribution, and p_{bg} is the background distribution (including all the possible data). If the likelihood ratio $L(x)$ is larger than a certain value T, a decision of *accept* is made for the input x, otherwise a decision of *reject* is made.

As a holistic feature, the preprocessed face image is stacked into a feature vector x. A small enough face image, for example, with the size of 32×32, already has 1,024 pixels, which implies 1,024 degrees of freedom for the feature vector. The verification of a face image, therefore, is normally in a very high-dimensional space. High-dimensional space potentially has very large power of discrimination [166]. In the following, we will explain this with a simple example.

Suppose each of the user and the background class take up a hyper-sphere with radius r_{user} and $r_{\text{bg}} = a \cdot r_{\text{user}}$, $a > 1$, in a N dimensional space, as shown in Fig. 4.17. For a single dimension, the ratio of volume between the two spaces is

$$\frac{V_{\text{bg}}}{V_{\text{user}}} = a \tag{4.21}$$

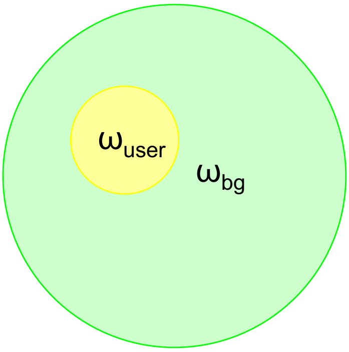


Figure 4.17: The distribution of the user data and the background data.

which means given an arbitrary point in the one-dimensional space, the chances that it belongs to the background class ω_{bg} is α times of the chance that it belongs to the user class ω_{user}.

From all the N dimensions, however, the ratio becomes

$$\frac{V_{\mathrm{bg}}}{V_{\mathrm{user}}} = a^N \tag{4.22}$$

when N is large, e.g., $N = 1000$, and a is moderate, e.g. $a = 1.5$, $\alpha^N = 1.5^{1000} \sim 10^{176}$ is almost infinite. This means that for an arbitrary N-dimensional feature vector, the chance that it falls into the user class ω_{user} is almost none. Therefore, the user face vectors, derived after illumination normalization of the face images taken under different illuminations, must lie within an extremely small area of the feature space, otherwise they get easily rejected.

In other words, the discrimination capability of such a likelihood-ratio classifier in the high-dimensional space is very high, whereas the generalization capability of it is very low. Generalization capability and discrimination capability are two equally important aspects in verification. For our MPD application, they are closely related to the convenience requirement and the security requirement, respectively. However, in the high-dimensional feature space, the prospects of these two aspects are imbalanced.

The problem, on the other hand, can be solved in such a way that in the illumination normalization stage, more emphasis is put on maintaining its generalization capability, rather than its discrimination capability. Consequently, the relative volume between ω_{bg} and ω_{user} is reduced, or equivalently, a is reduced. When a^N is not so prohibitively high, the generalization is basically much easier. This justifies the large reduction of the image information by our proposed methods, which makes both class, after illumination normalization, much smaller in volume. In comparison to the user class, the background class is more substantially reduced, as the methods discard much information that is useful for discriminating different subjects. At the same time, the discarded information is the illumination-sensitive part, which greatly increases the generalization capability to images of the user taken under different illuminations. Thanks to the high dimensionality, enough discrimination capability is preserved despite the loss of such information.

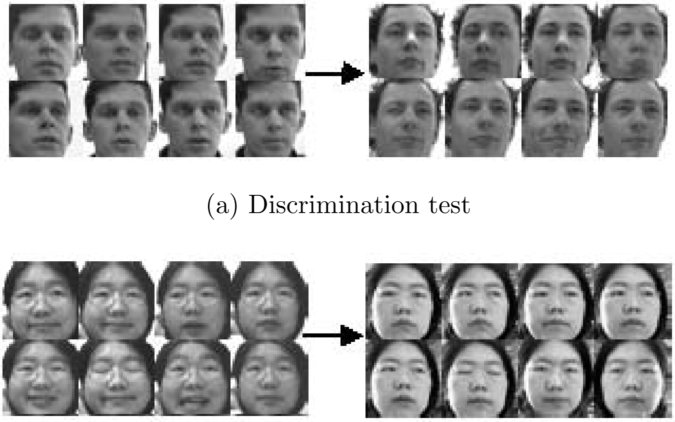
4.6 Experiments and Results

The experimental set up has already been described in Section 3.5. The test protocols remain the same, i.e., the illumination normalization methods are evaluated using the cross-session data. We used the face scale of 36 × 36, and used the full dimensionality in the ROI, i.e. 432, for better generalization. The Gaussian model is used to estimate the probability densities. As discussed in the previous section, the balance between the generalization capability and discrimination capability is important. For this reason, we distinguish between two types of test: discrimination and generalization. The first type is closely related to the security aspect of the system, and the second type is closely related to the convenience aspect [157]. Discrimination can be tested on different subjects under the same illumination, as shown in Fig. 4.18 (a), while generalization can be tested on the same subject under different illumination, as shown in Fig. 4.18 (b).

In Section 3.3.2, we have proved that the likelihood ratio can be expressed in the form

$$\ln L(x) = \frac{1}{2}\left(\underbrace{(x-\mu_{\text{bg}})^{\mathrm{T}}\Sigma_{\text{bg}}^{-1}(x-\mu_{\text{bg}})}_{d^2_{\text{Maha}}(x)\text{ in }\omega_{\text{bg}}} - \underbrace{(x-\mu_{\text{user}})^{\mathrm{T}}\Sigma_{\text{user}}^{-1}(x-\mu_{\text{user}})}_{d^2_{\text{Maha}}(x)\text{ in }\omega_{\text{user}}}\right) + c \tag{4.23}$$

where μ_{user}, μ_{bg}, Σ_{user}, Σ_{bg} are the means and covariances of the user class and background class, respectively. The second term c is a constant related to the



(a) Discrimination test

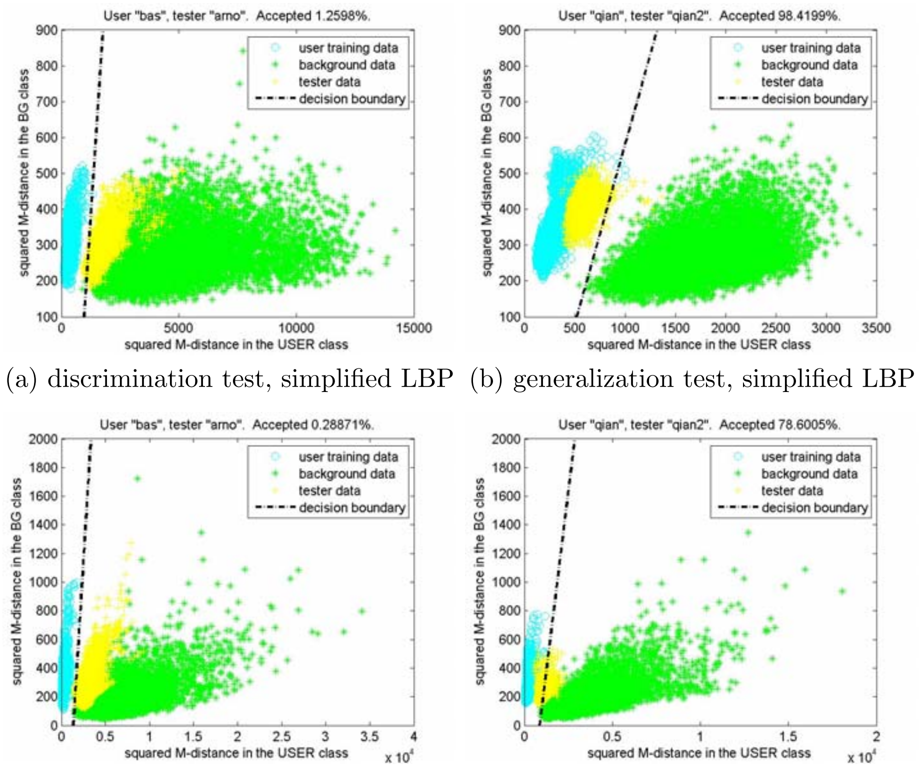
(b) Generalization test

Figure 4.18: Two types of test: discrimination and generalization.

covariances of the two classes, which can be absorbed into the thresholds of the likelihood ratio without influencing the final ROC measure. As (4.23) shows, the logarithm essentially reduces the probability measure to the difference between the two squared Mahalanobis distances in the user and the background class.

We can scatter the two squared Mahalanobis distances, as in (4.23), onto a two-dimensional scatterplot. Obviously, the decision boundary is a straight line with a slope of 1. As an example, Fig. 4.19 shows the results of these two tests, using the two proposed illumination normalization methods, respectively. We visualize the results in a two dimensional scatter plot, with the two dimensions indicating the squared Mahalanobis distances in the user and the background space, respectively.

In both figures, the circles $\circ$ denote the user training data, the stars $*$ denote the background data which are used in the training as the impostor data, the crosses $+$ denote the tester data, and the line denotes the decision boundary. Such a two-dimensional plot gives a clearer view of the distribution of the user data, background data, and impostor data. Furthermore, Fig. 4.19 also indicates that the likelihood ratio method involving two opposite classes is superior than the one-class method, e.g., the maximum likelihood ratio, which involves only the user class. This can be observed by comparing the distributions along



(a) discrimination test, simplified LBP (b) generalization test, simplified LBP

(c) discrimination test, horizontal filter (d) generalization test, horizontal filter

Figure 4.19: Scatter plots of the discrimination and generalization tests using the two illumination normalization methods.

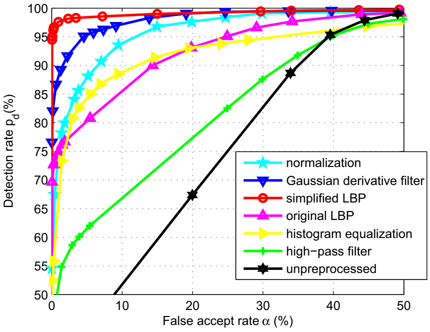


Figure 4.20: Comparison of the ROCs using different illumination normalization methods, from up down: simplified LBP, Gaussian derivative filter, zero-mean and unit-variance normalization, original LBP, histogram equalization, high-pass filtering, and the unpreprocessed.

the two-dimensional space and along the one-dimensional space (user class axis). Closer observation of (a) and (b) reveals that the user spaces for different users differ vastly, by comparing the same background data distribution in the different user spaces on the horizontal axis. Therefore this user-specific space described by μ_{user} and Σ_{user} is able to give a better description of the user, compared to a general intra-personal space sharing covariance among different users [108].

We draw a decision boundary based on the distribution of the user and impostor training data. The term c in (4.23) is calculated in the SVM-like way, from the points in both classes that are distributed nearest to the boundary. It can be observed, from the statistics on top of each figure, that the simplified LBP method has relatively higher generalization capability and lower discrimination capability compared to the horizontal filtering method. To give a more comprehensive view of the performance, we compute the ROCs of the verification performance using the two illumination normalization methods. As in Chapter 3, the test protocol remains the same.

Fig. 4.20 shows the comparison of the ROCs using different illumination normalization methods. As observed, the simplified LBP achieves the best performance, while the Gaussian second-order derivative filtering is also robust against the cross-session variations. The linear high-pass filtering, which is realized by subtracting the low-pass filtered image by rotationally symmetric Gaussian fil-

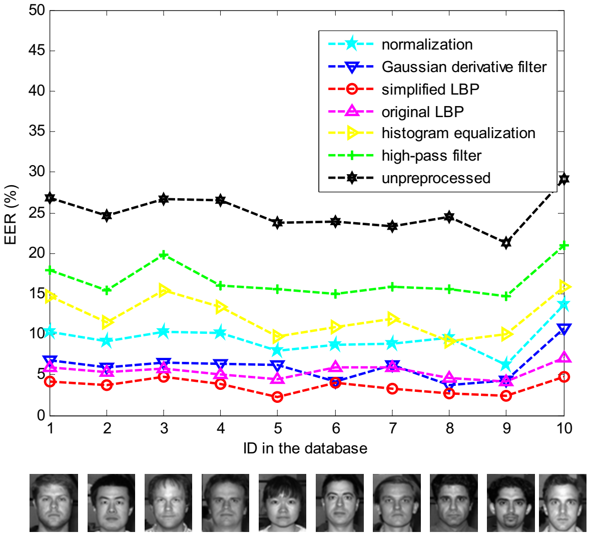


Figure 4.21: Comparison of the ROCs using different illumination normalization methods, from up down: simplified LBP, Gaussian derivative filter, zero-mean and unit-variance normalization, original LBP, histogram equalization, high-pass filtering, and the unpreprocessed.

ter from the original image, yields poor performance, indicating that direction of the filter is indeed important in illumination normalization.

The algorithm has also been tested on the Yale database B [56], which contains the images of 10 subject, each seen under 576 viewing conditions (9 poses × 64 illuminations). Examples of Yale database B and the effects of three preprocessing methods are shown In Fig. 4.15.

In our test, for each subject, the user data are randomly partitioned into 80% for training, and 20% for testing. The data of the other 9 subjects are used as the impostor data. To illustrate the performances of different illumination normalization methods, we used the EER as the performance measure. The random partition process is carried out 20 rounds for each subject. We obtain an average EER for each of the 10 subjects. As a result, the performances of different illumination methods are compared in Fig. 4.21.

It can be seen from Fig. 4.21 that for all the subjects in the Yale database B, the simplified LBP preprocessing consistently achieves the best performance.

This indicates that the simplified LBP preprocessing has higher robustness to large illumination variability compared to the other methods. The original LBP and the Gaussian derivative filtering methods are also good compared to the rest and yield comparable performances. In this experiment within the Yale database, it is interesting to notice that the more the user looks alike to the average face, the worse the verification performance is, and vice versa.

4.7 Summary

This chapter presents a close study of the illumination normalization problem. Basically, there are two methodologies to deal with this problem. The first methodology is to approach the problem from the three dimensional point of view, trying to the recover the fundamental three-dimensional information of the face, like surface normals and albedos. This category of methods is theoretically optimal, but have certain drawbacks that limit their applicability. Firstly, the widely-used Lambertian imaging model simplifies the situation by assuming single light source from the infinite distance, and the model does not account for the shadows and spectacular reflectance that usually exist in face images. Secondly, over-stringent assumptions are often made in this category of methods, such as stability of the subject, or fixed three-dimensional shape of the face class. Thirdly, for training, this category of methods generally need relatively complicated database with respect to different illuminations, even the database of the specific user under defined illuminations. Moreover, the computation involved in the three-dimensional methods is usually high. For our MPD-based application, the three-dimensional methods can hardly be applied.

The second methodology, in contrast, deals with the face image from the pixel point of view, and is thus more direct and simpler. A literature review shows that most of this category of methods tends to remove the low frequency components, which are easily influenced by the illumination, in a linear or non-linear way. Due to the complexity of the imaging process, however, the remaining high frequency components contain both the illumination-free and the illumination-sensitive edge information. Strict invariance to illumination by the two-dimensional filters has been proved to be impossible [25], instead, we aim to make the filters as insensitive to illumination as possible.

In the face image, there are basically two components that are sensitive to the illumination changes: the first is certain image textures, which are easily altered by the illumination, like the textures around the nose region; the second is the pixel values, which are in close correlation with the illumination intensities. The first illumination-insensitive method, the horizontal Gaussian derivative filter, deals with the first type of sensitivity. It extracts the horizontal-directional image textures that are important of a face, but orthogonal to the vertical-directional image textures that are sensitive to illuminations. Besides, we have proved that the proposed filter is invariant to many patterns that can be modeled as illumination changes.

The second illumination-insensitive method, the simplified local binary pattern as a filter, deals with the second type of sensitivity. It is strictly invariant to any monotonic change of the pixel values, as LBP is a relatively measure of the image patterns. In the simplified LBP, we further remove the sensitivity of the LBP value to the direction, assigning uniform weights to the relative patterns in the eight directions. This is especially useful to remove the noises in the image regions where no distinctive textures are present. The method can be well interpreted by the simple Lambertian model.

Obviously, the two filters both filter out certain image information that is sensitive to the illuminations. However, will the remaining image information be enough for the subsequent verification purpose? We answer this question in Section 4.5, and discussed the generalization and discrimination capabilities in a high-dimensional space. In theory and by experiments, we have proved that depside the loss of certain illumination-sensitive information, the proposed filters still preserve enough discriminative information between the user class and the background class. Especially the simplified LBP filter is able to achieve the best performance in our experiments.

Chapter 5

Decision Level Fusion

5.1 Introduction

[1]Fusion is a popular practice to increase the reliability of the biometric verification by combining the information of multiple classifiers [133] [170] [48] [178]. Generally speaking, fusion can be done at four different levels: sensor level, feature level, matching score level, and decision level [133] [48], as illustrated in Fig. 5.1.

Fusion at sensor level is closely related to the specific sensor types and the corresponding signal/image processing methods. For a more compact review, we will concentrate on the last three levels, which are closely related to a classifier. At the feature level, for each classifier, the feature vector is in a high dimensional space: $x_i \in \mathbb{R}^{m_i}, m_i \geq 1$, $i = 1, 2, ..., N$. Note that the dimensionalities m_i and m_j could be different for $i \neq j$. At the matching score level, the feature vector is reduced to a scalar value, $s_i \in \mathbb{R}$, $i = 1, 2, ..., N$. At the decision level, the matching scores s_i are compared to the thresholds T_i, and the outputs are binary decisions $d_i \in \{1, 0\}$, $i = 1, 2, ..., N$.

Fusion at the feature level is not often used in practice [133]. This is due to the fact that the feature sets of different modalities can be incompatible, for example some feature values might be locations (e.g. of the minutiae set of the fingerprint) while some might be grayvalues (e.g. of the face images), which makes it infeasible to combine them on the same ground. Moreover, even if a combination rule could be designed, the size of the resulting feature vector will

[1] This Chapter is based on the publication [156], [160], [178].

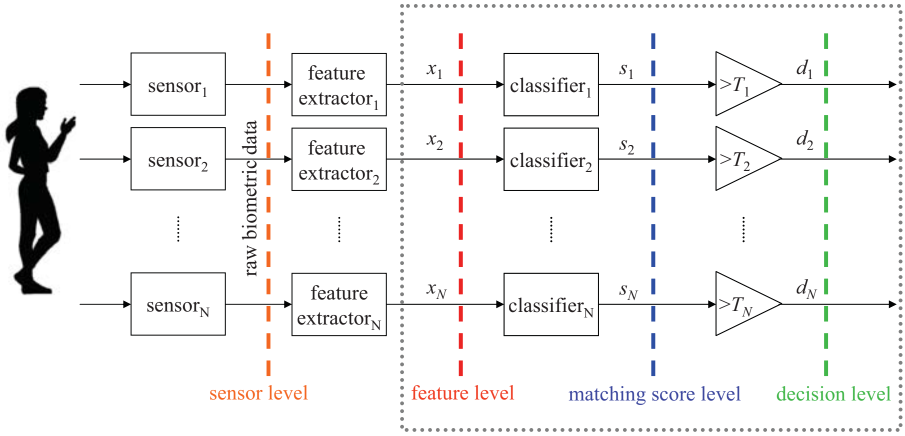


Figure 5.1: Different levels of fusion: sensor level, feature level, matching score level, and decision level.

often increase. This, in turn, increases the complexity of the system, making it more difficult to design and train the classifier.

Fusion at matching score level is the most popular way of fusion, offering the best tradeoff between information content and ease of fusion [133]. Matching score level fusion has been extensively studied in literature, such as [89, 182, 141, 132, 57], etc. There are basically three types of fusion schemes at the matching score level. The first type of fusion scheme is *transformation-based*. Firstly, all the component matching scores are transformed/normalized so that they are on a comparable scale. Then simple scalar functions are applied on the transformed matching scores, resulting in a new matching score. Examples of the functions are product, sum, mean, max, etc. [89] [102] [101]. It can be proved that under certain ideal situations, for example, taking the product of independent likelihood ratios can achieve the statistically optimal performance in the Neyman-Pearson sense [174]. The second type of fusion scheme is *density-based*. It relies on the estimation of the joint densities of the matching scores, and the fusion is done by statistical tests, like the likelihood ratio test [35] [76] [112] [170] [129] according to the estimated score distributions. This type of fusion scheme achieves optimal performance if the densities could be accurately learnt, under the situation when a large number of representative training matching scores are available. The third type of fusion scheme is *classifier-based*. It concatenates the component matching scores as a new feature vector, and trains additional classifiers on them. Examples are neural networks [182], support vector machines [141], decision trees [132]. This type of fusion needs to train the relevant classification parameters.

Matching score normalization is necessary for the first type of matching score level fusion, especially when the fusion is done between different classifiers or modalities, with the output matching scores defined in their own different ways. Such normalization is also important for the remaining two fusion schemes, as it can essentially affect the matching score densities.

Fusion at decision level is less studied in literature, as it is often considered inferior to matching score level fusion, on the basis that decisions are too "hard" and have less information content compared to "soft" matching scores. One example of fusion on decision level is the majority vote [44, 89], which counts the number of decisions d from the component classifiers, and chooses the majority of the decision as the final decision. Its derivative, weighted majority voting [99], assigns different weights according to different performances of the component classifiers. This consequently transforms the output value from logical numbers to continuous numbers. More of such examples are the Bayesian decision fusion [84], Dempster-Shafer theory of evidence [91], which also convert the decisions into scores, with the converting parameters learned from a training score set.

"Soft" measures with information of the confidence level have always been preferred in fusion. It has been shown by [39] that the combination of two matchers using AND or OR rule might actually degrade the overall performance when the performances of component classifiers are significantly different. Due to this phenomenon, AND and OR rules are rarely recommended in practice [133]. In this chapter, however, we propose a decision-level fusion scheme, by the AND and OR rule, in an optimal way such that it always gives an improvement in terms of error rates over the classifiers that are fused. Here, optimal is in the Neyman-Pearson sense [174]: at a given false-acceptance rate (FAR) α, the decision-fused classifier has a false-rejection rate (FRR) β that is minimal, and never larger than the FRR of the classifiers that are fused at the same α; or at a given β, the decision-fused classifier reaches a minimal α.

There exist some scenarios in which the proposed decision-level fusion is preferable to score-level fusion. For example, in template-protected biometrics, an accept or reject decision is based on the equality between a binary string extracted from the biometric data and a reference binary string [168]. This means the matching score is not available for fusion. If such a system is based on a fuzzy commitment [82] or a fuzzy vault scheme [113], the error correction is a part of the extraction of the binary string. The error correction can, within limits, be tuned to correct a certain maximum number of errors. This determines the point of operation on the ROC, and is equivalent to tuning a matching score threshold. Therefore, the proposed optimized decision-level fusion can be used to fuse two template protected biometric systems, and the optimal fusion can be achieved by tuning the number or corrected errors. Another scenario is that when the outliers are present in the biometric data. In that case, as we will discuss in Section 5.4.1, the proposed OR rule fusion often outperforms the conventional score-level fusion methods.

This Chapter is organized as follows. Section 5.2 and Section 5.3 presents the threshold-optimized decision-level theory on statistically independent and dependent classifiers, respectively. Section 5.4 discusses two useful biometric application of the proposed method. Section 5.5 presents the experimental results of the fusion between two face modalities and different algorithms. Section 5.6 summarizes this chapter.

5.2 Threshold-Optimized Decision-Level Fusion of Independent Decisions

5.2.1 The Decision and the ROC

A decision can be denoted by a logical number $d \in \{1, 0\}$, where 1 is for "accept" and 0 for "reject". From a classifier point of view, any decision d_i is obtained by comparing the matching scores s_i with a certain threshold T_i (see Fig. 5.1). In the proposed decision level fusion with optimized thresholds, we do not prefix the thresholds T_i of the individual component classifiers as is common in conventional decision-level fusion [39], instead, we optimize the combination of these thresholds, according to their joint behavior in the AND or OR rule fusion.

Before discussing the optimization process in detail, let us first look at the characterization of individual classifiers. Each decision d of a classifier is characterized by two error probabilities: the first is the probability of a false acceptance, the FAR, α, and the second is the probability of a false rejection, FRR, β. Obviously, FAR and FRR are both functions of T. When T varies, the FRR can be seen as a function of the FAR, $\beta(\alpha)$, known as the detection error trade-off characteristic (DET) [103]. DET is an indication of classification performance, revealing the inherent separability of the two opposite classes. An equivalent measure is the receiver operating characteristic (ROC), in which the detection rate $p_{\mathrm{d}} = 1 - \beta$ is expressed as a function of α, $p_{\mathrm{d}}(\alpha)$ [49]. We will use ROC for illustration throughout this and the following chapter. However, as we deal with the OR rule most of the time, it is more convenient to use $p_\mathrm{r}(\beta)$ ($p_\mathrm{r} = 1-\alpha$ is the correct rejection rate), a classifier's *rejection characteristic*, equivalent to the ROC, in all the mathematical derivations.

Depending on statistical properties of the component decisions, two different situations are identified. First, the multiple decisions d_i, $i = 1, 2, ..., N$ are statistically independent. This is desirable in fusion, as it has been observed that fusion works better when the fused components are independent [44, 89] or negatively dependent [92]. This situation occurs in many multi-modal biometric fusion cases, and facilitates a fast training based on ROC, as will be shown in Section 5.2.3. Second, the multiple decisions d_i, $i = 1, 2, ..., N$ possess some dependencies. Threshold-optimized decision-level fusion can also be solved for dependent decisions in a non-parametric manner, but the training is much slower and the optimized thresholds are more sensitive to the training set. Actually, the ROC-based training for independent decisions suffices for most fusion applications, even when some dependency exists. This is analog to the Naive Bayes classifier [44], which also assumes independency between different features, but whose good performance in dependency cases has been acknowledged in a wide range of applications [187] [43].

In all the following derivations, we will mainly focus on the OR rule, which is of more practical interest than the AND rule.

5.2.2 Problem Definition

Suppose we have N statistically independent decisions d_i, $i = 1, 2, ..., N$. To analyze the OR rule we have to work with the rejection rate, β and p_r. After application of the OR rule to decisions $d_i, i = 1, ..., N$, we have, under the assumption that all decisions are statistically independent, that

$$\beta = \prod_{i=1}^{N} \beta_i, \quad p_\mathrm{r}(\beta) = \prod_{i=1}^{N} p_{\mathrm{r},i}(\beta_i) \tag{5.1}$$

with β the false-rejection rate and p_r the correct-rejection rate of the final fused decision, respectively. The optimized OR rule decision fusion can then be formally defined by finding

$$\hat{p}_\mathrm{r}(\beta) = \max_{\beta_i | \prod \beta_i = \beta} \prod_{i=1}^{N} p_{\mathrm{r},i}(\beta_i) \tag{5.2}$$

where $\hat{p}_{\rm r}$ is the maximal correct rejection rate at β. In other words, the β_i's of the component classifiers are tuned during this optimization, so that the fused classifier can give maximal $p_{\rm r}$ at a fixed $\beta = \prod_{i=1}^{N} \beta_i$.

Likewise, the optimized AND rule decision fusion can be also formulated

$$\hat{p}_{\rm d}(\alpha) = \max_{\alpha_i \,|\, \prod \alpha_i = \alpha} \prod_{i=1}^{N} p_{{\rm d},i}(\alpha_i) \tag{5.3}$$

It is easily proved that the optimized correct-rejection rate $\hat{p}_r(\beta)$ is never smaller than any of the $p_{{\rm r},i}$'s at the same β

$$\hat{p}_{\rm r}(\beta) \geq p_{{\rm r},i}(\beta) \qquad i = 1, ..., N \tag{5.4}$$

Because, by definition

$$\hat{p}_{\rm r}(\beta) = \max_{\beta_i \,|\, \prod \beta_i = \beta} \prod_{i=1}^{N} p_{{\rm r},i}(\beta_i) \geq \prod_{j=1}^{N} p_{{\rm r},j}(\beta_j) \Bigg|_{\prod_{i=1}^{N} \beta_i = \beta} \tag{5.5}$$

As it holds for any classifier that, $p_{{\rm r},i}(1) = 1$, (5.4) readily follows by setting $\beta_j = \beta$ and $\beta_i = 1$ for all $i \neq j$.

By solving the optimization problem in (5.2) and (6.18), the optimal operation points for every component classifiers are obtained.

5.2.3 Problem Solution

In the work of [188], a similar optimization problem as in (6.18) is reformulated in a logarithmic domain. Under the assumption that $\log(p_{{\rm r},i}(\beta_i))$ is a concave function of $\log(\beta_i)$, it is proposed to find the optimal operation points by solving the unconstrained Lagrange optimization problem

$$\begin{aligned} \max\{\log p_{\rm r} - \lambda \log \beta\} &= \max \left\{ \sum_{i=1}^{N} \log(p_{{\rm r},i}(\beta_i)) - \lambda \left(\sum_{i=1}^{N} \log(\beta_i) \right) \right\} \\ &= \sum_{i=1}^{N} \max\{\log(p_{{\rm r},i}(\beta_i)) - \lambda \log(\beta_i)\} \end{aligned} \tag{5.6}$$

Due to the log-concavity assumption of each individual ROC, this optimization can be done by maximizing the value of $\log(p_{{\rm r},i}(\beta_i)) - \lambda \log(\beta_i)$ for each

ROC individually, and thus avoiding exhaustive search. For more details, see [188] [119]. One drawback of this method is that it does introduce a possibly too restrictive assumption on the ROC. The concavity of $p_{\mathrm{r}}\beta$ and $p_{\mathrm{d}}(\alpha)$ always holds in the original domain, but it does not always apply in the logarithmic domain. To avoid this drawback, we present an alternative approach, without any additional assumption or approximation. We propose that the optimization problem (5.2) and (6.18) be solved in a recursive manner: first fuse two arbitrary classifiers from the set of component classifiers, compute the ROC of the fused classifier, and then fuse the resulting ROC with the next arbitrary component ROC, and so on. This means that every time we only have to fuse two classifiers, thus avoiding the exponential explosion in computational complexity in combining multiple classifiers. We summarize the solution in the following:

1. Given N component classifiers, each characterized by $p_{\mathrm{d},i}(\alpha_i)$ or $p_{\mathrm{r},i}(\beta_i)$, $i = 1, ..., N$. Each operation point corresponds to a threshold.

2. Take any two ROCs and do threshold-optimized decision fusion.

3. Replace the two ROCs with the optimally fused ROC. Note that for a single operation point on the already fused ROC, there are now multiple thresholds coming from the component classifiers.

4. Repeat step (2)-(3) until all the classifiers have been combined.

5. A final ROC $p_{\mathrm{d}}(\alpha)$ or $p_{\mathrm{r}}(\beta)$ is obtained, with each operation point corresponding to N thresholds from the N component classifiers.

The only problem left now is the fusion of two ROCs in step (2). In real situations, $\hat{p}_{\mathrm{d}}(\alpha)$ is not available in its analytical form, but instead characterized by a set of discrete operation points. Therefore, we solve the fusion of two ROCs in a brute-force manner. Suppose we have two ROCs, denoted by N_1 and N_2 discrete operation points, respectively

$$\begin{aligned}\mathrm{ROC}_1 &= \{(\beta_1^i,\, p_{\mathrm{r},1}^i)\}, \qquad i = 1, 2, ..., N_1 \\ \mathrm{ROC}_2 &= \{(\beta_2^j,\, p_{\mathrm{r},2}^j)\}, \qquad j = 1, 2, ..., N_2\end{aligned}$$

The fusion of these two classifiers, under the independent assumption, can have in total $N_1 \cdot N_2$ possible combinations after OR rule fusion (AND rule fusion can be derived similarly by using α and p_{d})

$$\text{OR rule} \quad : \qquad \{(\beta_1^i \beta_2^j, \;\; p_{\mathrm{r},1}^i p_{\mathrm{r},2}^j)\}$$

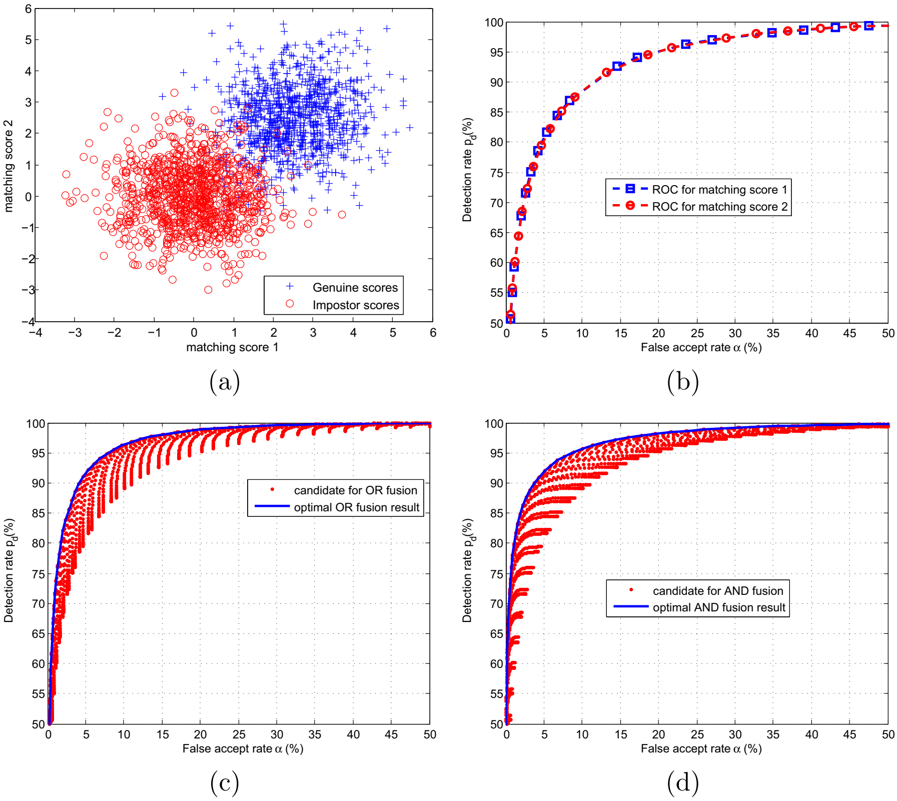
where $i = 1, 2, ..., N_1, \;\; j = 1, 2, ..., N_2$. Obviously, each pair of operation points corresponds to a pair of thresholds (T_1, T_2) with T_1 from the first classifier and T_2 from the second classifier. To get the optimized fusion, we select those operation points which form a concave hull of all the possible combinations. Fig. 5.2 illustrates this optimization process. In this example, we have generated the genuine and impostor scores independently for two classifiers. The genuine scores of the two classifiers has a multivariate Gaussian distribution of $N\left((2.5, 2.5), \left(\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right)\right)$, while the impostor scores of the two classifiers has a multivariate Gaussian distribution of $N\left((0, 0), \left(\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right)\right)$. In Fig. 5.2 (c) and (d), the dots denote all the possible combinations for the OR rule and the AND rule fusion, and the solid line marks the concave hull, which is optimal in the Neyman-Pearson sense. The optimized thresholds for decision level fusion are, therefore, obtained as the thresholds corresponding to the selected points of operation. It can be seen that both the OR rule and the AND rule fusion result in a better ROC than the original two ROCs.

Note that the optimality of the solution is strictly true in independent cases, and the ROCs in Fig. 5.2 (c) and (d) are the estimation of the fused ROCs under the independency assumption. When the matching scores have some dependencies, as we will show in the next section, the margin of ROC improvement is smaller compared to that of the independent case.

5.2.4 Optimality of Recursive Fusion

This procedure leads to an optimal solution, which is shown below for the OR-rule. The proof for the AND-rule is similar. In the following derivation, the matching scores of different classifiers are assumed to be independent.

Let $\mathcal{I}$ and $\mathcal{J}$ denote the index sets, such that $\mathcal{I} \cap \mathcal{J} = \emptyset$ and $\mathcal{I} \cup \mathcal{J} =$



(a)

(b)

(c)

(d)

Figure 5.2: Threshold-optimized decision fusion in the independent case: (a) the scatter plot of two matching scores; (b) two ROCs of the two matching scores, respectively; (c) all the possible OR fused points and the optimal ROC selected; (d) all the possible AND fused points and the optimal ROC selected.

$\{1, \ldots, N\}$. Define

$$p_{\mathrm{r}}^{\mathcal{I}}(\beta) \quad = \max_{\beta_i \,|\, \prod \beta_i = \beta} \prod_{i \in \mathcal{I}} p_{\mathrm{r},i}(\beta_i), \tag{5.7}$$

$$p_{\mathrm{r}}^{\mathcal{J}}(\beta) \quad = \max_{\beta_j \,|\, \prod \beta_j = \beta} \prod_{j \in \mathcal{J}} p_{\mathrm{r},j}(\beta_j), \tag{5.8}$$

and

$$p_{\mathrm{r}}^{\mathcal{I}\mathcal{J}}(\beta) = \max_{\beta^{\mathcal{I}} \beta^{\mathcal{J}} = \beta} p_{\mathrm{r}}^{\mathcal{I}}(\beta^{\mathcal{I}}) p_{\mathrm{r}}^{\mathcal{J}}(\beta^{\mathcal{J}}). \tag{5.9}$$

First, expanding $p_{\mathrm{r}}^{\mathcal{I}\mathcal{J}}(\beta)$ results in a product $\prod_{k=1}^{N} p_{\mathrm{r},k}(\beta_k)$ for some β_k, $k = 1, \ldots, N$, satisfying $\prod_{k=1}^{N} \beta_k = \beta$. Therefore, we have

$$p_{\mathrm{r}}^{\mathcal{I}\mathcal{J}}(\beta) \leq \max_{\beta_k \,|\, \prod \beta_k = \beta} \prod_{k=1}^{N} p_{\mathrm{r},k}(\beta_k). \tag{5.10}$$

Second,

$$\begin{aligned} p_{\mathrm{r}}^{\mathcal{I}\mathcal{J}}(\beta) \quad &\geq \quad p_{\mathrm{r}}^{\mathcal{I}}(\beta^{\mathcal{I}}) p_{\mathrm{r}}^{\mathcal{J}}(\beta^{\mathcal{J}})\Big|_{\forall \{\beta^{\mathcal{I}}, \beta^{\mathcal{J}}\}:\ \beta^{\mathcal{I}} \beta^{\mathcal{J}} = \beta} \\ &\geq \quad \prod_{i \in \mathcal{I}} p_{\mathrm{r},i}(\beta_i)\Bigg|_{\forall \{\beta_i\}_{i \in \mathcal{I}}:\ \prod \beta_i = \beta^{\mathcal{I}}} \prod_{j \in \mathcal{J}} p_{\mathrm{r},j}(\beta_j)\Bigg|_{\forall \{\beta_j\}_{j \in \mathcal{J}}:\ \prod \beta_j = \beta^{\mathcal{J}}} \\ &= \quad \prod_{k=1}^{N} p_{\mathrm{r},k}(\beta_k)\Bigg|_{\forall \{\beta_k\}_{k=1}^{N}:\ \prod \beta_k = \beta} \\ &\geq \quad \max_{\beta_k \,|\, \prod \beta_k = \beta} \prod_{k=1}^{N} p_{\mathrm{r},k}(\beta_k). \end{aligned} \tag{5.11}$$

The latter inequality follows by choosing the β_k such that they maximize $p_{\mathrm{r}}(\beta)$.

On combining (5.10) and (5.11) we have,

$$p_{\mathrm{r}}^{\mathcal{I}\mathcal{J}}(\beta) = \max_{\beta_k \,|\, \prod \beta_k = \beta} \prod_{k=1}^{N} p_{\mathrm{r},k}(\beta_k). \tag{5.12}$$

This means that *if the optimal ROCs are known for arbitrary disjoint index subsets $\mathcal{I}$ and $\mathcal{J}$, the overall optimal ROC can be found by optimally fusing the subsets.* Note that this statement is strictly true in ideal conditions, i.e., when the ROC is complete, with every point present on the ROC. In practice, however, the ROC cannot be complete, but represented by a limited number of operation points. The order of fusion, in this case, has some influences, but to an extent only as small as any other common numerical problems. As long as there are enough operation points from the ROC, the influences of the fusion order can well be neglected.

5.2.5 Additional Remarks

The ROC is a very useful but indirect indication of the score distributions. A highlight of the proposed decision-level fusion method is that it works on the operation points on the ROC, instead of on the matching scores as many other conventional fusion methods do. In practice, the number of the training matching scores could be enormous, but after calculating the ROC from the set, the number of ROC operation points is usually much smaller. On the other hand, when the number of the training matching scores is very small, the ROC points can even be interpolated and smoothed to produce a robust estimation. This simplifies the problem by converting any number of training scores to a manageable number of operation points on ROC. The optimization of the proposed decision-level fusion is again very simple. This makes the algorithm very efficient with training data sets of any size.

The computation involved in the training stage is the estimation of the ROC and the selection of the optimal ROC points. Given the training score set, it is very easy to calculate the ROC by comparing the scores with a number of thresholds, and estimate the FAR and FRR. The optimization, as in (5.2) and (6.18), is achieved simply by taking the outer boundary points in the $\alpha - p_{\text{d}}$ plane. In the verification stage, the calculation is extremely fast: for N classifiers, only N comparisons and $N-1$ AND or OR operations are required. Both the training and the fusion are simpler compared with advanced score-level fusion methods such support vector machines or likelihood ratio methods based Gaussian mixtures.

Score normalization is important in matching score level fusion. From the Neyman-Pearson point of view, it is most desirable that the matching score $s(x)$ be normalized in such a way that it is equal, or proportional to, the likelihood ratio of the feature vector x: $F(s(x)) = \frac{p(x|\omega_{\text{gen}})}{p(x|\omega_{\text{imp}})}$, where $F(\cdot)$ is a monotonic normalization function. Different normalization functions result in different decision boundaries in matching score level fusion. In comparison, an advantage of threshold-optimized decision level fusion is that the optimization is invariant to any monotonic transformation of the original matching scores. A monotonic function changes the absolute value of the matching scores, but does not alter the relative relationship between the matching scores. The operation points on the ROC, therefore, cannot be changed. As a result, the optimized operation points are invariant to any monotonic normalization. This implies that the final performance remains identical for any kind of score normalization function $F(\cdot)$.

There is always certain discrepancy between training and testing scores, which is one of the causes of overtraining. In many score-level fusion methods, such as the likelihood ratio method, SVM, or NN, there are a number of parameters to be estimated from the training data. The more parameters needed for characterization, the more flexible the boundary is in the score space, and the more sensitive it is to overtraining. In our decision-level method, we expect that, due to the coarser partitioning of the score space, the proposed fusion is more robust to model deviations between the training and testing data. This will be supported by results of the the fusion experiments in Section 5.5.

5.3 Threshold-Optimized Decision-Level Fusion on Dependent Decisions

It has been shown that to solve the proposed decision fusion problem under independency assumptions, we work directly on the ROCs and skip matching scores. In the dependent case, however, the fusion performance cannot be estimated as in (5.1). Instead, we return to the matching score space, and estimate the fusion performance in a nonparametric manner.

To illustrate the fusion process, we simulate two matching scores with dependency. The genuine matching scores have a multivariate Gaussian distribution of $N\left((2.5, 2.5), \left(\begin{smallmatrix}1 & 0.25\\ 0.25 & 1\end{smallmatrix}\right)\right)$, while the impostor matching scores have a multivariate Gaussian distribution of $N\left((0, 0), \left(\begin{smallmatrix}1 & 0.25\\ 0.25 & 1\end{smallmatrix}\right)\right)$. The matching scores are depicted by a scatter plot in a two-dimensional space, as shown in Fig. 5.3 (a).

To estimate the performance of fusion, we created an threshold grid covering the matching score space, as shown in Fig. 5.3 (a) by the cross points. The false-acceptance rate α and detection rate p_d at each operation point can be estimated simply by applying the AND or OR rule, and then counting the number of false-acceptances or false-rejections. Suppose we have N_gen genuine samples and N_imp impostor samples, then from two classifiers, we have N_gen pair of genuine scores $(s_1^\mathrm{gen}, s_2^\mathrm{gen})$, and N_imp pair of impostor scores $(s_1^\mathrm{imp}, s_2^\mathrm{imp})$. At any threshold (T_1, T_2), the ROC points by the OR and AND fusion can be

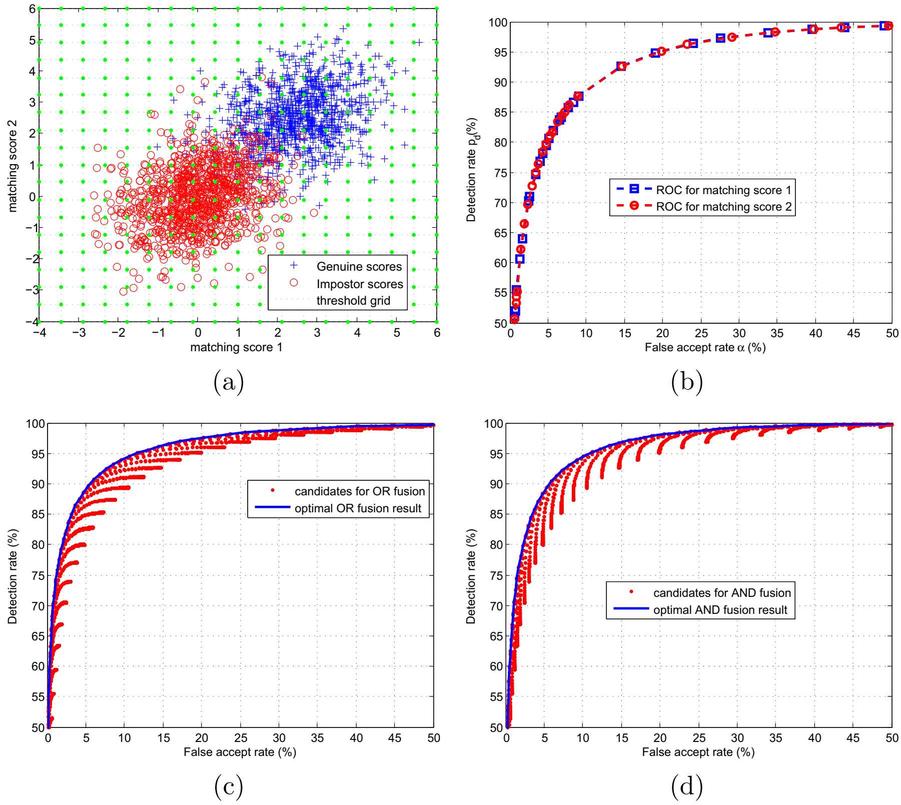


Figure 5.3: Threshold-optimized decision fusion in the dependent case: (a) the scatter plot of two matching scores, and the threshold grid; (b) two ROCs of the two matching scores, respectively; (c) all the possible OR fused points and the optimal ROC selected; (d) all the possible AND fused points and the optimal ROC selected.

easily calculated

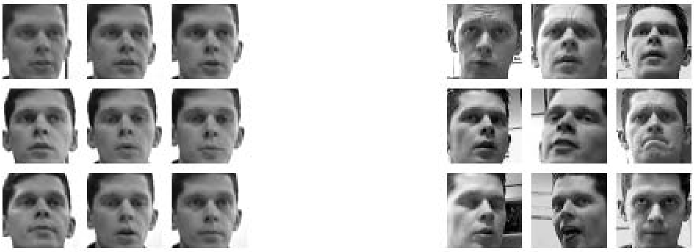
$$
\begin{aligned}
\alpha_{\mathrm{OR}}(T_1,T_2) &= \frac{\left\|\{(s_1^{\mathrm{imp}},s_2^{\mathrm{imp}})|(s_1^{\mathrm{imp}}\geq T_1)\vee(s_2^{\mathrm{imp}}\geq T_2)\}\right\|}{N_{\mathrm{imp}}}\\
p_{\mathrm{d\ OR}}(T_1,T_2) &= \frac{\|\{(s_1^{\mathrm{gen}},s_2^{\mathrm{gen}})|(s_1^{\mathrm{gen}}\geq T_1)\vee(s_2^{\mathrm{gen}}\geq T_2)\}\|}{N_{\mathrm{gen}}}
\end{aligned}
$$

$$
\begin{aligned}
\alpha_{\mathrm{AND}}(T_1,T_2) &= \frac{\left\|\{(s_1^{\mathrm{imp}},s_2^{\mathrm{imp}})|(s_1^{\mathrm{imp}}\geq T_1)\wedge(s_2^{\mathrm{imp}}\geq T_2)\}\right\|}{N_{\mathrm{imp}}}\\
p_{\mathrm{d\ AND}}(T_1,T_2) &= \frac{\|\{(s_1^{\mathrm{gen}},s_2^{\mathrm{gen}})|(s_1^{\mathrm{gen}}\geq T_1)\wedge(s_2^{\mathrm{gen}}\geq T_2)\}\|}{N_{\mathrm{gen}}}
\end{aligned}
$$

where $\|\cdot\|$ denotes size of the set. Consequently, for every threshold on the grid, a ROC points can be obtained, as shown in Fig. 5.3 (c) and (d) by dots. Like the independent case, we again select those operation points which form a concave hull of the candidate points, as shown in Fig. 5.3 (c) and (d). The optimized thresholds for decision level fusion is therefore obtained as the thresholds corresponding to the selected points of operation.

Without the independency assumption, the presented decision fusion still has the good property that, similar to (5.4), the resulting ROC outperforms either component ROCs. This can be proved by the fact that in fusion, the original points of (α, p_{d})'s on ROC$_1$ and ROC$_2$ are still existent in the pool of candidate points to be selected[2]. Therefore, the resulting ROC, after the optimization of the concave hull, is again more favorable over the original ROCs in the Neyman-Pearson sense. It can be noticed, however, the margin of improvement becomes smaller compared to the independent case, as dependency of the two classifiers implies less added information.

[2] For ROC$_1$, the original operation points are obtained when the operation points of ROC$_2$ are tuned to extremes: for AND rule, $T_2 \to -\infty$; for OR rule, $T_2 \to \infty$. The same is true for ROC$_2$.



(a) (b)

Figure 5.4: (a) Normal samples of the user data, (b) outliers in the user data.

5.4 Application of Threshold-Optimized Decision-Level Fusion to Biometrics

In the previous section, we have presented the theory of threshold-optimized decision fusion and the optimization method in detail. In this section, we will discuss some interesting applications of threshold-optimized decision fusion.

5.4.1 OR fusion in Presence of Outliers

In this section, we will discuss the situation when the proposed OR rule decision level fusion is especially favorable. Outliers, in biometric verification, refer to the biometric data which belong to the genuine user, but deviate from the genuine user distribution. Taking face for example, outliers can be caused by extraordinary expressions, poses, illuminations, or mis-registrations. Some examples are given in Fig. 5.4. Outliers cause false rejections most of the time.

Suppose the outlier scores have a probability density function of $\Psi_{\mathrm{out}}(s)$. This function could be approximated by the impostor distribution $\Psi_{\mathrm{imp}}(s)$, based on the fact the outlier scores have values that could otherwise be taken as impostors. Suppose the genuine score has a probability density function of $\Psi_{\mathrm{gen}}(s)$, and the prior probability of outliers occurring in the genuine score is p_{o}. Taking into account the outliers, the probability of the genuine score s is

$$\Psi'_{\mathrm{gen}}(s) = (1 - p_{\mathrm{o}}) \cdot \Psi_{\mathrm{gen}}(s) + p_{\mathrm{o}} \cdot \Psi_{\mathrm{imp}}(s) \tag{5.13}$$

Suppose we are fusing two independent classifiers, both with outliers in the genuine score. The joint probability of two independent samples s_1 and s_2 is

$$\begin{aligned}\Psi(s_1, s_2) \quad = \quad & (1 - p_{\mathrm{o},1})(1 - p_{\mathrm{o},2}) \cdot \Psi_{\mathrm{gen},1}(s_1)\Psi_{\mathrm{gen},2}(s_2) \\ & + p_{\mathrm{o},1}(1 - p_{\mathrm{o},2}) \cdot \Psi_{\mathrm{gen},1}(s_1)\Psi_{\mathrm{imp},2}(s_2) \\ & + (1 - p_{\mathrm{o},1})p_{\mathrm{o},2} \cdot \Psi_{\mathrm{imp},1}(s_1)\Psi_{\mathrm{gen},2}(s_2) \\ & + p_{\mathrm{o},1}p_{\mathrm{o},2} \cdot \Psi_{\mathrm{imp},1}(s_1)\Psi_{\mathrm{imp},2}(s_2) \qquad (5.14)\end{aligned}$$

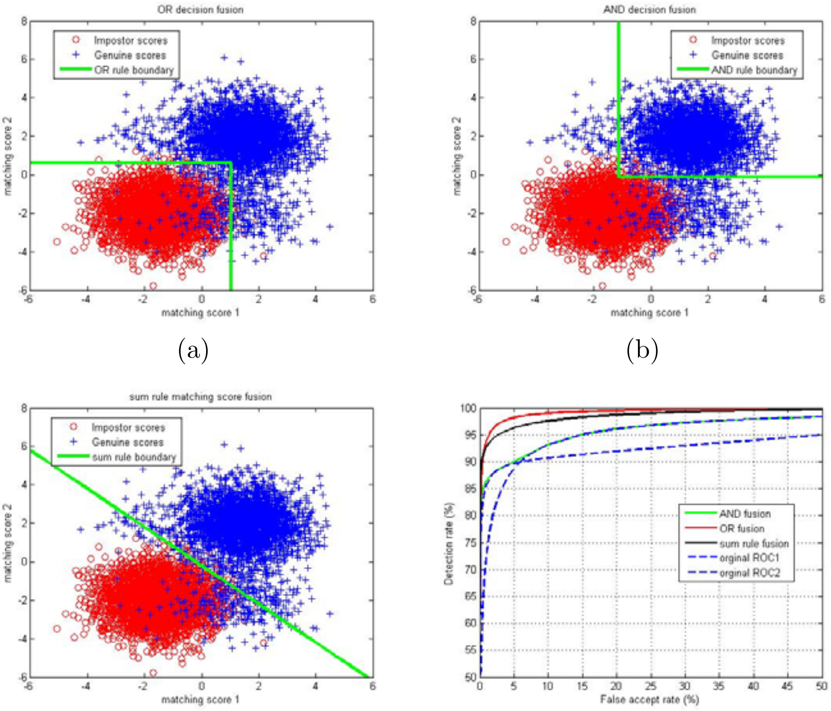
where the subscripts 1 and 2 indicate the first and the second classifier, respectively.

In the example in Fig. 5.5, for the first classifier, $p_{\mathrm{o},1} = 0.03$, $\Psi_{\mathrm{gen},1}(s_1) \sim N(1.5, 1)$, $\Psi_{\mathrm{imp},1}(s_1) \sim N(-1.5, 1)$, while for the second classifier, $p_{\mathrm{o},2} = 0.10$, $\Psi_{\mathrm{gen},2}(s_2) \sim N(2, 1)$, $\Psi_{\mathrm{imp},2}(s_2) \sim N(-2, 1)$. Fig. 5.5 (a), (b), and (c) show the boundaries of AND rule decision fusion, OR rule decision fusion, sum rule matching score fusion, respectively, at the fixed FAR $\alpha = 0.01$. Fig. 5.5 (d) compares the resulting ROC by different fusion schemes. Under the given situations with outliers, OR rule decision fusion achieves the best performance in a large range, for $\alpha > 0.005$. The AND rule fusion, in comparison, is not suitable for the given score distributions as it only results in the better of the two ROCs.

It is interesting to notice that in (5.14), the OR rule boundary accepts all the terms except the last one, which is negligible because of the small value of $p_{\mathrm{o},1}p_{\mathrm{o},2}$. This explains why OR rule decision fusion is suitable for this kind of problem.

The type of matching score distribution as simulated in Fig. 5.5 is not a rare scenario. It is very often the case that a number of outliers occur in the genuine class, thus making the genuine distribution extending to the impostor class. The impostor class, however, is less likely to produce such a comparable proportion of "outliers". Such phenomenon can be explained by the great discriminating power of a high-dimensional space [166], which makes a classifier in it more ready to reject than to accept. This is especially true for our likelihood-ratio based classifier, as has been introduced in Chapter 3 and discussed in Chapter 4.

Other fusion methods could also be applied to the fusion problem with outliers, such as density-based fusion, e.g. likelihood ratio test, or classifier-based fusion, e.g. SVM, NN, which also takes care of the outliers during training. However, the resulting decision boundary is more dependent on the training data. To accommodate the outliers, for example, the outliers should be included in the training set. In comparison, OR-rule fusion always has good tolerance with



(a)

(b)

(c)

(d)

Figure 5.5: (a) scatter plot of the scores and OR rule boundary; (b) scatter plot of the scores and AND rule boundary; (a) scatter plot and sum rule boundary; (a) comparison of the ROCs.

outliers no matter if they are included in the training set or not. The advantage of the proposed OR rule decision fusion, moreover, is its simplicity. First, a normalization step is not required; second, the calculation is faster, as only a limited number of operation points are involved in the calculation. Third, there is potentially less overtraining possibilities, as the decision boundaries is much simpler than those of the SVM or NN.

5.4.2 Fusion of Identical Classifiers

Fusion of identical classifiers is a useful scenario in practice, which means that given one classifier and multiple input samples, one can fuse the multiple decisions, each decision from a different input but from the same classifier [161]. Without changing the original system, improvement of the performance can be readily achieved. For simplicity, we first discuss the AND rule decision fusion on two identical classifiers. In a similar way, the OR rule fusion on two identical classifiers can derived.

Suppose we have two statistically independent decisions with identical ROC $p_{\mathrm{d},1}(\alpha_1) = p_{\mathrm{d},2}(\alpha_2) = p_{\mathrm{d}}(\alpha)$, the optimization problem can be formulated as

$$\hat{p}_{\mathrm{fusion}}(\alpha) = \max_{\alpha \leq x \leq 1} \left\{ p_{\mathrm{d}}(x) \cdot p_{\mathrm{d}}(\frac{\alpha}{x}) \right\} \tag{5.15}$$

where x is an intermediate variable, and $\hat{p}_{\mathrm{fusion}}(\alpha)$ is the detection rate at α under the optimal AND fusion.

For any $0 < \alpha < 1$, $p_{\mathrm{d}}(x) \cdot p_{\mathrm{d}}(\frac{\alpha}{x})$ is a continuous function of x. Taking the derivative of $p_{\mathrm{d}}(x) \cdot p_{\mathrm{d}}(\frac{\alpha}{x}))$ with respect to x, we have

$$\frac{\partial p_{\mathrm{fusion}}}{\partial x} = p_{\mathrm{d}}'(x) p_{\mathrm{d}}(\frac{\alpha}{x}) - \frac{\alpha}{x^2} p_{\mathrm{d}}(x) p_{\mathrm{d}}'(\frac{\alpha}{x}) \tag{5.16}$$

Obviously, when $x = \sqrt{\alpha}$, i.e. $\alpha_1 = \alpha_2 = \sqrt{\alpha}$, the derivative reaches zero. This means when the two classifiers have the same operation points, the extremum is reached. This extremum is most often a maximum, in which cases the original ROC $(\alpha, p_{\mathrm{d}}(\alpha))$ is mapped to the AND-rule fused ROC $\left(\alpha^2, p_{\mathrm{d}}^2(\alpha)\right)$[3]. In rare cases, however, for some α's on some type of ROCs, this stationary point

[3] By such a mapping, improvements can be expected for most ROCs, but for some type of ROCs, for example, $p_{\mathrm{d}}(\alpha) = \alpha^{\gamma}$ $(0 \leq \gamma \leq 1)$, the fused ROC is unchanged after this mapping. It also happens when such a mapping results in a degraded ROC, as will be shown in Fig. 5.7 (a).

corresponds to a minimum. Then the optimum is found at the border, either $x = 1$ or $x = \alpha$, which means that only one of the two ROCs is taken.

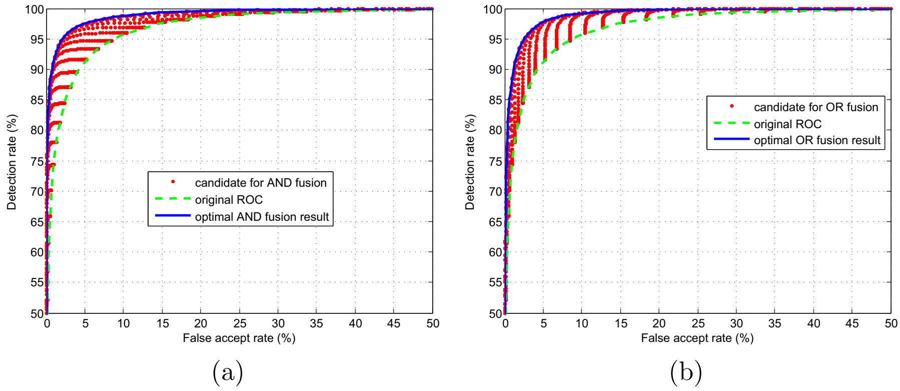
To illustrate the fusion of two identical classifiers, we simulated random data to generate the original ROC. Fig. 5.6 and Fig. 5.7 show two examples of fusion results on two identical classifiers. In the first example Fig. 5.6, the genuine score has a Gaussian distribution of $N(3, 1)$, while the impostor score has a Gaussian distribution of $N(0, 1)$. In the second example Fig. 5.7, the impostor score has the same Gaussian distribution of $N(0, 1)$, but the genuine score has a multimodal Gaussian distribution, with 90% of the data of the Gaussian distribution $N(3, 1)$, and the remaining 10% of the data of the Gaussian distribution $N(0, 1)$, simulating the outliers. In the first example of Fig. 5.6, improvements of performance can be observed both from AND and OR fusion. In the second example in Fig. 5.7, however, OR rule fusion is more suitable than AND rule fusion for the ROC with this specific shape. For AND fusion, it can be observed that in the low FAR region, fusion brings improvement, while in the high FAR region, fused performance is actually worse, therefore the original ROC is taken. This illustrates the case when identical operation points of the two component ROC correspond to a minimum instead of a maximum. This example shows that the solution to (5.15) is related not only to the shape of the original ROC, but also to the specific value of α in the resulting ROC.

We have discussed the fusion of two identical classifiers, and proved that in the optimal case, the component classifiers either work on identical points, or on extreme points ($\alpha = 1$ for AND fusion or $\beta = 1$ for OR fusion). This conclusion can be extended to the fusion of three or more classifiers. Suppose we have N identical classifiers $p_{\mathrm{d},1} = \cdots = p_{\mathrm{d},N} = p_\mathrm{d}$, the Lagrange optimization problem can be formulated in the logarithm domain

$$\begin{aligned} P_L(\alpha) \quad = \quad & \log p_\mathrm{d}(\alpha_1) + \cdots + \log p_\mathrm{d}(\alpha_N) \\ & -\lambda(\log \alpha_1 + \cdots + \log \alpha_N - \log \alpha) \end{aligned} \tag{5.17}$$

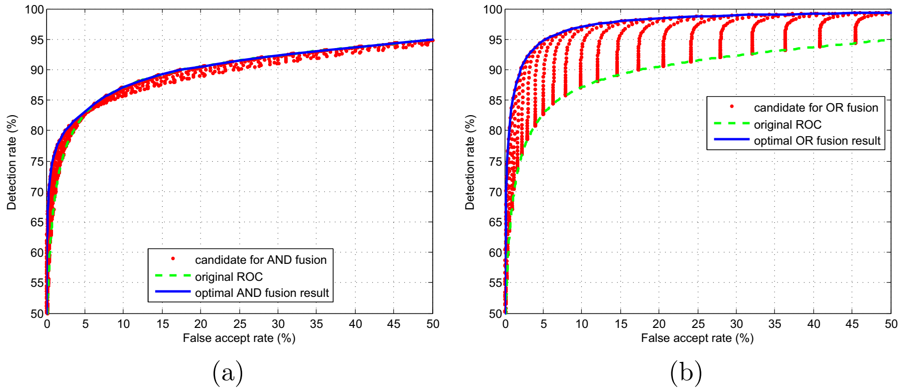
At a fixed α, taking derivative with respect to the component α_i, $i = 1, ..., N$, and at the extremum point, all the derivatives are zero. We have

$$\begin{aligned} \frac{\partial P_L}{\partial \alpha_1} &= \frac{p'_\mathrm{d}(\alpha_1)}{p_\mathrm{d}(\alpha_1)} - \frac{\lambda}{\alpha_1} = 0 \\ \vdots \quad & \quad \vdots \\ \frac{\partial P_L}{\partial \alpha_N} &= \frac{p'_\mathrm{d}(\alpha_N)}{p_\mathrm{d}(\alpha_N)} - \frac{\lambda}{\alpha_N} = 0 \end{aligned} \tag{5.18}$$



(a) (b)

Figure 5.6: Example 1: (a) AND rule decision fusion on identical classifiers. (b) OR rule decision fusion on identical classifiers.



(a) (b)

Figure 5.7: Example 2: (a) AND rule decision fusion on identical classifiers. (b) OR rule decision fusion on identical classifiers.

Therefore,

$$\lambda = \frac{\alpha_1 p'_{\mathrm{d}}(\alpha_1)}{p_{\mathrm{d}}(\alpha_1)} = \cdots = \frac{\alpha_N p'_{\mathrm{d}}(\alpha_N)}{p_{\mathrm{d}}(\alpha_N)} \tag{5.19}$$

A solution to (5.18) and (5.19) is

$$\alpha_1 = \cdots = \alpha_N = \alpha^{\frac{1}{N}}, \qquad \lambda = \frac{\alpha^{\frac{1}{N}} p'_{\mathrm{d}}(\alpha^{\frac{1}{N}})}{p_{\mathrm{d}}(\alpha^{\frac{1}{N}})} \tag{5.20}$$

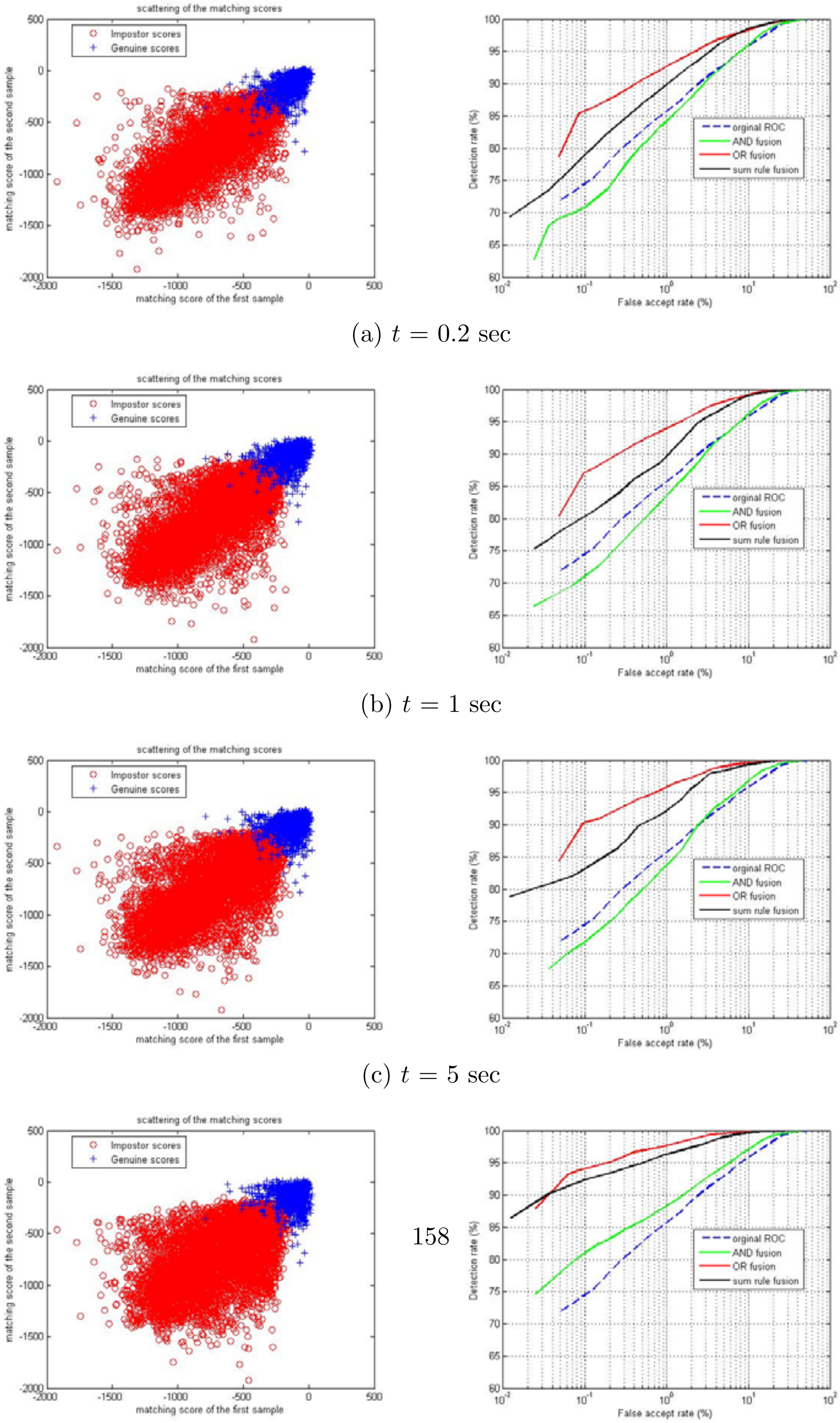
where α is a known quantity. Therefore, $\alpha_1, ..., \alpha_N$ and λ in (5.20) are legal solutions and correspond to the extremum of (5.17). As in the case of fusing two identical classifiers, this extremum is most often a maximum. When it corresponds to a minimum, the maximum occurs on the border, i.e., only part of the component classifiers are switched on.

For identical classifiers, the AND rule decision fusion is similar to the min rule matching score fusion (i.e. taking the minimum score as the final score), while the OR rule decision fusion is similar to the max rule matching score fusion (i.e. taking the maximum score as the final score) [89]. The decision fusion of identical classifiers, therefore, is to a certain extent comparable to the max or min rule matching score fusion. However, there are still differences which make the proposed decision fusion more favorable. Firstly, the decision fusion will avoid those situations when the fusion could actually degrade the performance (as illustrated in Fig. 5.7 (a)), and choose to use only part of the classifiers; Secondly, when the component classifiers are different, the proposed decision fusion automatically finds different thresholds for them, while for the max or min rule matching score fusion, the thresholds is the same for all component matching scores, so the scores have to be normalized first.

5.5 Experiments and Results

5.5.1 Experiments on the MPD Data

For intermodal optimized decision fusion, we present an example of a face verification system on the mobile device [157, 158]. The decision level fusion is done on identical classifiers from multiple inputs. In the original face verification system, faces are first detected by the Viola-Jones method [179], then registered by aligning prominent facial landmarks also detected by Viola-Jones method [10],



(a) $t = 0.2$ sec

(b) $t = 1$ sec

(c) $t = 5$ sec

(d) $t = 60$ sec

as introduced in Chapter 2. Illumination normalization is done by applying the simplified local binary patterns as an preprocessing method, as introduced in Chapter 4. The feature vector is then obtained by stacking the pixels from the preprocessed image. A likelihood ratio classifier [6, 174] is used, as introduced in Chapter 3.

In the new face verification system with decision fusion, two frames with certain intervals t are taken as the input, and decision fusion is done with the two output decisions. From a database of 6 subjects collected for mobile scenarios, the genuine scores are taken from one subject, while the impostor scores are taken from the other 5 subjects collected under the same illumination. According to the length of the interval, the two decisions exhibit different degree of dependency. The longer the interval, the less the dependency. We will test different time intervals t of 0.2 sec, 1 sec, 5 sec, and 60 sec. Furthermore, we will compare and proposed AND and OR rule decision fusion with the conventional matching score fusion by the sum rule. If the interval t is long enough so that the two input feature vector x_1 and x_2 are statistically independent, the sum rule is in theory the best fusion scheme for the log-likelihood ratios. It can be easily proved

$$\begin{aligned} s(x_1, x_2) &= \log \frac{p_{\text{user}}(x_1, x_2)}{p_{\text{bg}}(x_1, x_2)} \\ &= \log \frac{p_{\text{user}}(x_1) p_{\text{user}}(x_2)}{p_{\text{bg}}(x_1) p_{\text{bg}}(x_2)} \\ &= \log \frac{p_{\text{user}}(x_1)}{p_{\text{bg}}(x_1)} + \log \frac{p_{\text{user}}(x_2)}{p_{\text{bg}}(x_2)} \\ &= s(x_1) + s(x_2) \end{aligned} \tag{5.21}$$

Fig. 5.8 shows the scatter plot and the ROCs of fusion at different time intervals t. An advantage of this specific application is that, according to Section 3.1, in most cases, the optimal thresholds of the two classifiers do not need to be trained, and can be simply taken identical. To illustrate this, in this example we skipped training and simply took identical thresholds for two classifiers in cases of both AND and OR decision fusion. Actually, this causes problems with AND decision fusion for the presented type of ROC, like in Fig. 5.8 (a) and (b), where the AND rule fused ROC is actually worse than the original. This shows that the AND rule decision fusion cannot improve the performance of the presented type of ROC. In comparison, the OR rule decision fusion works especially well, even outperforming the sum rule, which phenomenon could be explained by

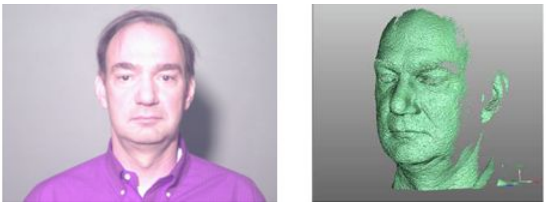


Figure 5.9: Example from the FRGC database: the 2D texture and the 3D shape recorded simultaneously recorded.

the existence of outliers. As can be further observed, with the increase of t, the dependency of the two decisions becomes less, and the improvement of performance by fusion is more obvious. When $t = 1$ sec, the EER of the ROC by the OR rule decision fusion is already reduced to half of the original. This implies that by taking an extra face frame, the performance of the original system can be easily improved by means of simple OR rule decision fusion.

5.5.2 Experiments on the 3D-Face Data

Another context of this work is the EU FP6 3D-Face project [1], which aims to use 3D facial shape data in combination with 2D texture data for reliable passport identification [156]. The first database that the algorithms were developed on is the FRGC database [124], which contains the 2D face texture and 3D face shape data collected simultaneously. An example of the two modalities is shown in Fig. 5.9. The database contains data of 465 subjects and has in total 4,007 samples. The classifiers that produce the matching scores are trained on 309 subjects in the database. To train fusion, another 100 subjects are taken to obtain the matching scores from the trained classifier, resulting in 25,520 genuine scores and 2,568,190 impostor scores. The remaining 56 subjects are used for evaluation, resulting in 12,270 genuine scores and 700,910 impostor scores.

For either modality, the matching scores are derived and provided by L-1 Identity Solutions (L1), Cognitec Systems (COG), and the University of Twente (UTW). In the L-1 method, the matching scores are computed using the hierarchical graph matching (HGM) methods [69], which represents the facial geometry by means of a flexible grid. Similar to the biological structures in the human brain, a set of specific filter structures is assigned to each node of the graph and analyzes the local facial characteristics [70] [184]. With HGM, approximately 2,000 characteristics are used to represent a face and an individual identity. For the analysis of a face, the shape (”landmarks”) and the structure (”features”) of the face are separated, making HGM a very robust facial recognition method providing a basis for both 2-D and 3-D face recognition. In the COG method, for 2D faces, the feature components are retrieved by applying local image Gabor transforms at facial feature locations. These component are then concatenated to form the raw 2-D face feature vector. For 3-D faces, the face shape is firstly registered and smoothed to form the raw 3-D face feature vector. Global transformations are applied on the raw feature vectors in both cases, in order to maximize the ratio of inter-personal variance to intra-personal variance [108]. The final scores are obtained by simple similarity measures of the transformed feature vectors. In the UTW methods, holistic approach is taken, and the feature vectors are derived by the conventional PCA and LDA transformation, and the scores are computed as the likelihood ratio of the feature vector in the feature space. More details of the mathematics can be found in [6].

For comparison, we also implemented three other typical score-level fusion methods, namely, sum rule (transformation-based), likelihood ratio (density-based), SVM (classifier-based), which are explained in more detail in the following:

1. Sum Rule
   In this transformation-based method, we used the simple and effective Z-normalization [133], which normalizes the genuine or impostor scores to unit variance. In the comparison, we use the Z-normalization based on the genuine scores[4].

2. Likelihood Ratio
   In this density-based method, the score density is first estimated using Gaussian mixture models (GMM) [51], as in the work of [129]. Then the likelihood ratio is calculated based on the estimation of both genuine and impostor score distributions.

3. SVM
   In this classification-based methods, we used SVM as the classifier. The decision boundary is trained using the radius basis function (RBF) kernels

[4] We only present this one for readability of the figures. Z-normalization based on the impostor scores and other normalization techniques like Min-max-normalization and Tanh-normalization [133] have also been tried and yielded similar results.

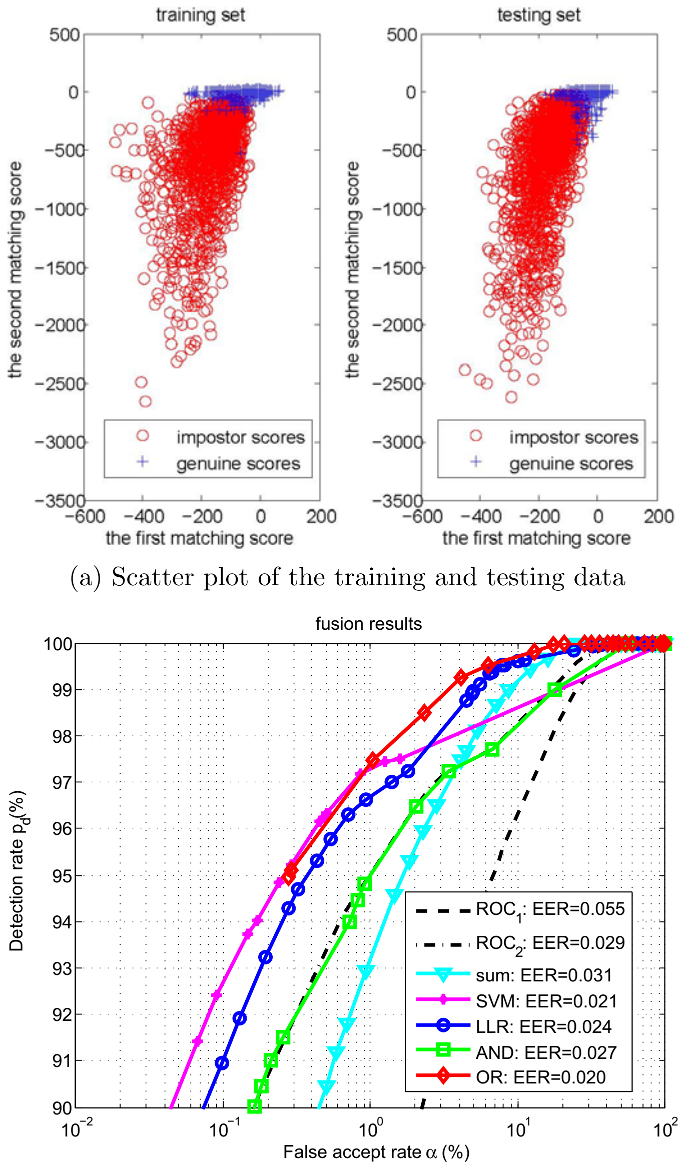
[30]. The scores are firstly Z-normalized with a variance of 1, and the RBF radius is chosen as 1. More implementation details can be found in [77].

Fusion is done between the two face modalities with scores derived from different algorithms. In each experiment, a training set is used first to find the parameters for fusion. In the decision-level fusion, the parameters refer to the optimized thresholds; while in the score-level fusion, the parameters refer to the normalization factors in the method (1), distributional parameters in the method (2), and SVM coefficients in the method (3). Then the evaluation of the methods are conducted on the testing data. For each fusion method, the resulting ROC are calculated and compared. Note that here we do not compare only a single operation point, instead we give an overview of the performance by plotting ROC, i.e., all the possible operation points.

Fig. 5.10 to Fig. 5.13 show 4 different combinations of the fusion between the face texture modality and shape modality. The ROC and the EER are shown, as well as the training score and testing score scatter plot. As observed, in Fig. 5.10 and 5.12 the OR rule fusion outperforms all other score-level fusion methods with respect to EER. The method even works better than the theoretically optimal LLR method. This can be explained by the discernible difference between the training and testing scores, which means that the probability density function might be over-tuned during the training. For the same reason, the support vectors are also different in the training and test set, thus accounting for the unsatisfactory performance of the SVM fusion method. Compared to the score-level fusion, the proposed ROC-based decision-level fusion are less sensitive to the training-testing data deviation, as indicated by Fig. 5.10. Another factor that makes the OR rule fusion favorable is its robustness against the outliers, as explained in Section 5.4.1. The AND rule fusion, however, does not work well, yielding performance sometimes even worse than the component ROC[5].

In Fig. 5.11 the OR rule fusion works well, outperforming the score-level fusion methods except the likelihood ratio one, on the FAR range from 0.5% to 100%, but not as well on the lower FAR range (note the logarithm scale exaggerates this part). The likelihood ratio fusion method, as in [112] and [129], remains the best. In Fig. 5.13 the OR rule fusion also performs worse than the LLR method in the lower FAR region, but equally well as far as the EER is concerned.

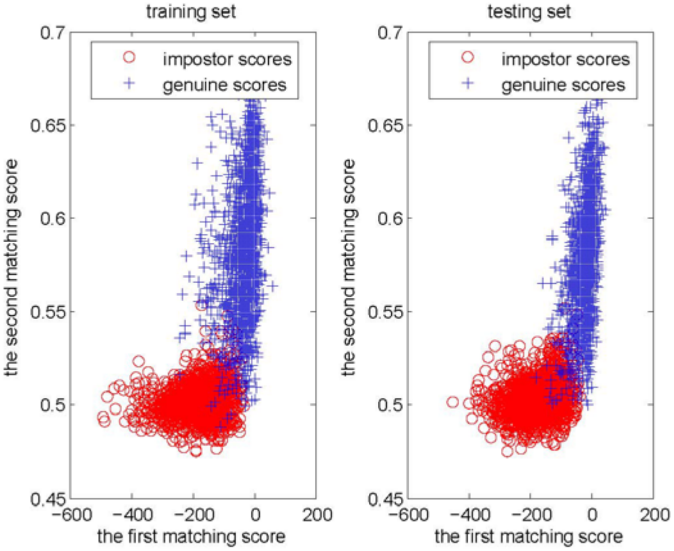
[5] In theory, this is not possible, but it happens when the training set and testing set are different.



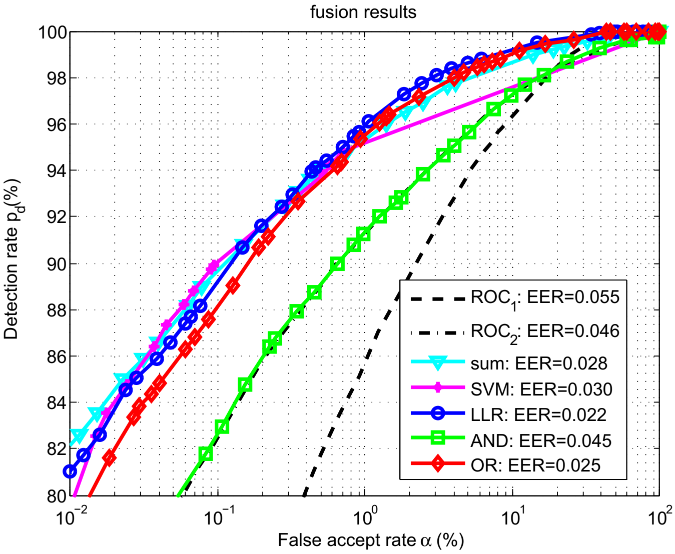
(a) Scatter plot of the training and testing data

(b) Fusion results

Figure 5.10: Fusion between the UTW texture and UTW shape data: scatter plot and the fusion results.

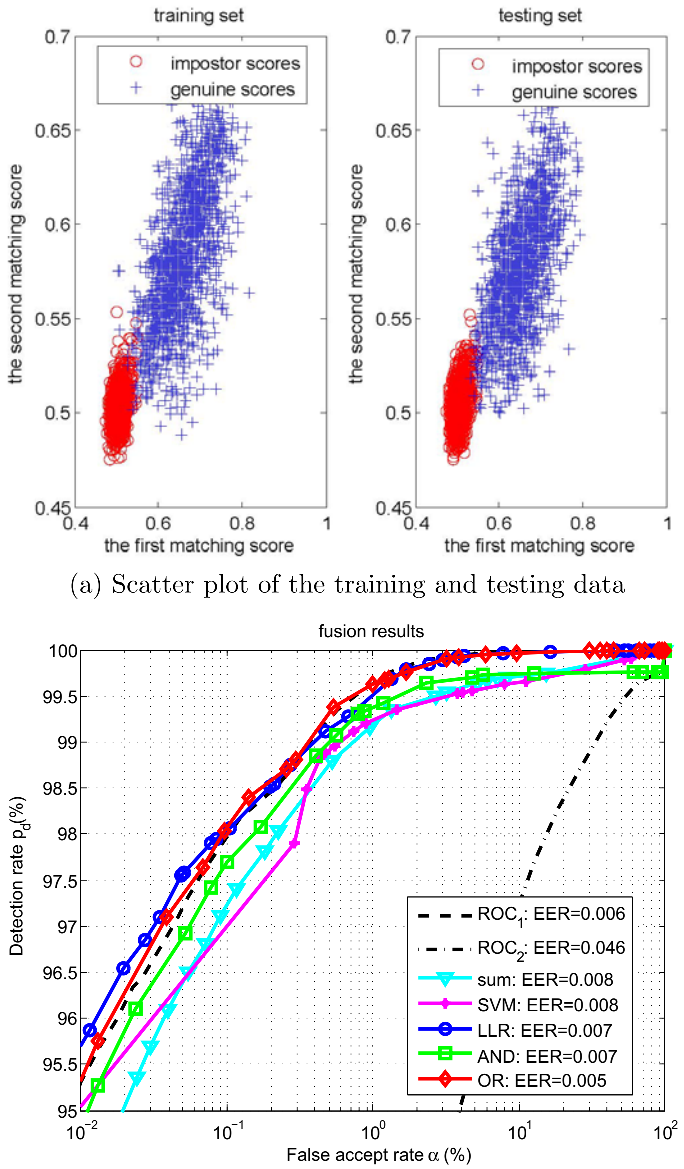


(a) Scatter plot of the training and testing data



(b) Fusion results

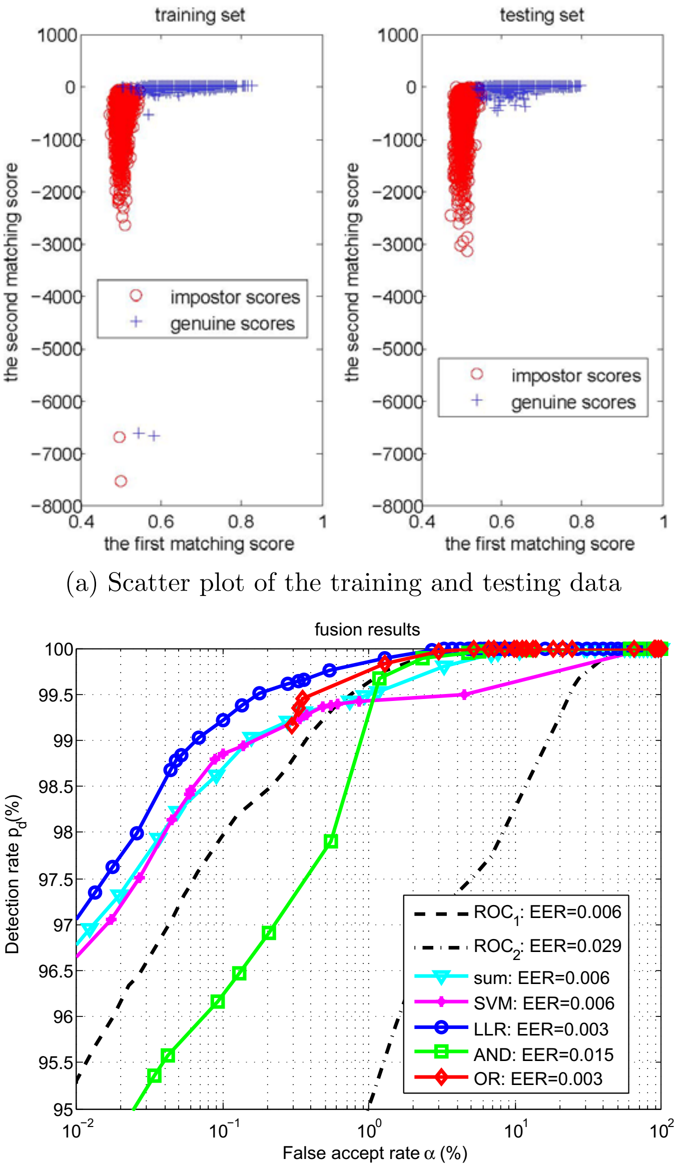
Figure 5.11: Fusion between the UTW texture and L1 shape data: scatter plot and the fusion results.



(a) Scatter plot of the training and testing data

(b) Fusion results

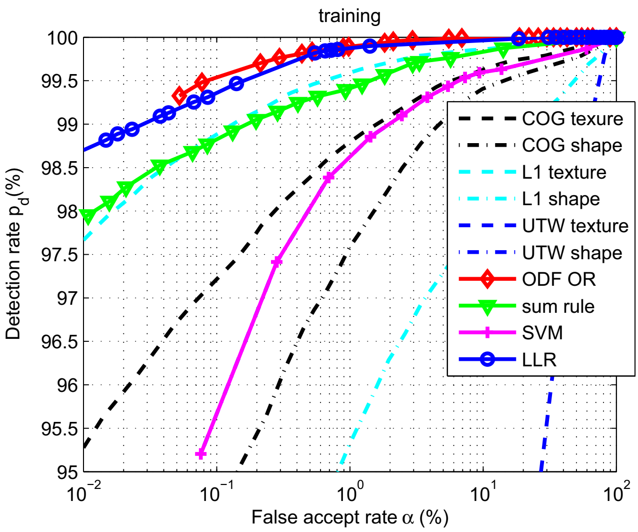
Figure 5.12: Fusion between the L1 texture data and L1 shape data: scatter plot and the fusion results.



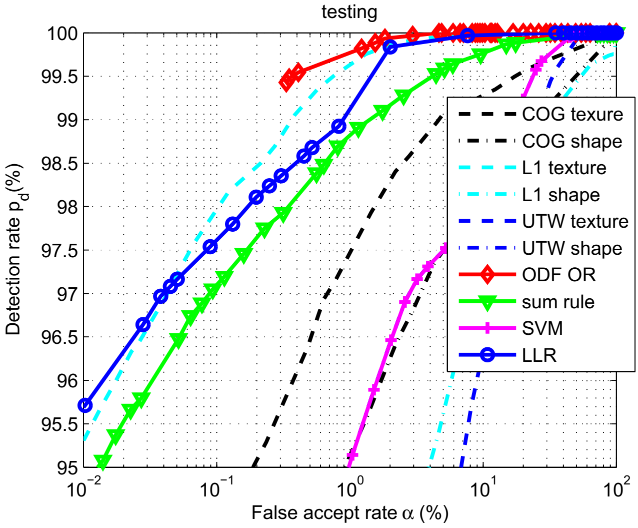
(a) Scatter plot of the training and testing data

(b) Fusion results

Figure 5.13: Fusion between the L1 texture data and UTW shape data: scatter plot and the fusion results.



(a) training ROCs



(b) testing ROCs

Figure 5.14: Fusion of all the 6 classifiers: (a) training ROCs, (b) testing ROCs.

We have further combined all the 6 classifiers: 3 texture classifiers and 3 shape classifiers. The fusion is done in a recursive way as introduced in Section 5.2.4. We show the fusion results on both the training ROCs and the testing ROCs in Fig. 5.14. It can be seen that the proposed decision-fusion outperforms other score-level methods, with considerable margin in the testing case. Such good performance is accounted by the outlier phenomenon existing in some component scores, as well as the obvious discrepancies between the training and testing data, which can be observed by comparing the 6 component ROCs in Fig. 5.14 (a) and (b).

The LLR fusion is able to achieve the statistically optimal results and does outperform all the other methods in some experiments presented above, because it has the strongest theoretic support. Nevertheless, we still emphasize three properties of the proposed decision fusion: (1) Good performance at lower complexity. For example, the LLR fusion method needs to learn the joint probability distribution of the training scores, either in a parametric or a non-parametric way, and the SVM fusion methods needs to learn the support vectors and their corresponding weights, all of which has high computational complexity. (2) Tolerance to overtraining. This is mainly due to the simplicity of the decision boundary, as well as the fact that we only work on the ROC operation points, which is already a reduced representation of the scores. (3) Insensitivity to outliers. This has been elaborated on in Section 5.4.1, and illustrated in Fig. 5.10, in which the outlier phenomenon is most pronounced.

For the decision-level fusion, we have implemented the optimization methods derived in Section 5.2, which is simple, but assumed independencies between the scores. Despite the certain degree of dependency between the two component scores, however, the final fused ROC on the testing data still demonstrate satisfactory performance. This can be explained by the fact that the main purpose of the proposed solutions is to find the optimal combination of thresholds which have the highest estimation of performance, instead of estimating the performance itself. In many dependent cases, the optimized thresholds are still plausible solutions, although the fused ROC is over-estimated. This is similar to the Naive Bayes classifier [44], which uses the independency assumption to estimate the class-conditional probabilities and then compare them. The estimated probabilities may very well be inaccurate, but the rank of them remains correct in many cases. The optimality of Naive Bayes classifier has been studied in literature [187] [43].

5.6 Summary

In this chapter, a new fusion method called threshold-optimized decision-level fusion is proposed [160] [165]. Both the theoretical analysis and the experimental results have been presented. In theory, the proposed decision fusion will always bring improvements over the original classifiers that are fused, and in practice, it also improves the system performance effectively, in a way comparable or even better than the conventional matching score fusion.

Fusion at decision level by AND and OR rule is not a popular practice, but we have shown that it can be done in an optimal manner, by optimizing the thresholds of component classifiers, such that it can be very beneficial. By threshold-optimized decision fusion, matching score normalization is not needed, and the component classifiers are automatically balanced through the optimization process in training, thus reducing the risk of performance degradation, when the component classifiers have significantly different performances. In this way the certain drawbacks related with AND/OR decision level fusion [39] can be avoided. It is also noteworthy that the optimization is only based on the limited number of operation points on the ROC instead of directly on the matching scores.

We have further introduced two scenarios in which the proposed decision fusion is especially useful. The first is the outlier scenario. The OR rule decision fusion is very robust to the outliers in the user class, and can easily achieve better performance than many score-level fusion methods. The second is the fusion of identical classifiers. This scenario is taken for the MPD face verification, in which a more reliable decision is made through fusing the decisions of consecutive frames. Good performance is demonstrated by the experimental results in Section 5.5.

Threshold-optimized decision-level fusion based on optimizing the ROC is an interesting fusion method both in theory and in practice. From a Neyman-Pearson point of view, the improvements brought by the proposed decision fusion on FAR (FRR) with respect to a fixed FRR (FAR) is always very desirable for any biometric system.

Chapter 6

Score Level Fusion

6.1 Introduction

[1]In the last chapter, we have introduced a novel fusion method at the decision level by optimizing the thresholds of the component classifiers. In this chapter, we will concentrate on the score-level fusion of biometrics, which is recognized as the most powerful level of fusion [133], offering the best tradeoff between information content and ease of fusion.

As introduced in Chapter 5, there are three types of fusion schemes at the score level [133]. The first type of fusion scheme is transformation-based. Firstly all the component matching scores are transformed (or normalized) so that they are on a comparable scale. Then simple scalar functions are applied on the transformed matching scores, resulting in a new matching score. Examples are product, sum, mean, max, etc. [89]. It can be proved that under certain ideal situations, for example, taking the product of independent likelihood ratios, can achieve the optimal performance in the Neyman-Pearson sense. The second type of fusion scheme is density-based. It relies on the estimation of the joint densities of the matching scores, and the fusion is done by statistical tests, such as the likelihood ratio test [35, 76] according to the user score and imposter score distributions. This type of fusion scheme achieves good performance if the densities can be well learnt, given that a large number of representative training matching scores are available. The third type of fusion scheme is classifier-based. It concatenates the component matching scores as a new feature vector,

[1] This Chapter is based on the publication [162], [163].

and train additional classifiers on them. Examples are neural networks [182], support vector machines [141], decision trees [132]. This type of fusion needs to train an extra classifier, therefore the performance is dependent on the specific training set of matching scores.

The proposed optimal likelihood ratio based fusion fits into all the three types listed above. It is transformation-based because the way to derive the likelihood ratio from the score can be seen as a well-designed transformation. It is density-based because the resulting likelihood ratio is closely related to density. It is classifier-based because the likelihood ratio of the score is an optimal statistic for score classification in the Neyman-Pearson sense [174]. The biggest advantage of the proposed method over the methods in literature, however, is that it avoids the often inaccurate estimation of the genuine and impostor score probability density functions. Instead, we directly map the matching score to its LLR value. The complexity, difficulty, and inaccuracy involved for density estimation are thus avoided, while the robustness and flexibility are gained because of the mapping strategies we use [163].

We further propose a hybrid fusion framework [162], which combines the proposed LLR based score-level fusion with the OR rule fusion at the decision level. The benefit of introducing the hybrid fusion is that it is able to improve the performance of the proposed fusion method even further, especially when there are outliers in the matching scores, because the OR rule fusion brings additional robustness. We will demonstrate this by the comparison experiments in Section 6.5.

This chapter is organized as follows. Section 6.2 introduces the optimal likelihood ratio based fusion theory, and Section 6.3 presents the methods of estimating the mapping from the score to the likelihood ratio. Section 6.4 introduces the hybrid fusion method. Section 6.5 gives experimental results of the proposed methods and the hybrid fusion based on the proposed methods, and compares them with other fusion methods. Section 6.6 sums up this chapter.

6.2 Optimal Likelihood Ratio Based Fusion

6.2.1 The LLR and the ROC

The receiver operation characteristic (ROC) is a commonly accepted measure of the verification performance. It denotes the detection rate p_{d} as a function of the false acceptance rate α. Theoretically, p_{d} and α are computed from the genuine and impostor score distributions ϕ_g and ϕ_i

$$\alpha(t) = \int_t^\infty \phi_\mathrm{i}(s)\mathrm{d}s \tag{6.1}$$

$$p_\mathrm{d}(t) = \int_t^\infty \phi_\mathrm{g}(s)\mathrm{d}s \tag{6.2}$$

where t is the application-defined threshold. The ROC is then expressed by the curve $\{\alpha(t), p_\mathrm{d}(t)\}_{t=t_\mathrm{min}}^{t_\mathrm{max}}$.

Taking the derivative of the ROC, we have

$$\frac{\mathrm{d}p_\mathrm{d}}{\mathrm{d}\alpha} = \frac{\frac{\mathrm{d}p_\mathrm{d}}{\mathrm{d}t}}{\frac{\mathrm{d}\alpha}{\mathrm{d}t}} = \frac{\phi_\mathrm{g}(t)}{\phi_\mathrm{i}(t)} \tag{6.3}$$

and this is, by definition, the likelihood ratio of the matching score at $s = t$. Therefore, it follows for the log-likelihood ratio

$$l(s) = \log(\frac{\mathrm{d}p_\mathrm{d}}{\mathrm{d}\alpha})\bigg|_{\alpha=\alpha(s)} \tag{6.4}$$

This result implies that, if the ROC $p_\mathrm{d}(\alpha)$ is known, the log log likelihood ratio of the matching score s can be computed by (6.4) without first estimating $\phi_\mathrm{g}(s)$ and $\phi_\mathrm{i}(s)$.

In practice, the ROC is estimated from a set of genuine and imposter matching scores, by first comparing the matching scores with the application-defined threshold t, and then counting the ratio of falsely and correctly accepted samples. The resulting ROC is in the form of a set of discrete points: $\{\alpha(t), p_\mathrm{d}(t)\}_{t=t_\mathrm{min}}^{t_\mathrm{max}}$.

As an example, Fig. 6.1 shows a ROC from the simulated genuine and impostor matching scores. In this example, we assume that the genuine scores have a Gaussian distribution of $N(1, 1)$, and the impostor $N(-1, 1)$. The operation points corresponding to 3 different thresholds $t = -0.8$, $t = 0$ and $t = 0.8$ are marked to illustrate the relationship between different thresholds and operation points.

6.2.2 LLR-Based Fusion

Let $l_1(s_1), ..., l_N(s_N)$ denote the log-likelihood ratios of statistically independent matching scores $s_1, ..., s_N$, respectively. The log likelihood ratio of the fused

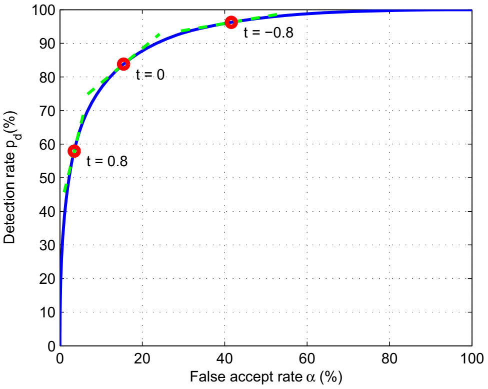


Figure 6.1: ROC of the simulated scores. The example operation points related to different thresholds t and the derivative at the points are also shown.

biometric $s = [s_1 \; ... \; s_N]$ is then given by

$$\begin{aligned} l(\vec{s}) &= \log \frac{\phi_{\mathrm{g}}(\vec{s})}{\phi_{\mathrm{i}}(\vec{s})} \\ &= \log \frac{\prod_{k=1}^{N} \phi_{\mathrm{g},k}(s_k)}{\prod_{k=1}^{N} \phi_{\mathrm{i},k}(s_k)} \\ &= \log \prod_{k=1}^{N} \frac{\phi_{\mathrm{g},k}(s_k)}{\phi_{\mathrm{i},k}(s_k)} \\ &= \sum_{k=1}^{N} l_k(s_k) \end{aligned} \tag{6.5}$$

Notice that (6.5) can be seen as a normalized sum rule, with $l_N(s_N), ..., l_N(s_N)$ the normalization functions.

The independency assumption is satisfied in many cases, especially when biometrics of different modalities (e.g. face, fingerprint, iris) are fused. In most cases when this assumption is not strictly satisfied, the LLR-based fusion in (6.5) still yields nearly optimal performance. This is similar to the Naive Bayes problem [44], which also assumes independency between different features, but whose optimality in dependency cases has been acknowledged in a wide range of applications [187][43].

6.3 Estimation by Fitting

6.3.1 Robust Estimation of the Derivative

The theory of estimating the LLR, as introduced in Section 6.2.1, is clear from the mathematics point of view. In practice, however, care must be taken in calculating the mapping $l(s)$. The reason is as follows. Firstly, instead of a smooth function, the ROC is a set of discrete points empirically derived from the training matching scores, and possibly contains noise. Secondly, taking the derivative of the ROC tends to amplify the noise that already exists. To overcome this problem, we use a smooth fitting method for estimating the $l(s)$ in a simple and robust manner.

In theory, the ROC is a concave function on the $\alpha - p_{\mathrm{d}}$ plane [49], as shown in Fig. 6.1. Therefore, the mapping $l(s)$ in (6.4) should be a monotonically increasing function. To guarantee this property, in our estimation of ROC, we always take the concave hull of the estimated ROC points $\{\alpha(t), p_{\mathrm{d}}(t)\}_{t=t_{\min}}^{t_{\max}}$ as a preprocessing of the ROC, before taking its derivative. This avoids the amplification of the noise on the estimated ROC caused by taking its derivative.

Another important problem is how to estimate the LLR for a certain $s = t$ on the ROC. This can of course be done by calculating the differences of neighboring points in a discrete way, but we choose to make this estimation more robust by first fitting a smooth (continuously differentiable) function in the neighborhood of this operation point, and then calculating the derivative of the function at the point. Fig. 6.2 illustrates the method in a small region of the ROC in Fig. 6.1. The derivative at the center operation point is estimated by first fitting with a second order polynomial, and then take its derivative.

Now that the LLR at a certain s is obtained, the remaining problem is how to estimate the continuous mapping $l(s)$. As we can obtain the LLR at any s, an obvious way is to calculate the mapping using the above mentioned method at every s with sufficiently fine scale, constituting a look-up dictionary. This method, however, is not only calculation-insensitive, but also noise-sensitive as the estimation of a point is local to its neighborhood on the ROC. Besides, the derivative of the ROC becomes inaccurate at extremely small or large thresholds, due to the insufficient number of the samples to reliably estimate the ROC operation points at the tail of distribution, as indicated by Fig. 6.3 (a). We again use the parametric fitting method to estimate the mapping of $l(s)$.

It is worth noting that the $l(s)$ function is a monotonically increasing function, so it is important that this property be satisfied in the parametric fitting. To give sufficient robustness as well as flexibility of the fitted $l(s)$, we propose

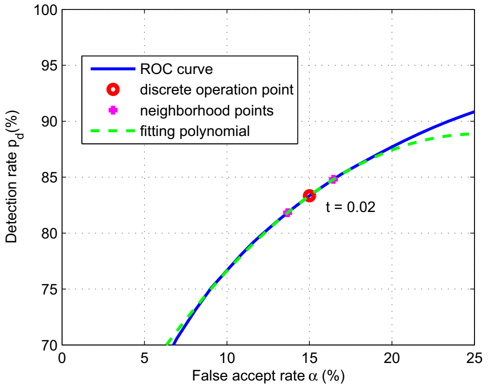
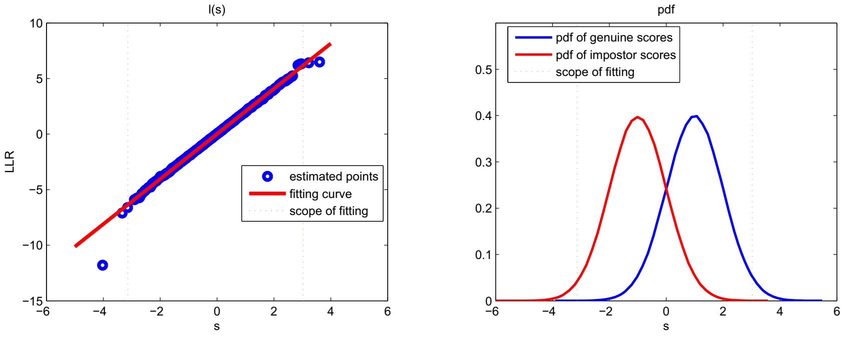


Figure 6.2: The way to calculate the derivative at an operation point by fitting within its neighborhood. In this example, a second order polynomial function is adopted.



(a) (b)

Figure 6.3: (a) Parametric fitting of the mapping $l(s)$, with the fitting scope shown. (b) Illustration of the fitting scope with respect to the score distribution functions.

to use the piecewise polynomial fitting. The method fits the function with piecewise ”smooth” low-order (e.g., 1-3) polynomials, where ”smooth” means continuity of the value as well as derivatives within the fitting scope. In this way the local smoothness and global flexibility of the curve are satisfied simultaneously. In simple situations, a single polynomial function would be sufficient to describe the mapping function $l(s)$, for example in the Gaussian distribution case as can be strictly proved. The mathematics of the fitting will be presented in the following. The minimum square error criterion is used to estimate the fitting parameters.

6.3.2 Robust Estimation of the Mapping

Single Polynomial Fitting

Suppose we use a p-order polynomial to fit a set of matching scores $s_1, ..., s_N$ and their corresponding LLR values $l_1, ..., l_N$, let $\mathbf{x} = [c_p, ..., c_1, c_0]^T$ be the unknown parameters, where c_i is the ith polynomial coefficient, $i = 0, ..., p$. The problem can be formulated as

$$\begin{pmatrix} s_1^p & s_1^{p-1} & \dots & s_1 & 1 \\ s_2^p & s_2^{p-1} & \dots & s_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_N^p & s_N^{p-1} & \dots & s_N & 1 \end{pmatrix} \mathbf{x} = \begin{pmatrix} l_1 \\ l_2 \\ \vdots \\ l_N \end{pmatrix} \tag{6.6}$$

To avoiding overtraining, the number of sample points N is normally much larger than the fitting order p, and $\mathbf{x}$ is taken as the least square solution of (6.6).

Piecewise Polynomial Fitting

Suppose we have N matching scores $s_1, ..., s_N$ and N corresponding LLR values $l_1, ..., l_N$. Assume the matching scores range from $s_{\min}$ to $s_{\max}$, and three pieces are taken from this range: $[s_{\min}, t_1]$, $[t_1, t_2]$, $[t_2, s_{\max}]$, where $s_{\min} < t_1 < t_2 < s_{\max}$[2].

In the following derivation, we take polynomials of the order 2. Higher order piecewise-polynomials can also be used, with similar mathematics derivation, but too high orders are not recommended because they possibly cause overfitting

[2] More pieces can be taken, and the conjoining points t_i can be taken in any defined manner (e.g. uniform intervals). Similar derivations follow.

and oscillation. Let the three polynomial function be $F_1(s)$, $F_2(s)$, and $F_3(s)$, then the fitting error is to be minimized

$$\sum_{s_{\min} \leq s_i \leq t_1} (F_1(s_i) - l_i)^2 + \sum_{t_1 < s_j \leq t_2} (F_2(s_j) - l_j)^2 + \sum_{t_2 < s_k \leq s_{\min}} (F_3(s_k) - l_k)^2 \tag{6.7}$$

As defined previously, smooth piecewise polynomial fitting means continuity of the value as well as derivatives within the fitting scope. Then the following equations should be satisfied at the conjoining points t_1 and t_2

$$F_1(t_1) = F_2(t_1), \qquad F_2(t_2) = F_3(t_2) \tag{6.8}$$

$$F_1'(t_1) = F_2'(t_1), \qquad F_2'(t_2) = F_3'(t_2) \tag{6.9}$$

This is a second order optimization problem with constraints, and can be formulated into a standard quadratic programming problem with respect to the polynomial coefficients of the three piecewise functions. A standard quadratic programming problem is written as follows [114]

$$x = \arg\min\left(\frac{1}{2}x^T Hx + fx\right),$$

$$\text{subject to one or more conditions: } A_1 x \leq b_1, \quad A_0 x = b_0 \tag{6.10}$$

The unknowns are the nine polynomial coefficients of the three functions. Let us put them into one coefficient vector

$$\mathbf{x} = [c_{1,2}, c_{1,1}, c_{1,0}, c_{2,2}, c_{2,1}, c_{2,0}, c_{3,2}, c_{3,1}, c_{3,0}]^T \tag{6.11}$$

where for each $c_{i,j}$, the first subscript i denotes the function number, and the second subscript j denotes the coefficient order. It is easy to transform the 9-coefficient vector back to the 3-coefficient vector of the three individual functions via a 3 × 9 matrix

$$[c_{1,2}, c_{1,1}, c_{1,0}]^T = \mathbf{P}_1\mathbf{x}, \qquad \mathbf{P}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$[c_{2,2}, c_{2,1}, c_{2,0}]^T = \mathbf{P}_2\mathbf{x}, \qquad \mathbf{P}_2 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

$$[c_{1,2}, c_{1,1}, c_{1,0}]^T = \mathbf{P}_3\mathbf{x}, \qquad \mathbf{P}_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

To deal with the three piecewise polynomials, partition the input matching scores into three ranges $[s_{\min}, t_1]$, $[t_1, t_2]$, $[t_2, s_{\max}]$, and assume the scores in the first range be $\{s_1, ..., s_{N_1}\}$, in the second range be $\{s_{N_1+1}, ..., s_{N_1+N_2}\}$, in the third range be $\{s_{N_1+N_2+1}, ..., s_{N_1+N_2+N_3}\}$, where $N_1 + N_2 + N_3 = N$. Define three matrices in similar form as in (6.6), each matrix with three columns, in the second order polynomial case

$$\mathbf{S}_1 = \begin{pmatrix} s_1^2 & s_1 & 1 \\ \vdots & \vdots & \vdots \\ s_{N_1}^2 & s_{N_1} & 1 \end{pmatrix}$$

$$\mathbf{S}_2 = \begin{pmatrix} s_{N_1+1}^2 & s_{N_1+1} & 1 \\ \vdots & \vdots & \vdots \\ s_{N_1+N_2}^2 & s_{N_1+N_2} & 1 \end{pmatrix}$$

$$\mathbf{S}_3 = \begin{pmatrix} s_{N_1+N_2+1}^2 & s_{N_1+N_2+1} & 1 \\ \vdots & \vdots & \vdots \\ s_N^2 & s_N & 1 \end{pmatrix}$$

Likewise, define three vector of the LLR values

$$\mathbf{l}_1 = \begin{pmatrix} l_1 \\ \vdots \\ l_{N_1} \end{pmatrix}$$

$$\mathbf{l}_2 = \begin{pmatrix} l_{N_1} \\ \vdots \\ l_{N_1+N_2} \end{pmatrix}$$

$$\mathbf{l}_3 = \begin{pmatrix} l_{N_1+N_2+1} \\ \vdots \\ l_N \end{pmatrix}$$

The function to minimize, (6.7), can be rewritten as

$$\| \mathbf{S}_1\mathbf{P}_1\mathbf{x} - \mathbf{l}_1 \|^2 + \| \mathbf{S}_2\mathbf{P}_2\mathbf{x} - \mathbf{l}_2 \|^2 + \| \mathbf{S}_3\mathbf{P}_3\mathbf{x} - \mathbf{l}_3 \|^2$$

Let $\mathbf{Q}_1 = \mathbf{S}_1\mathbf{P}_1$, $\mathbf{Q}_2 = \mathbf{S}_2\mathbf{T}_2$, $\mathbf{Q}_3 = \mathbf{S}_3\mathbf{T}_3$. By extending this function and with reference to (6.10) we obtain

$$H = \mathbf{Q}_1^T\mathbf{Q}_1 + \mathbf{Q}_2^T\mathbf{Q}_2 + \mathbf{Q}_3^T\mathbf{Q}_3 \tag{6.12}$$

$$f = -\mathbf{Q}_1^T\mathbf{l}_1 - \mathbf{Q}_2^T\mathbf{l}_2 - \mathbf{Q}_3^T\mathbf{l}_3 \tag{6.13}$$

The smoothness at the conjoining points can be formulated as the equality constraints. Put the two conjoining points t_1 and t_2 into two vectors $\mathbf{t}_1 = [t_1^2\ t_1\ 1]^T$ $\mathbf{t}_2 = [t_2^2\ t_2\ 1]^T$, then (6.8) can be written as

$$\mathbf{t}_1^T\mathbf{P_1x} = \mathbf{t}_1^T\mathbf{P_2x}, \qquad \mathbf{t}_2^T\mathbf{P_2x} = \mathbf{t}_2^T\mathbf{P_3x}$$

The first order derivative can be obtained via the following matrix

$$\mathbf{D} = \begin{pmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Using D, (6.9) is then written as

$$\mathbf{t}_1^T\mathbf{DP_1x} = \mathbf{t}_1^T\mathbf{DP_2x}, \qquad \mathbf{t}_2^T\mathbf{DP_2x} = \mathbf{t}_2^T\mathbf{DP_3x}$$

Referring to (6.10), we have

$$A_0 = \begin{pmatrix} \mathbf{t}_1^T\mathbf{P_1} - \mathbf{t}_1^T\mathbf{P_2} \\ \mathbf{t}_2^T\mathbf{P_2} - \mathbf{t}_2^T\mathbf{P_3} \\ \mathbf{t}_1^T\mathbf{DP_1} - \mathbf{t}_1^T\mathbf{DP_2} \\ \mathbf{t}_2^T\mathbf{DP_2} - \mathbf{t}_2^T\mathbf{DP_3} \end{pmatrix} \tag{6.14}$$

$$b_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{6.15}$$

So far we have completely converted the smooth piecewise polynomial fitting problem into a standard quadratic programming problem as in (6.10), with H, f, A_0, b_0 calculated by (6.12), (6.13), (6.14), (6.15). The problem is then solved using the standard quadratic programming methods. Further reference of quadratic programming can be found in [114].

A simple example of the parametric fitting of the mapping $l(s)$ is illustrated in Fig. 6.3, and some more complicated examples will follow in the next section. In this example, the same simulated data as in Fig. 6.1 is used. The discrete points are estimated from the ROC derivatives using the aforementioned method. To exclude the unreliable points at extremes, we restrict the fitting scope to be within a reasonable range of s, defined by $\alpha(s) < 1 - \epsilon$ and $p_{\mathrm{d}}(s) > \xi$, where ϵ and ξ are small quantities. For example, when we have M positive scores and N negative scores, ϵ can be set at $\frac{3}{N}$ and ξ at $\frac{3}{M}$, because we assume that the α and p_{d} estimated with less than 3 samples are unreliable. It can be observed from Fig. 6.3 (b) that the fitting scope is the region where the genuine and impostor score overlap. Actually, outside this overlapping scope, the estimation of the LLR is not critical any more, or in other words, classification can be very reliably done for very large or very small matching scores. As $l(s)$ is monotonic increasing, we fit outside this scope with a linear function adapting the slope at the ends of polynomials.

By using such parametric fitting, the LLR of any s can be easily calculated from a small number of parameters. The minimum square error (MSE) criterion, furthermore, makes the estimation robust because it optimizes on the global fitting scope. We have observed that most often 2-4 polynomials are sufficient to represent the flexibility of the $l(s)$ function. In Fig. 6.3, for example, the genuine and impostor scores are two gaussian distributions with identical covariances, therefore, the $l(s)$ function can be proved to be linear, i.e. with the polynomial order of 1. The parametric fitting, therefore, is very suitable to account for the degree of freedom of such a function.

6.3.3 Visualization of the Decision Boundary

It is interesting to investigate the decision boundary of our proposed method in the matching score space. To illustrate this we give both simulated and realistic examples. In the simulated example, the first biometric has the the genuine score distribution of $N(1,1)$, and impostor score distribution of $N(-1,1)$; the second biometric has the the genuine score distribution of $N(1,3)$, and impostor score distribution of $N(-2,2)$. The fitting functions, scatter plot and boundaries, and the ROCs are given in Fig. 6.4.

Since the two classes have Gaussian joint distributions with different covariances, the theoretically optimal decision boundary should be a second-order polynomial. Our optimal-LLR fusion method by polynomial fitting, therefore, is able to yield the ideal shape of boundary. By calculating the LLR values in the two dimensional score space using (6.4) and (6.5), we can obtain the LLR field and then derive the boundaries by drawing the contours of this field. Suppose we have two mapping functions $l_1(s_1)$ and $l_2(s_2)$, then the contours can be written in the mathematic form

$$l(s_1, s_2) = l_1(s_1) + l_2(s_2) = t \qquad t = \{t_1, ..., t_N\} \tag{6.16}$$

where t is a set of thresholds corresponding to different performance (FAR or FRR) requirements. Obviously, in the fitting scope, the shape of contour is determined by the parametric functions of $l_1(s_1)$ and $l_2(s_2)$. The implies that the robustness and flexibility of the mapping function is actually the robustness and flexibility of the decision boundary. Outside the fitting scope, the boundary is linear, which is discriminative enough for uncritical classification situations.

We give a more difficult example from the public database BA-fusion (Biometric Authentication Fusion Benchmark Database) [128] developed from the XM2VTS database [107], which contains the matching scores from face video and speech data. The matching scores are derived from various baseline systems (for details, see [128]). We show in Fig. 6.5 one example from the database, and demonstrate that our method works very well in difficult cases when the probability density is very difficult to estimate.

It can be observed that due to its particular distribution, the mapping $l(s)$ of the first matching score has more flexibility than any examples given before. For clarity we only show the estimated points within the fitting scope. Within this scope we partition the data into 3 segments. In the given example, the partitioning is uniform with respect the s axis, but it can also be defined otherwise. Under the global constraint that the curve is continuous and smooth, we fit each segment with a second order polynomial. Mathematics details are found in Section 6.3.2. The resulting curve exhibit both robustness and flexibility. In Fig. 6.5 (c) we can see that the resulting decision boundary is very well tuned to the distribution, while at the same time very smooth and robust, enabling better generalization capability of the boundary.

6.4 Hybrid Fusion

In order to further improve the performance of the proposed LLR based fusion, we incorporate it into the hybrid fusion framework, which combines the score-level and decision-level fusion and takes the advantage of both fusion modes [162]: the score-level fusion is advantageous because it is able to take care of the arbitrary score distributions, while the decision-level fusion is beneficial because it deals with outliers in the matching scores, which part cannot be accounted for by probability distribution functions [160].

The reason of introducing the hybrid fusion is that, although the LLR based method is plausible in the sense that it is density-based, in practice there are still sometimes outliers in the matching scores which cannot be modeled by the densities. The OR rule decision level fusion is especially suitable for handling such situations, as proved in [162]. Therefore, by hybrid fusion, further improved performance over the proposed LLR based fusion can be expected.

6.4.1 A Decision-Level Fusion Framework

The decision fusion framework has been proposed in Chapter 5, and the hybrid fusion is an extension of this work. Suppose we have N component ROCs $p_{\mathrm{d},i}(\alpha_i)$, $i = 1, ..., N$, derived from N independent classifiers. In practice, the OR rule fusion is of more interest, so we will work with the reject rate for the impostors, with $p_{\mathrm{r},i} = 1 - \alpha_i$ as the correct reject rate, and β_i as the false reject rate. Under the independency assumption, the operation points after fusion is

$$\beta = \prod_{i=1}^{N} \beta_i, \quad p_{\mathrm{r}}(\beta) = \prod_{i=1}^{N} p_{\mathrm{r},i}(\beta_i) \tag{6.17}$$

with β the false reject rate and p_{r} the correct reject rate of the OR rule fused decision. The optimization of the OR rule fused ROC, $p_{\mathrm{r}}(\beta)$, is done in such a way that at a fixed β, the p_{r} of the fused ROC is the highest. The optimality is thus in the Neyman-Pearson sense [174], formally formulated as

$$\hat{p}_{\mathrm{r}}(\beta) = \max_{\beta_i \mid \prod_{i=1}^{N} \beta_i = \beta} \left\{ \prod_{i=1}^{N} p_{\mathrm{r},i}(\beta_i) \right\} \tag{6.18}$$

which means that the resulting detection rate $\hat{p}_{\mathrm{r}}$ at β is the maximal value of the product of the detection rates at a certain optimal combination of β_i,

$i = 1, ..., N$, which satisfy $\prod_{i=1}^{N} \beta_i = \beta$. In other words, at a prefixed β, the highest p_r is obtained by optimizing (6.18). Consequently, the thresholds of component biometric systems can be readily obtained as the ones corresponding to the optimized operation points.

It is easily proved that the optimized correct reject rate $\hat{p}_\mathrm{r}(\beta)$ in (6.18) is never smaller than any of the component $p_{\mathrm{d},i}$, $i = 1, ..., N$, at the same β. The solutions of the optimization problem in and (6.18) are given in [160].

6.4.2 Score-level Fusion vs. Decision-level Fusion

Score-level fusion is the most popular way of fusion. The advantage of it is obvious. As a quantitative similarity measure it contains rich information about the biometric input, and yet it is still easy to process compared to sensor-level or feature-level data. In many cases, score-level fusion is able to achieve theoretically optimal performance. For example, taking product of the matching scores, which are independent and proportional to the likelihood ratio (in the feature space), is an ideal estimation of the joint likelihood ratio. Also, in the density-based score-level fusion [35], the ROC corresponding to the likelihood ratio statistic (in the matching score space), is optimal in the Neyman-Pearson sense.

A disadvantage of score-level fusion is that, because it works in the matching score space, it is subject to considerable flexibilities. For example, different normalization methods of the matching scores lead to different decision boundaries. Also, a too small training set of scores might easily overfits the data, especially in methods with flexible boundaries.

There are also advantages and disadvantages of the decision-level fusion. First of all, the framework is simple and clear from a mathematical point of view. Only a compact set of operation points is involved, and the Neyman-Pearson criterion is very beneficial for any biometric system. Besides, the optimization is not influenced by any score normalization, to which the ROCs are strictly invariant. Furthermore, the OR rule fusion is very suitable for many real world biometric applications, with outliers existent in the genuine class [160]. Basically, when the distributions of the genuine and impostor class are not symmetric, as is often true, the AND or OR decision fusion is very likely to fit because they have unsymmetrical support for the two classes.

The common criticism on decision-level fusion is that it has small and rigid information content. In the framework described in Section 6.4.1, however, the decision-level fusion has been adapted in such a way that the operation points are not fixed anymore, instead they are tunable and can be optimized with respect to performance. The disadvantage of decision-level fusion, nevertheless, is still the limited possibility of decision boundaries, because the operations are restricted to thresholding, AND, and OR.

6.4.3 Hybrid Fusion Scheme

In the general decision fusion framework, any two or more ROCs can be fused together. A biometric system, which has already been fused, can be easily put into this framework. This enables us to design a new hybrid biometric fusion scheme, combining score-level and decision-level fusion. Suppose the decision-level fusion can be expressed by

$$r_{\text{decision}} = D(r_1, ..., r_N) \tag{6.19}$$

where $r_1, ...r_N$ are the component ROCs to be fused, D is the decision fusion function, and r_{decision} is the resulting ROC. Similarly, suppose the score-level fusion is expressed by

$$r_{\text{score}} = S(r_1, ..., r_N) \tag{6.20}$$

where S is the score fusion function, and r_{score} is the resulting ROC. The general hybrid fusion function H is defined as

$$H(r_1, ..., r_N) = D\left(r_1, ..., r_N, S_1, ..., S_M\right) \tag{6.21}$$

where $S_1, ..., S_M$ denotes the ROCs of different score-level fusion methods.

In Section 6.4.1, we have assumed independency between the component ROCs. In hybrid fusion, however, the assumption is not satisfied, as the inputs in (6.6), $r_1, ..., r_N$ and $S(r_1, ..., r_N)$, are dependent. Strictly speaking, we have to go back to the matching score space, and take into account the joint probabilities of the component matching scores. For example, suppose we are fusing two classifiers with matching scores s_1 and s_2, with the genuine score distribution $p(s_1, s_2|\omega_1)$, and the impostor score distribution $p(s_1, s_2|\omega_0)$. The optimization at decision level, in the Neyman-Pearson sense, is

$$\hat{p}_{\text{d}}(\alpha) = \max_{t_1,t_2}\left\{\int_{t_1}^{\infty}\int_{t_2}^{\infty} p(s_1, s_2|\omega_1)\text{d}s_1\text{d}s_2)\right\} \tag{6.22}$$
$$\text{subject to} \quad \int_{t_1}^{\infty}\int_{t_2}^{\infty} p(s_1, s_2|\omega_0)\text{d}s_1\text{d}s_2 = \alpha$$

There are methods to solve (6.22), however, in practice we found that the independency assumption, i.e., solving (5.2) to obtain the thresholds corresponding to the optimal α_i's, is just adequate. The independency assumption might change the estimation of $\hat{p}_{\mathrm{d}}(\alpha)$, but the thresholds t_1 and t_2 corresponding to its maximal value is often unchanged, or close enough to the real t_1 and t_2 under the dependent assumption. Actually, we have observed that in many cases, the results from independency assumption is even better than the results from the dependency solutions. This can be explained by that fact that the optimization problem in (6.22) has much larger complexity than (5.2) and therefore more prone to overfit the solutions to the specific training set of matching scores.

Solving the hybrid fusion using the ROCs, instead of the matching scores, not only preserves the simplicity of the method, but also makes the solution more robust to the deviations between the training and testing scores. We summarize the hybrid fusion method as follows:

1. Given a set of component matching scores, and a set of score-level fusion methods.

2. (Training) Derive individual ROCs from the component matching scores and the score-level fused matching scores. Fuse all the ROCs under the fusion framework by the AND rule (5.2) or OR rule (6.18), and obtain the optimal combination of operation points.

3. Obtain the thresholds corresponding to those optimized operation points.

4. (Testing) Apply the trained thresholds on the component matching scores the score-level fused matching scores, and fuse the decisions by the AND rule or OR rule as the final decision.

So far we have introduced the general hybrid fusion of multiple classifiers. Fig. 6.6 gives an example of the diagram of the hybrid fusion between two component classifiers. In this framework, the LLR based decision-level fusion is combined with the OR rule decision level fusion.

6.5 Experiments and Results

With the proposed optimal LLR based fusion, we combine the two-dimensional face texture and three-dimensional face shape information for improved face recognition performance. The context of this work is the EU FP6 3D-Face project [1], which aims to achieve reliable biometric authentication using the face in its two-dimensional and three-dimensional modalities. The first database that the face recognition algorithms were developed on is the FRGC database [124], which contains the 2D face texture and 3D face shape data collected simultaneously. The database contains data of 465 subjects and has in total 4,007 samples. The classifiers that produce the matching scores are trained on 309 subjects in the database. To train fusion, another 100 subjects are taken to obtain the matching scores from the trained classifier, resulting in 25,520 genuine scores and 2,568,190 impostor scores (fusion training data). The remaining 56 subjects are used for evaluation, resulting in 12,270 genuine scores and 700,910 impostor scores (fusion testing data). In all the following experiments, we train the mapping by the fusion training data, and evaluate on the fusion testing data.

For either modality, the matching scores are derived and provided by L-1 Identity Solutions (L1), Cognitec Systems (COG), and the University of Twente (UTW). In the L-1 method, the matching scores are computed using the hierarchical graph matching (HGM) methods [69], which represents the facial geometry by means of a flexible grid. Similar to the biological structures in the human brain, a set of specific filter structures is assigned to each node of the graph and analyzes the local facial characteristics [70] [184]. With HGM, approximately 2,000 characteristics are used to represent a face and an individual identity. For the analysis of a face, the shape (”landmarks”) and the structure (”features”) of the face are separated, making HGM a very robust facial recognition method providing a basis for both 2-D and 3-D face recognition. In the COG method, for 2D faces, the feature components are retrieved by applying local image Gabor transforms at facial feature locations. These component are then concatenated to form the raw 2-D face feature vector. For 3-D faces, the face shape is firstly registered and smoothed to form the raw 3-D face feature vector. Global transformations are applied on the raw feature vectors in both cases, in order to maximize the ratio of inter-personal variance to intra-personal variance [108]. The final scores are obtained by simple similarity measures of the transformed feature vectors. In the UTW methods, holistic approach is taken, and the feature vectors are derived by the conventional PCA and LDA transformation, and the scores are computed as the likelihood ratio of the feature vector in the feature space. More details of the mathematics can be found in [6].

We applied the proposed LLR based fusion, and the hybrid fusion of the LLR based score-level fusion and OR rule decision-level fusion on the biometric scores, as illustrated by Fig. 6.6. Comparison of the performance is done with the following three other fusion methods:

1. Sum Rule

   Before applying the sum rule, the scores are transformed using the Z-normalization [133], which normalizes the genuine or impostor scores to unit variance.

2. Likelihood Ratio by GMM

   The joint density of the matching scores is firstly estimated using Gaussian mixture models (GMM) [51], using the method in [129]. Then the likelihood ratio is calculated based on the estimation of both genuine and impostor score distributions.

3. SVM

   Taking the concatenated component matching scores as a feature vector, we use SVM to do the classification. The decision boundary is trained using the radius basis function (RBF) kernels [30]. The scores are firstly Z-normalized with a variance of 1, and the RBF radius is chosen as 1. Implementation details can be found in [77].

For the purpose of comparison, we have calculated the full ROC, i.e., all the possible operation points, instead of a single operation point, to present the fusion performance on the entire range of α. We will show the fusion between the two modalities, and visualize the decision boundaries of different fusion methods in the two-dimensional matching score space, to give a clear view of how the methods work.

We show three representative examples of the 2D texture and 3D shape fusion. Fig. 6.7, Fig. 6.9, Fig. 6.11 illustrate the decision boundaries of different fusion methods, and Fig. 6.8, Fig. 6.10, Fig. 6.12 show the performance with respect to the ROCs. Note that the scatter plots of the matching scores in Fig. 6.7, Fig. 6.9, Fig. 6.11 are those of the training set, showing the fitting of the training of different fusion methods. In Fig. 6.8, Fig. 6.10, Fig. 6.12, (a) is the ROC of the trained fusion on the training data itself, while (b) is the ROC of the trained fusion on the testing data. The differences between the original ROCs (ROC1 and ROC2) in (a) and those in (b) indicate the discrepancies of the training and testing data distributions. The discrepancies is the underlying reason why we seek for smooth mapping $l(s)$ by least square solutions.

It can be observed from the figures that the simple sum rule with Z-score normalization is the weakest, and the other four advanced fusion methods, namely, SVM, LLR by GMM estimation, LLR by fitting, and hybrid fusion, all yield better performances. In the training, the four advanced methods produce close ROCs, indicating good fitting to the score data. In the testing, however, some difference are shown, indicating different capabilities for generalization. For example, it can be observed that in Fig. 6.10, our proposed method outperforms the LLR by GMM method, especially in the testing. Furthermore, the hybrid fusion based on the proposed LLR fusion yields even better performances in general.

The LLR by GMM estimation is the theoretically optimal fusion method, if we look at fusion from a classification point of view [132]. It uses the optimal LLR statistic and taking into consideration of the dependencies between the component scores. The GMM estimation from the scores, and the subsequent LLR calculation from the estimated densities, however, is much more expensive in computation than our direct mapping from the scores to their LLR values. Moreover, due to its many parameters to estimate and the limited number of training sample, the GMM estimation is likely to overfits the training data.

It can be observed that the decision boundaries produced by the sum rule are often under-trained, while those produced by the SVM and the LLR by GMM estimation are often over-trained, especially when there are outliers, i.e., extraordinary samples, in the training data. Consequently, the decision boundaries produced by SVM and GMM are likely to be overfitted due to the freedom offered by the classifier parameters. In comparison, the decision boundaries yielded by the proposed LLR by fitting is both robust and flexible, giving simple decision boundaries smoothly adapted to the score distributions.

Theoretically speaking, the mapping from the matching scores to the LLR values is a strictly monotonically increasing function. From this aspect, our method is particularly plausible, because this property has been guaranteed by using monotonic functions in the fitting. In the SVM or GMM method, however, this property is not guaranteed as they can still produce decision boundaries that violate the monotonicity, e.g. closed boundaries. In other words, the SVM or GMM methods are general classification models, which cannot take into consideration the special monotonic property of the matching scores. The merit of our method, therefore, lies in its simplicity by nature, because the key step of fitting of such a monotonic function $l(s)$, is low in complexity, as shown in Fig. 6.4 (a) (b), and Fig. 6.5 (a) (b) even when the distributions are complicated and unconventional. As a result, the LLR estimation by fitting the $l(s)$ function is simpler, and can be more reliably done, than estimating the probability densities or finding the support vectors. Moreover, the complexity of such a fitting, and hence the complexity of the resulting decision boundary, can be easily controlled by allowing different levels of freedom (i.e. parametric forms) on the fitting curves.

The drawback of our method lies in the independency assumption of the LLRs, and this explains why our method is sometimes outperformed by the GMM method. Nevertheless, this is not a serious problem in many biometric fusion applications. Firstly, different biometrics are very likely to be independent if they are acquired from different physical modalities. Secondly, even if the absolute values of the joint LLR, as calculated in (6.5), are not estimated accurately, the relative values of the joint LLR, i.e., the amplitude relationship between the LLR values in the score field, can still be correct. Analogy can be made to the Naive Bayes classifier [44] [187] [43]. Furthermore, incorporated with the hybrid fusion, the proposed LLR based fusion can yield further improved performances.

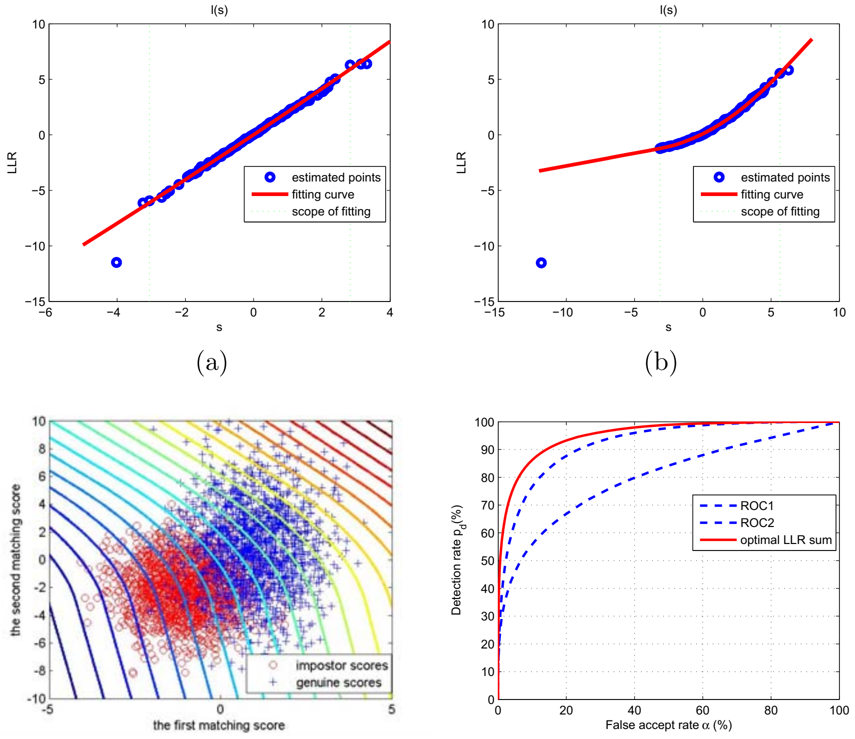
6.6 Summary

In this chapter we have proposed an optimal likelihood ratio based fusion method of biometric scores. The biggest merit of our method is that the solution is LLR-based, but the complicated, and often inaccurate, estimate of the genuine and impostor score probability density functions are avoided. Instead of calculating the LLR from the two estimated densities, we map the matching score s directly to its LLR value via the ROC. The complexity, difficulty, and inaccuracy involved for density estimation are thus avoided.

Parametric fitting is used to reliably estimate the derivative on the ROC, resulting in the mapping from a certain score to its corresponding likelihood ratio. Consequently, a number of discrete points in the score-likelihood ratio space are obtained. Then the continuous score-to-likelihood-ratio mapping is obtained by a second parametric fitting, which smoothly connects the set of discrete points obtained from the previous step. The fitting strategies of the mapping, piecewise polynomial fitting, make the function very robust to possible noise or outliers in the training scores, and flexible to arbitrary matching score distributions. We presented the mathematics, and show how robustness and flexibility are acquired by the mapping strategies.

We compared our methods with other popular score-level fusion methods, especially with the likelihood ratio method using density estimation by Gaussian Mixture Models. Simplicity and robustness are demonstrated for our method under a large range of score distributions.

Under the optimal decision-level fusion framework proposed in Chapter 5 and taking advantage of the score-level fusion in this chapter, we further proposed an interesting hybrid fusion scheme, which combines both decision level fusion and score level fusion. The score-level fusion is advantageous in the sense that it is able to take care of the arbitrary score distributions, while the decision-level fusion is beneficial when there are outliers in the matching scores, which part cannot be accounted by probability distribution functions. Consequently, further improved performance over the LLR-based fusion can be achieved. Experiments show that in different cases, with different matching score distributions, the hybrid fusion method is able to adapt for improved performance over the two levels of fusion.



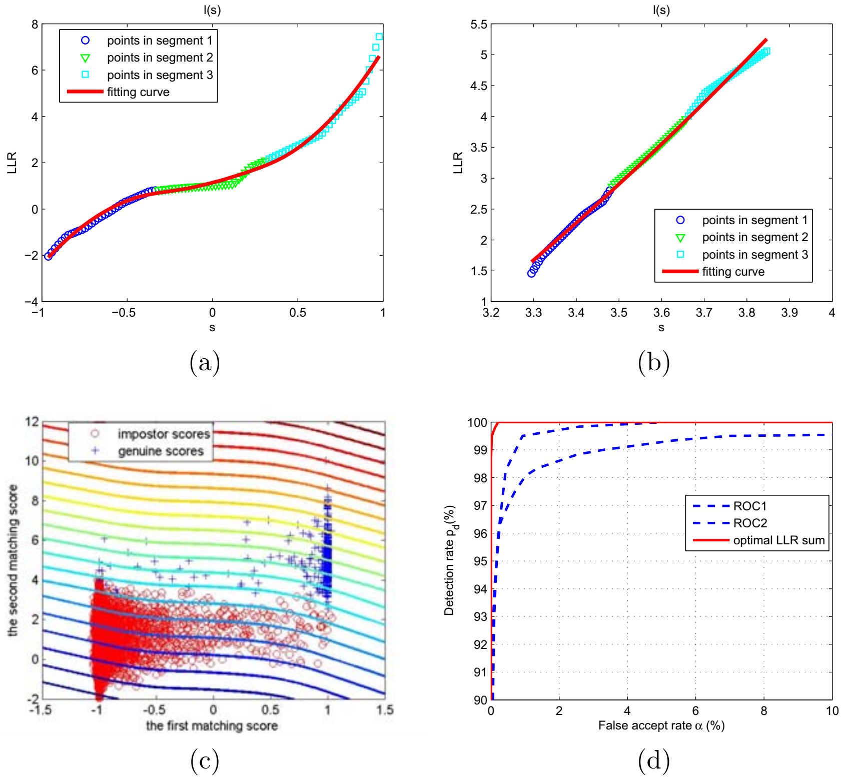
(a)

(b)

(c)

(d)

Figure 6.4: (a) Mapping $l(s)$ of the first score using second order polynomial; (b) Mapping $l(s)$ of the second score using second order polynomial; (c) Score distribution and the decision boundaries; (d) ROCs of the individual scores and of the optimal LLR fused scores.



(a)

(b)

(c)

(d)

Figure 6.5: (a) Mapping $l(s)$ of the first score using piecewise polynomial; (b) Mapping $l(s)$ of the second score using piecewise polynomial; (c) Score distribution and the decision boundaries; (d) ROCs of the individual scores and of the optimal LLR fused scores.

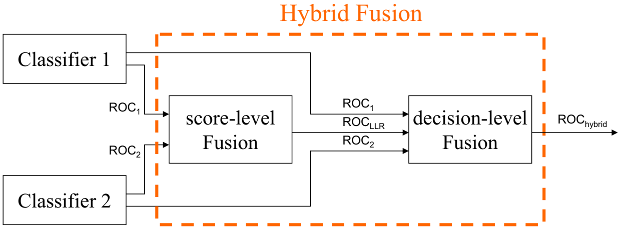
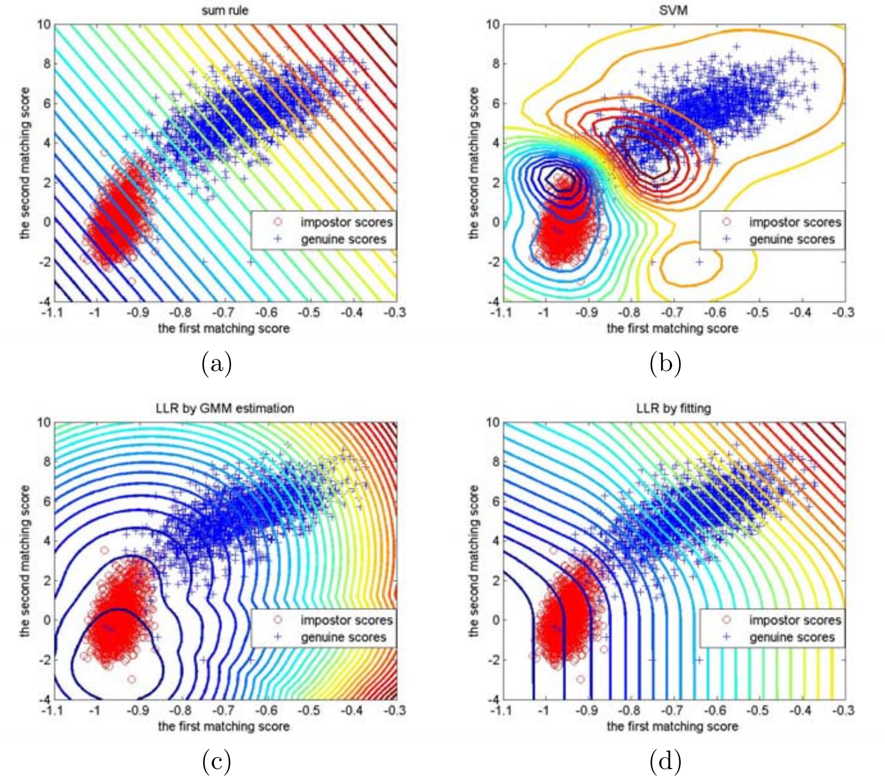
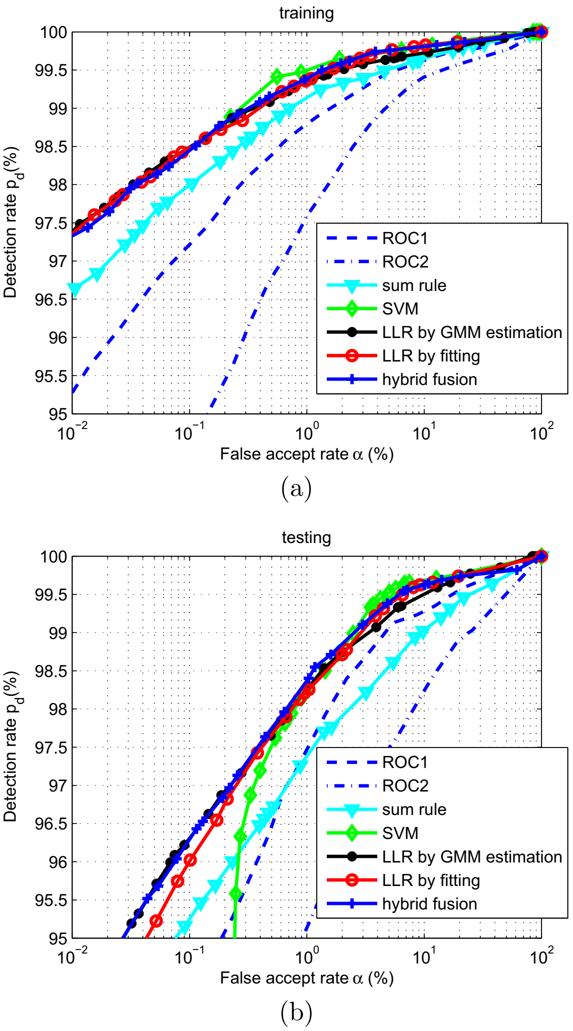


Figure 6.6: The diagram of hybrid fusion for two component classifiers.



(a) (b) (c) (d)

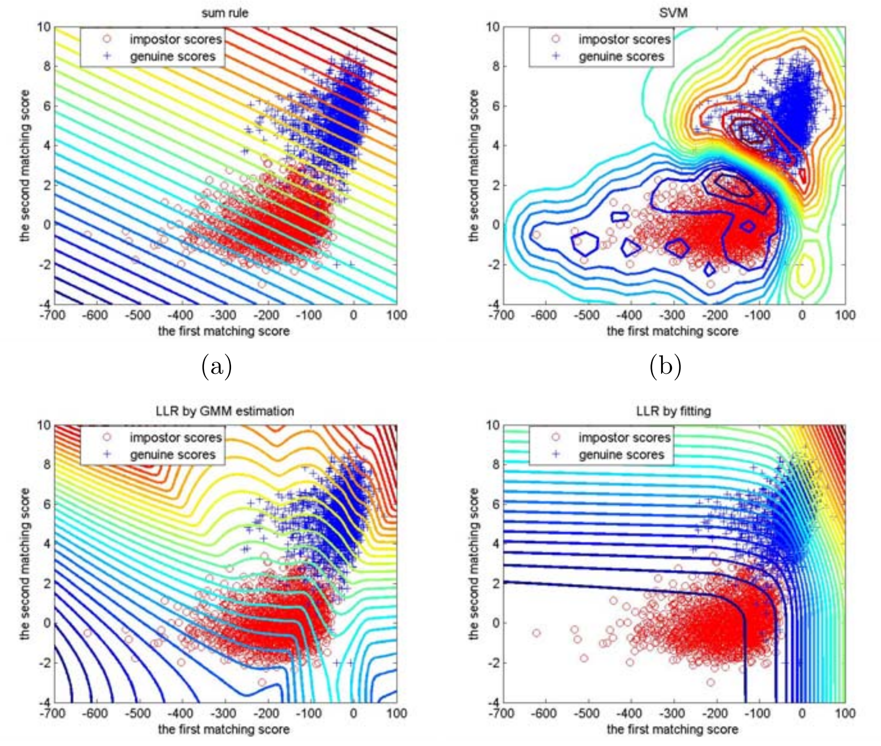
Figure 6.7: Fusion of the COG texture data and COG shape data: (a) Decision boundaries of the sum rule fusion with Z-normalization; (b) Decision boundaries of SVM fusion; (c) Decision boundaries of LLR fusion based on GMM estimation; (d) Decision boundaries of LLR fusion by fitting the $l(s)$ function, proposed in this chapter.



(a)

(b)

Figure 6.8: Fusion results of the COG texture data and COG shape data: (a) ROCs of the training set. (b) ROCs of the testing set. The fusion parameters are trained on the training set.



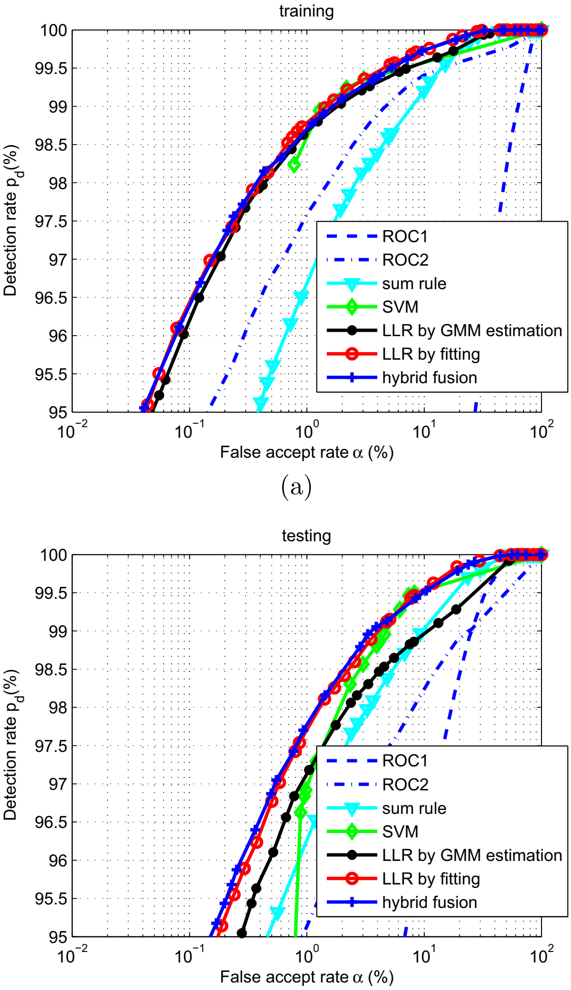
(a)

(b)

(c)

(d)

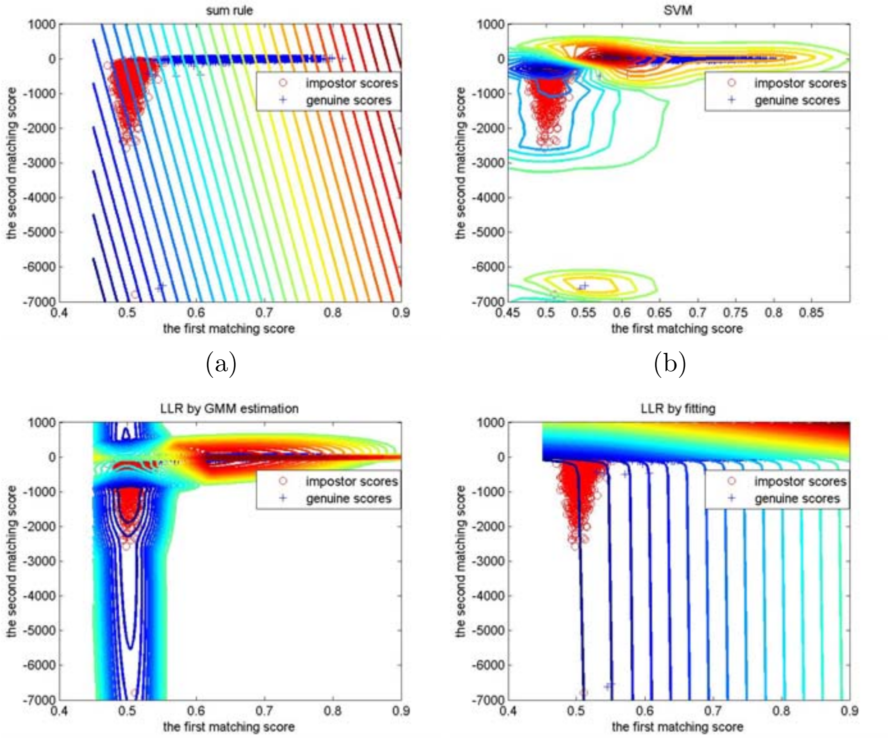
Figure 6.9: Fusion of the UTW texture data and COG shape data: (a) Decision boundaries of the sum rule fusion with Z-normalization; (b) Decision boundaries of SVM fusion; (c) Decision boundaries of LLR fusion based on GMM estimation; (d) Decision boundaries of LLR fusion by fitting the $l(s)$ function, proposed in this chapter.



(a)

(b)

Figure 6.10: Fusion results of the UTW texture data and COG shape data: Fusion results of the UTW texture data and COG shape data: (a) ROCs of the training set. (b) ROCs of the testing set. The fusion parameters are trained on the training set.



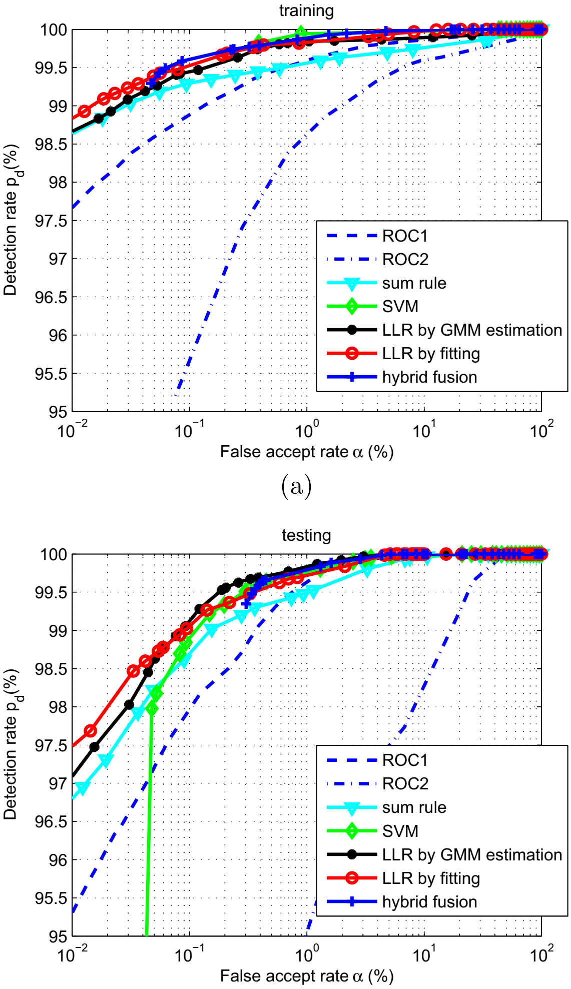
(a)

(b)

(c)

(d)

Figure 6.11: Fusion of the L1 texture data and UTW shape data: (a) Decision boundaries of the sum rule fusion with Z-normalization; (b) Decision boundaries of SVM fusion; (c) Decision boundaries of LLR fusion based on GMM estimation; (d) Decision boundaries of LLR fusion by fitting the $l(s)$ function, proposed in this chapter.



(a)

(b)

Figure 6.12: Fusion results of the L1 texture data and UTW shape data: (a) ROCs of the training set. (b) ROCs of the testing set. The fusion parameters are trained on the training set.

Chapter 7

Summary and Conclusions

7.1 Summary

In this thesis, we presented a detailed study of the face verification problem on the mobile device, covering every component of the system, as shown in Fig. 7.1. The study includes face detection, registration, normalization, and verification. Furthermore, the information fusion problem is studied to verify face sequences, and to fuse different modalities. Although the work is application-specific, the thesis is not limited to the application, but more extensive. In every step, we have justified the methods we choose both from the theoretical and the practical point of view. In the review part of each chapter, we discussed principles and methodologies on a higher level, for a better understanding of the problems in general. In our solutions, on the other hand, we have taken care of the application requirements, and put much emphasis on the efficiency and simplicity of the methods.

The face detection is done by the Viola-Jones method, which is fast in detection because of its easily scalable features and the cascaded structure. As introduced in Chapter 2, most of the effort is spent on the training stage for selecting the pool of Haar-like features and their corresponding weights. Once the training is done, these parameters can be fixed into the hardware device once and for all, and can be applied to any user as thanks to the robustness of the method.

For face registration, we proposed to first detect the facial landmarks, and then registrate the face to a standard orientation and scale. We trained 13 facial

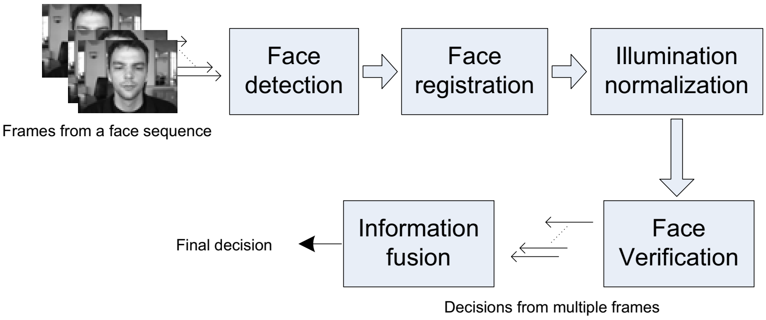

Frames from a face sequence
Face detection
Face registration
Illumination normalization
Face Verification
Decisions from multiple frames
Information fusion
Final decision

Figure 7.1: Diagram of the face verification system.

feature detectors by the specially tuned Viola-Jones method as presented in Chapter 2. A major problem in landmark-based registration is the unavoidable falsely detections. For this purpose, we proposed a very fast post-selection strategy, based on the error occurring model, which is accurate and specific to the detection method as well as to the objects. The proposed post-selection strategy does not introduce any statistical model or iteration step, instead, it only relies on the scale information and operates only once. Compared to many other registration methods that incorporate more complicated shape or texture models, and introduce iterative convergence, the method is very fast even on a mobile device.

For the illumination normalization problem, we discard the three-dimensional modeling methods, which are not only complicated in computation, but also too delicate to generalize to many scenarios. Instead, we implemented the simple and efficient two-dimensional preprocessing methods. As introduced in Chapter 4, the two illumination insensitive filters we proposed are the Gaussian derivative filter in the horizontal direction and the simplified local binary pattern as a filter. The two methods, especially the later, are computationally low-cost, and meanwhile exhibit a high degree of insensitivity to illumination variations. The implementation of the simplified local binary pattern is extremely fast.

In the verification stage, we proposed to use the likelihood ratio based classifier, which is statistically optimal in theory, and easy to implement in practice. On the mobile device, the enrolment can be done by taking a video sequence of several minutes. Above all, the method is chosen because the verification problem has a largely overlapping distribution of the classes, and therefore can be better solved by density-based methods than boundary-based methods. In Chapter 3, we have investigated the influence of various dimensionality reduction methods on the verification performance, and observed that using the full dimensionality of the properly reduced image is more beneficial than learning the dimensionality reduction from a large image. Besides, we have also compared the single Gaussian model and the Gaussian mixture model, and proved that the former has better performance, lower complexity, and higher generalization capability. Consequently, the verification measure, the likelihood ratio, can be reduced to the differences between two squared Mahalanobis distances. The computation involves only linear matrix manipulations and the complexity is low.

To achieve as good performance as possible, we fused the information in the time domain, i.e., the decisions from multiple face frames from a time sequence are fused. To do this, we studied the information fusion problem in Chapter 5 and Chapter 6, and proposed the threshold-optimized decision-level fusion, LLR-based score-level fusion, and hybrid fusion. The decision-level fusion is suitable for the situation in our system, because in theory we have proved that fusing identical classifiers does not even need the training. Furthermore, the proposed OR rule fusion is very suitable for the scenario when outlier data exist in the user class. This gives more accommodation to the user's poses and expressions, thus benefiting the user-friendliness form a system point of view. All the fusion methods are also successfully applied to the 3D-Face project, yielding good performance for fusing the two-dimensional face texture and three-dimensional face shape data.

The main contributions of the thesis are listed in the following:

- We extended the work of Viola and Jones on face detection to the detection of more unstable and vague objects: facial features. Novel error models are built up, circumventing the more complicated statistical shape models. Fast and robust facial feature detectors are built up based on this model.

- We gave some insight into the face detection and recognition problems, which share important common parts of feature extraction and classifier design, but differs substantially in class distributions. The unbalanced prospects of FAR and FRR are studied in the context of likelihood ratio based verification.

- We proposed to pursue illumination insensitivity, instead of invariance, of the face images using simple preprocessing methods. We proposed two illumination-insensitive filters, namely, the Gaussian second-order derivative filter and the simplified LBP filter, and justified the proposed methods under the likelihood ratio based verification framework.

- An interesting decision-level fusion method based on optimizing the ROC operational points is proposed. With this method, we do not have to deal with the large number of matching scores, but work on the operation points on the ROC. The optimization is in the Neyman-Pearson sense. The method is especially suitable for verification problem with outliers.

- A statistically optimal score-level fusion is proposed which is able to avoid the complicated, and often inaccurate, density estimation. The method maps the scores directly to their corresponding likelihood ratios, via the ROC. The method is robust and flexible thanks to the parametric fitting strategies we used.

- Under the optimal decision-level fusion framework, we proposed a hybrid scheme combining decision-level and score-level fusion, which takes advantage of both fusion modes. The score-level fusion takes care of the arbitrary probability distribution functions of the matching scores, while the decision-level takes care of the outliers, which part cannot be accounted by probability distribution functions. As a result, the hybrid fusion yields even improved performance over both.

7.2 Hardware Implementation

The efficiency and simplicity every step as shown by Fig. 7.1 enables realistic implementation of this system on a MPD. We chose the Eten M500 Pocket PC as a demonstrator, and transformed our algorithms that are written in the C language onto the Windows Mobile 5 platform of the device. We used the Intel OpenCV library [73] to facilitate the implementation of many functions.

In the initial trial, due to the limitation of the power and computation capability of the current mobile device, we still do the enrolment on the PC: the MPD takes a sequence of the user images of about 2 minutes and transfers them to the PC to process them, with the user mean and covariance extracted for calculating the Mahalanobis distance in the user class. The background mean and covariance have been pre-stored in the mobile device for calculating the Mahalanobis distance in the background class. Once the enrolment is finished, the mobile device are ready to use. The user image sequences then pass through the diagram in Fig. 7.1 till the final decision of accept or reject is made. The implementation of the system in the project framework has been reported in [41].

Even without optimization, our system has already achieved a frame rate of around 10 frames per second on the current laptop with Intel(R) CPU, 1.66GHz, 2GB of RAM. On the mobile device, with the Samsung S3C2440 400Mhz Processor and 64MB of SDRAM, the time is substantially longer, about 8 seconds a frame. Profiling of the system indicates that the face detection and registration are still the most time-consuming part, while the illumination normalization and likelihood ratio based verification are extremely fast. This system will become practical in use with further optimization both in hardware and software [41].

From a hardware point of view, there are several things of interest to try in the future. As a commercial attraction, nowadays the face detector has been implemented on some small electronic devices like the digital camera. This implies that the Viola-Jones face detector in the mobile system, which is much faster than we have now, is feasible if careful optimization is made in the implementation. For example, the selected Haar-like features can be calculated by specially mapped integrate circuits and facilitate much faster detection. In the same manner, even faster facial feature detectors in our optimized form are also realizable with pre-trained parameters and the fast post selection strategy. As pointed out in Chapter 2, our method enables a stand-alone facial feature detector. From that on, all the subsequent calculation are simple enough, involving only linear manipulations and comparisons.

Under harsh situations, like very dark or bright weather, the illumination might still cause problem for the verification. A more fundamental way to solve this problem is to use an alternative hardware camera, which is itself insensitive to illumination. For example, an infrared camera is interesting for this purpose as it is invariant to the visible lights.

A more intelligent system is achievable if more than one templates (i.e., mean and covariance) of the user are stored over time. That means the user can enrol the device at different time, under diverse scenarios, and store all the user information into the device. In operation, the device then do fusion on the decisions or matching scores from multiple verification results. The time information of the enrolment can be used as a weighting or forgetting parameter in fusion. In this process, the generalization capability of the verification system increases, so care must be taken not to degrade the discrimination capability at the meantime. In theory, this is possible as we are operating in a high dimensional space which possesses sufficient discrimination power.

7.3 Conclusions

Face verification on the mobile device provides a secure connection between the user and the personal network. In this thesis, we have proposed the solution for this face verification system, including face detection, registration, illumination normalization, verification, and fusion. Besides the high computational efficiency, the current system exhibits robustness to considerable face variability that is possible to occur in the face verification scenario. As the example given in Fig. 5.9, under the difficult test protocol in which the training and testing data are taken under completely different illuminations, with variations in expressions and poses, a detection rate of 75% can be achieved at the low false accept rate of 0.1%. With fusion between the time frames, the detection rate can reach 95% at the same false accept rate (see Fig. 5.9 (d)). When the illumination of the testing are same or similar to that of the training, the generalization is more easily done, and the system can reach an even higher detection rate.

The system has dealt with the security, convenience, and complexity requirements, which are put forward in Chapter 1. As discussed in Chapters 3 and 4, we verify the input pattern in a high-dimensional space, which is in theory sufficiently discriminative and guarantees security. For the user convenience, a low false rejection rate is essential, and this has been taken care of by the illumination normalization and the decision fusion. The time-sequence verification, furthermore, not only improves security, preventing the scenario that the device is taken away after being logged-on, but also increases the user-convenience by lowering the false rejection rate with the fusion rule. Finally, the algorithmic complexity has always been an important concern of our work, and the system has been successfully implemented on the mobile device with limited computational resources.

Apart from the face verification system on the mobile device in the PNP2008 project, the work has been further extended to the fusion of different face modalities in the European FP6 3D Face project. We have carried out a thorough study on fusion at different levels of the biometrics system, and proposed efficient fusion schemes respectively at the decision level, score level, and a combination of both. As can be observed from Fig. 5.10 - 5.14 and Fig. 6.10 - 6.12 in Chapters 5 and 6, the performance of the original system has be improved substantially even at a very low false acceptation rate. In this context as well as many other biometric applications, the integration of multiple biometrics is of great interest to achieve a more secure, reliable, and robust system.

Samenvatting

In dit proefschrift, presenteren wij een gedetailleerd onderzoek naar gezichtsverificatie op een mobiele telefoon of PDA, waarin we alle componenten van het systeem zullen behandelen. Ons onderzoek omvat gezichtsdetectie, gezichtsregistratie, gezichtsnormalisatie en gezichtsverificatie. Daarbij, hebben we ook nog voor opeenvolgende opnamen van gezichten gekeken naar het combineren van informatie uit deze opnamen ten behoeve van gezichtsverificatie. Ondanks dat ons werk applicatiespecifiek is, bevat het proefschrift algemenere oplossingen, die niet alleen gelden voor onze applicatie. We hebben alle gebruikte methoden verantwoord vanuit theoretisch en praktisch oogpunt. In het overzicht van elke hoofdstuk, bekijken we de principes en methodologie op een hoger niveau om de algemene problemen beter te snappen. Onze oplossingen voldoen aan de eisen die gesteld werden door de applicatie, waar het accent op effectieve en simpele methoden ligt.

De gezichtsdetectie wordt gedaan door de Viola-Jones methode, die snel is omdat deze methode meeschalende features en een cascade structuur gebruikt. In Hoofdstuk 2 laten we zien dat meeste inspanning in het trainen van deze methode gaat zitten, waarbij een set van zogenaamde 'Haar' features en corresponderend gewichten geselecteerd wordt. De instellingen, die gevonden zijn tijdens de training, kunnen worden overgenomen in hardware en worden gebruik voor elke gebruiker omdat de Viola-Jones methode zeer robuust is.

Voor het registreren van een gezicht, detecteren we eerst 'landmarks' (orientatiepunten in het gezicht) en daarmee registeren we het gezicht naar een standaard ori?ntatie en schaal. We hebben voor 13 'landmarks' special geoptimaliseerde Viola-Jones detectoren (Hoofdstuk 2) getraind. Een groot probleem in landmark gebaseerde registratie zijn de false detecties. Daarom hebben we een snelle extra strategie voorgesteld, die door middel van een model foute landmarks detecteert en verwijdert. Deze voorselectie strategie maakt geen gebruik van een statistisch model of iteratieve stappen. In vergelijking met ander regis-

tratie methoden die ingewikkelde modellen van vorm en textuur maken is onze methode erg snel, zelfs op de simpele processor van een mobiele telefoon.

Voor de belichtingsnormalisatie, hebben we geen drie dimensionale modellen waar veel rekenkracht voor nodig is en die moeilijk generaliseren gebruikt. In plaats daarvan hebben we simpele en effectieve twee dimensionale preprocessing methoden gebruikt. In Hoofstuk 4, worden door ons twee belichting ongevoelige filters voorgesteld, namelijk filteren met de afgeleide van de Gaussian in horizontale richting en 'simplified local binary patterns'. De twee methoden, vooral de tweede, gebruiken weinig rekentijd, terwijl ze een hoge graad van ongevoeligheid onder verschillende belichtingsvariaties demonstreren.

In de verificatiefase, gebruiken we de 'likelihood ratio' methode, die in theorie statistische optimaal moet zijn en in praktijk gemakkelijk te implementeren is. We kunnen op de mobiele telefoon een gebruikerstemplate aanmaken door een video te maken van een paar minuten. We hebben voor deze methode gekozen omdat er bij de verificatie van gezichten een grote overlap tussen die kansdichtheden van verschillende gezichten is, waarbij deze methode het beste werkt. In Hoofdstuk 3, hebben we de effecten van het reduceren van het aantal features waarmee gezichten worden gemodelleerd onderzocht en we hebben geobserveerd dat we beter gebruik kunnen maken van alle features van een gereduceerde afbeelding dan dat we een feature reducerende functie toepassen voor grotere afbeeldingen. Daarnaast hebben we een vergelijking getrokken tussen een Gaussian model en een Gaussian mixture model, waarvan de eerst een betere prestatie leverde, minder complex was en beter kon generaliseren. Daarnaast blijk dat de likelihood ratio als verificatie afstand kan worden teruggebracht naar het verschil tussen twee Mahalanobis afstanden. De berekening hiervan betref alleen maar lineare matrix berekeningen en de complexiteit daarvan is laag

Om een zo goed mogelijk resultaat te halen, combineren we de informatie in het tijddomein. Met andere woorden: de beslissingen die gemaakt zijn per beeld dat een gezicht bevat worden gecombineerd. Om dit te doen hebben we in Hoofdstuk 5 en 6, het combineren van informatie bestudeerd en stellen we daartoe 3 methoden: 'threshold-optimized decision-level fusion', 'LLR-based score-level fusion' en 'hybrid fusion' voor. Decision-level fusion is bruikbaar in ons systeem, omdat we in theorie hebben bewezen dat voor het combineren van dezelfde beslissers er geen training nodig is. Daarnaast is de voorgestelde 'OR rule fusion' goed bruikbaar in scenario's waarin veel grote fouten voorkomen. Dit maakt ons systeem robuuster tegen variaties van expressies op gezichten en gezichten die onder een hoek naar de camera kijken, wat de gebruikersvriendelijkheid van ons systeem ten goede komt. Alle fusion methoden zijn ook succesvol gebruikt in het 3D-Face project, waarin twee dimensionale gezichtsafbeelding met de drie dimensionale vorm van het gezicht werden gecombineerd.

Acknowledgements

Four years ago, I was invited to the Netherlands for the Ph.D. interview. This trip impressed me very much with the nice Dutch people and the beautiful landscape of the Netherlands in April, and started a new page of my life. At the end of my Ph.D. work, I can still feel the happiness of the beginning.

I would like to express my sincere gratitude to my supervisor, Dr. Raymond Veldhuis, who had initialized the project, and supported my research work all through. Not only had I learned from him the inspiring ideas to solve specific scientific problems, but also gradually a more open and western way of thinking, both in work and in life.

I am grateful to my promoter, Prof. Dr. Kees Slump, for maintaining such cheerful and free atmosphere of our group, which any researcher would dream of. I am also grateful for the many trust and understanding he had shown on me in difficult times.

I would like to thank the colleagues who had worked closely with me on face recognition, Bas, Luuk, Gert, Andries, Robin, Ileana, Vidhan, for the interesting discussions and pleasant cooperations.

I want to thank our secretary, Anneke, for her endless kindness and help from the beginning to the end. (I will always remember the visit to the *gezellig* house of you and Gerrit.) Thank Geert Jan for his professional help with my computer and experimental setup. Thank Niels, Bas, Almar, Arno, for sharing with me the cozy office as well as many joyful moments. Special thanks to Bas for helping me with the Dutch translation in preparing the thesis.

My gratitude goes to all the members of the Signals and Systems Group, who had been together in this delightful and collaborative team. It also goes to the friends and neighbors on the picturesque campus of Twente, who had made my Ph.D. life truly colorful.

I would like to extend my thanks to my former supervisors at Fudan University, Shanghai, China: Prof. Weiqi Wang, Yuanyuan Wang, and Jianguo

Yu, who had initiated my interest in this research area, guided my first steps in Fudan, and kept a kind eye on me when I am here in Twente.

In the end, I would like to thank my family, father, mother, aunt, uncle and cousin, who had been standing by me all the time. Finally, I would like to thank my husband, Yi, for his infinite love and support, and would like to devote this thesis to the family of us.

Bibliography

[1] 3D Face. 3D Face biometric research. http://www.3dface.org/, 2006.

[2] T. Ahonen, M. Pietikainen, A. Hadid, and T. Maenpaa. Face recognition based on the appearance of local regions. In *International Conference on Pattern Recognition*, 2004.

[3] S. Arca, P. Campadelli, and R. Lanzarotti. A face recognition system based on local feature analysis. In *Audio- and Video-Based Biometric Person Authentication*, pages 182–189, 2003.

[4] M. Bartlett, H. Lades, and T. Sejnowski. Independent component representations for face recognition. In *the SPIE Symposium on Electonic Imaging: Science and Technology; Human Vision and Electronic Imaging*, San Jose, CA, 1998.

[5] R. Basri and D. Jacobs. Lambertian reflectances and linear subspaces. In *IEEE International Conference on Computer Vision*, 2001.

[6] A. Bazen and R. Veldhuis. Likelihood-ratio-based biometric verification. *IEEE transactions on circuits and systems for video technology*, 14(1):86–94, 2004.

[7] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[8] P. Belhumeur and D. Kriegman. What is the set of images of an object under all possible illumination conditions. *International Journal of Computer Vision*, 28(3):1–16, 1998.

[9] G. Beumer, A. Bazen, and R. Veldhuis. On the accuracy of EERs in face recognition and the importance of reliable registration. In *SPS IEEE Benelux DSP Valley*, 2005.

[10] G. Beumer, Q. Tao, A. Bazen, and R. Veldhuis. A landmark paper in face recognition. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 73–78, 2006.

[11] G. Beumer, Q. Tao, A. Bazen, and R.N.J. Veldhuis. Comparing landmarking methods for face recognition. In *16th Annual Workshop on Circuits Systems and Signal Processing*, Veldhoven, The Netherlands, 2005.

[12] C. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[13] R. Bolle, J. Connell, and N. Ratha. Biometric perils and patches. *Pattern Recognition*, 35(12):2727–2738, 2002.

[14] B. Boom, G. Beumer, L. Spreeuwers, and R. Veldhuis. The effect of image resolution on the performance of a face recognition system. In *Proceedings of the Ninth International Conference on Control, Automation, Robotics and Vision*, pages 409–414, Singapore, 2006.

[15] B. Boom, G. Beumer, L. Spreeuwers, and R. Veldhuis. Matching score based registration. In *PRORISC*, 2006.

[16] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *5th Annual Workshop on Computational Learning Theory*, Pittsburgh, USA, 1992.

[17] W. Boukabou, L. Ghouti, and A. Bouridane. Face recognition using a gabor filter bank approach. 2006.

[18] W. Boukabou, L. Ghouti, and A. Bouridane. Face recognition using a gabor filter bank approach. In *AHS '06: Proceedings of the first NASA/ESA conference on Adaptive Hardware and Systems*, pages 465–468, Washington, DC, USA, 2006. IEEE Computer Society.

[19] V. Bruce and M. Voi. Recognizing faces. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 302:423–436, 1983.

[20] M. Burl, T. Leung, and P. Perona. Face localization via shape statistics. In *International Workshop on Automatic Face and Gesture Recognition*, 1995.

[21] Rafael C. and Richard E. *Digital Image Processing.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[22] J. Cai and A. Goshtasby. Detecting human faces in color images. *Image Vision Comput.*, 18(1):63–75, 1999.

[23] T. Chan, J. Shen, and L. Vese. Variational pde models in image processing. *Notices of the American Mathematical Society*, 50(1):14–26, 2003.

[24] R. Chellappa, C.L. Wilson, and S. Sirohey. Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, 83(5):705–740, 1995.

[25] H. Chen, P. Belhumeur, and D. Jacobs. In search of illumination invariants. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 254–261, 2000.

[26] T. Cootes, G. Edwards, and J. Taylor. Active appreance models. In *European Conference on Computer Vision*, 1998.

[27] T. Cootes and J. Taylor. Active shape models - smart snakes. In *British Machine Vision Conference*, 1992.

[28] T. Cover. The best two independent measurements are not the two best. *IEEE Trans. Systems, Man, and Cybernetics*, 4:116–117, 1974.

[29] I. Craw, H. Ellis, and J. Lishman. Automatic extraction of face features. *Pattern Recognition Letters*, 5:183–187, 1987.

[30] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, 2000.

[31] D. Cristinacce and T. Cootes. Facial feature detection using adaboost with shape constraints. In *14th British Machine Vision Conference*, pages 231–240, 2003.

[32] D. Cristinacce, T. Cootes, and I. Scott. A multi-stage approach to facial feature detection. In *15th British Machine Vision Conference*, pages 277–286, 2004.

[33] D. Dai and P. Yuen. Regularized discriminant analysis and its application to face recognition. *Pattern Recognition*, 36(3):845–847, 2003.

[34] Y. Dai and Y. Nakano. Face-texture model-based on SGLD and its application in face detection in a color scene. *Pattern Recognition*, 29(6):1007–1017, 1996.

[35] S. Dass, K. Nandakumar, and A. Jain. A principled approach to score level fusion in multimodal biometric systems. In *Audio- and Video-Based Biometric Person Authentication*, pages 1049–1058, 2005.

[36] J. Daugman. Two-dimensional spectral analysis of cortical receptive field profile. *Vision Research*, 20:847–856, 1980.

[37] J. Daugman. Uncertain relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Optical Society American*, 2(7):1160–1169, 1985.

[38] J. Daugman. Complete discrete 2-d gabor transform by neural networks for image analysis and compression. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 36(7):1169–1179, 1988.

[39] J. Daugman. Combining multiple biometrics. `http://www.cl.cam.ac.uk/users/jgd1000/combine/combine.html`, 2000.

[40] J. Daugman. The importance of being random: Statistical principles of iris recognition. *Pattern Recognition*, 36(2):279–291, 2003.

[41] F. den Hartog, M. Blom, C Lageweg, M. Peeters, J. Schmidt, R. van der Veer, A. de Vries, M.R. van der Werff, Q. Tao, R. Veldhuis, N. Baken, and F. Selgert. First experiences with personal networks as an enabling platform for service providers. In *Second International Workshop on Personalized Networks*, Philadelphia, USA, 2007.

[42] O. Deniz, M. Castrillon, and M. Hernandez. Face recognition using independent component analysis and support vector machines. In *Audio- and Video-Based Biometric Person Authentication*, pages 59–64, 2001.

[43] P. Domingos and M. Pazzani. Beyond independence: Conditions for the optimality of the simple Bayesian classifier. In *13th Internat. Conf. on Machine Learning*, 1996.

[44] R. Duda, P. Hart, and D. Stork. *Pattern Classification (2nd ed.).* John Wiley and Sons, New York, 2001.

[45] M. Escobar and J. Ruiz del solar. Biologically-based face recognition using gabor filters and log-polar images. 2002.

[46] K. Etemad and R. Chellappa. Discriminant analysis for recognition of human face images. In *AVBPA '97: Proceedings of the First International Conference on Audio- and Video-Based Biometric Person Authentication*, pages 127–142, London, UK, 1997. Springer-Verlag.

[47] K. Etemad and R. Chellappa. Discriminant analysis for recognition of human face images. *Journal of the Optical Society of America A*, 14:1724–1733, 1997.

[48] M. Faundez-Zanuy. Data fusion in biometrics. *IEEE Aerospace and Electronic Systems Magazine*, 20(1):34–38, 2005.

[49] T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[50] R. Feris, J. Gemmell, K. Toyama, and V. Krueger. Hierarchical wavelet networks for facial feature localization. In *ICCV'01 Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*, 2001.

[51] M. Figueiredo and A. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.

[52] R. Fisher. The use of multiple measurements in taxonomic problems. *Annals Eugen.*, 7:179–188, 1936.

[53] Freeband. PNP2008: Development of a user centric ambient communication environment. `http://www.freeband.nl/project.cfm?language=en&id=530`.

[54] B. Fröba and C. Küblbeck. Real-time face detection using edge-orientation matching. In *Third International Conference on Audio- and Video-Based Biometric Person Authentication*, pages 78–83, London, UK, 2001.

[55] K. Fukunaga. *Introduction to Statistical Pattern Recognition, Second Edition.* Academic Press, 1990.

[56] A. Georghiades, P. 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.

[57] B. Gokberk and L. Akarun. Comparative analysis of decision-level fusion algorithms for 3d face recognition. In *the 18th International Conference on Pattern Recognition*, pages 1018–1021, Washington, DC, USA, 2006.

[58] V. Govindaraju. Locating human faces in photographs. *Int'l J. Computer Vision*, 19(2):129–146, 1996.

[59] G. Guo, S. Li, and K. Chan. Face recognition by support vector machines. *IEEE International Conference on Automatic Face and Gesture Recognition*, 00, 2000.

[60] T. Hastie and R. Tibshirani. Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society (B)*, 58:155–176, 1996.

[61] B. Heisele, P. Ho, and T. Poggio. Face recognition with support vector machines: Global versus component-based approach. In *International Conference on Computer Vision*, pages 688–694, 2001.

[62] R. Herpers, M. Michaelis, K. Lichtenauer, and G. Sommer. Edge and keypoint detection in facial regions. In *2nd International Conference on Automatic Face and Gesture Recognition*, pages 212–217, 1996.

[63] G. Heusch, F. Cardinaux, and S. Marcel. Lighting normalization algorithms for face verification. Technical Report 03, IDIAP, 2005.

[64] G. Heusch, Y. Rodriguez, and S. Marcel. Local binary patterns as image preprocessing for face authentication. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2006.

[65] E. Hielmas and B. Low. Face detection: a survey. *Computer Vision and Image Understanding*, 83:235–274, 2001.

[66] B. Horn. Shape from shading: a method for obtaining the shape of a smooth opaque object from on view. Technical report, AITR, MIT, 1970.

[67] R. Hsu, M. Abdel Mottaleb, and A. Jain. Face detection in color images. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 24(5):696–706, 2002.

[68] R. Hsu, M. Abdel-Mottaleb, and A. Jain. Face detection in color images. 24(5), 2002.

[69] M. Huesken, M. Brauckmann, S. Gehlen, K. Okada, and C. von der Malsburg. Evaluation of implicit 3D modeling for pose-invariant face recognition. In *Biometric Technology for Human Identification.*, volume 5404, pages 328–338, 2004.

[70] M. Huesken, M. Brauckmann, S. Gehlen, and C. von der Malsburg. Strategies and benefits of fusion of 2d and 3d face recognition. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, pages 174–180, Washington, DC, USA, 2005.

[71] M. Hunke and A. Waibel. Face locating and tracking for human-computer interaction. In *28th Asolimor Conference on Signals, Systems, and Images*, Monterey, USA, 1994.

[72] F. Iannarilli and P. Rubin. Feature selection for multiclass discrimination via mixed-integer linear programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):779–783, 2003.

[73] Intel. Open computer vision library. `http://sourceforge.net/projects/opencvlibrary/`.

[74] A. Jain, R. Bolle, and S. Pankanti. *Biometrics, Personal Identification in Networked Society.* Kluwer Academic Publishers, 1998.

[75] A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[76] A. Jain, K. Nandakumar, and A. Ross. Score normalization in multimodal biometric systems. *Pattern Recognition*, 38(12):2270–2285, 2005.

[77] T. Joachims. *Making large-scale support vector machine learning practical.* MIT Press, Cambridge, MA, USA, 1999.

[78] D. Jobson, Z. Rahmann, and G. Woodell. A multiscale retinex for bridging the gap between color images and the human observations of scenes. *IEEE Transactions on Image Processing*, 6(7), 1997.

[79] D. Jobson, Z. Rahmann, and G. Woodell. Properties and performance of a center/surround retinex. *IEEE Transactions on Image Processing*, 6(3):451–462, 1997.

[80] K. Jonsson, J. Matas, J. Kittler, and Y. Li. Learning support vectors for face verification and recognition. In *the IEEE Computer Society Conference on Automatic Face and Gesture Recognition*, pages 208–213, 2000.

[81] P. Juell and R. Marsh. A hierarchical neural-network for human face detection. *Pattern Recognition*, 29(5):781–787, 1996.

[82] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *ACM Conference on Computer and Communications Security*, pages 28–36, 1999.

[83] T. Kanade. Computer recognition of human faces. *Interdisciplinary Systems Research*, 47, 1977.

[84] S. Kay. *Fundamentals of Statistical Signal Processing: Detection Theory.* Prentice-Hall, Inc., 1998.

[85] T. Kim, H. Kim, W. Hwang, S. Kee, and J. Kittler. Face description based on decomposition and combining of a facial space with lda. In *International Conference on Image Processing*, pages Vol III: 877–880, 2003.

[86] T. Kim, H. Kim, W. Hwang, and J. Kittler. Independent component analysis in a local facial residue space for face recognition. *Pattern Recogniton*, 37(9):1873–1885, 2004.

[87] T. Kim and J. Kittler. Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):318–327, 2005.

[88] S. King, G.Y. Tian, D. Taylor, and S. Ward. Cross-channel histogram equalisation for colour face recognition. In *AVBPA03*, pages 454–461, 2003.

[89] J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[90] R. Kohavi and G. John. Wrapper for feature subset selection. *Artificial Intelligence*, 97(1):273–324, 1997.

[91] L. Kuncheva, J. Bezdek, and R. Duin. Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, 32(2):299–314, 2001.

[92] L. Kuncheva, C. Whitaker, A. Shipp, and R. Duin. Is independence good for combining classifiers? In *15th International Conference on Pattern Recognition*, pages 168–171, 2000.

[93] S. Kung and J. Taur. Decision-based neural networks with signal/image classification applications. *IEEE Transactions on Neural Networks*, 6(1):170–181, 1995.

[94] E. Land and J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 61(1):1–11, 1971.

[95] C. Lee, J. Kim, and K. Park. Automatic human face location in a complex background using motion and color information. *Pattern Recognition*, 29(11):1877–1889, 1996.

[96] S. Li and A. Jain. *Handbook of Face Recognition*. Springer-Verlag, 2004.

[97] S. Lin, S. Kung, and L. Lin. Face recognition/detection by probabilistic decision-based neural network. *IEEE Transactions on Neural Networks*, 8(1):114–132, 1997.

[98] J. Linnartz and P. Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In *4th Conference on Audio and Video-base Biometric Person Verification*, Guildford, UK, 2003.

[99] N. Littlestone and M. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[100] C. Liu and H. Wechsler. Evolutionary pursuit and its application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):570–582, 2000.

[101] A. Lumini and L. Nanni. Combining classifiers to obtain a reliable method for face recognition. *Multimedia Cyberscape Journal*, 3(3):47–53, 2005.

[102] G. Marcialis and F. Roli. Fusion of appearance-based face recognition algorithms. *Pattern Analysis and Applications*, 7(2):151–163, 2004.

[103] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET curve in assessment of detection task performance. In *Proc. Eurospeech '97*, pages 1895–1898, Rhodes, Greece, 1997.

[104] S. McKenna, S. Gong, and J. Collins. Face tracking and pose representation. In *British Machine Vision Conference*, volume 2, pages 755–764, Edinburgh, UK, 1996.

[105] G. McLachlan and K. Basford. *Mixture Models: Inference and Application to Clustering.* Marcel Dekker, New York, 1988.

[106] G. McLachlan and K. Basford. *The EM algorithm and Extensions.* John Wiley and Sons, New York, 1997.

[107] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. XM2VTsbd: The extended m2vts database. In *2nd Conference on Audio and Video-base Biometric Person Verification*, 1999.

[108] B. Moghaddam, T. Jebara, and A. Pentland. Bayesian face recognition. *Pattern Recognition*, 33:1771–1782, 2000.

[109] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *International Conference on Computer Vision*, pages 786–793, Cambridge, USA, June 1995.

[110] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):696–710, 1997.

[111] Y. Moses, Y. Adini, and S. Ullman. Face recognition: The problem of compensating for changes in illumination direction. In *In European Conference on Computer Vision*, 1994.

[112] K. Nandakumar, Y. Chen, S. Dass, and A. Jain. Likelihood ratio-based biometric score fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):342–347, 2008.

[113] K. Nandakumar, A. Jain, and S. Pankanti. Fingerprint-based fuzzy vault: Implementation and performance. *IEEE Transactions on Information Forensics and Security*, 2(12):744–757, 2007.

[114] J. Nocedal and S. Wright. *Numerical Optimization*. Springer-Verlag, 2006.

[115] Biometric System Laboratory University of Bologna. FVC2006: The fourth international fingerprint verification competition. `http://bias.csr.unibo.it/fvc2006/default.asp`.

[116] L. O'Gorman. Overview of fingerprint technologies. *Elsevier Information Security Technical Report*, 3(1), 1998.

[117] 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, 2004.

[118] N. Oliver, A. Pentland, and F. Berard. LAFTER: Lips and face real time tracker with facial expression recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[119] A. Ortega and K. Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine*, 15(6):23–50, 1998.

[120] E. Osuna, R. Freund, and F. Girosi. Training support vector machines:an application to face detection. In *the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997.

[121] H. Peng, F. Long, and C. Ding. Feaure selction based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[122] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, June 1994.

[123] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[124] P. Phillips, P. Flynn., T. Scruggs, K.W. Bowyer, J. Chang, K K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *Computer Vision and Pattern Recognition*, pages 947– 954, 2005.

[125] P. Phillips, H. Moon, S. Rizvi, and P. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[126] P. Phillips, W. Scruggs, A. O'Toole, P. Flynn, K. Bowyer, C. Schott, and M. Sharpe. FRVT 2006 and ICE 2006 large-scale results. *Technical Report NISTIR 7408, NIST*, 3(1), 2007.

[127] S. Phung, A. Bouzerdoum, and D. Chai. Skin segmentation using color pixel classification: Analysis and comparison. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 27(1):148–154, 2005.

[128] N. Poh and S. Bengio. Database, protocols and tools for evaluating score-level fusion algorithms in biometric authentication. *Pattern Recognition*, 39(2):223–233, 2006.

[129] S. Prabhakar and A. Jain. Decision-level fusion in fingerprint verification. *pattern Recognition*, 35:861–874, 2002.

[130] M. Przybocki and A. Martin. NIST speaker recognition evaluation chronicles. In *The Speaker and Language Recognition Workshop*, pages 12–22, Toledo, Spain, 2004.

[131] T. Riopka and T. Boult. The eyes have it. In *ACM SIGMM Multimedia Biometrics Methods and Applications Workshop*, pages 9–16, 2003.

[132] A. Ross and A. Jain. Information fusion in biometrics. *Pattern Recognition Letters*, 24(13), 2003.

[133] A. Ross, K. Nandakumar, and A. Jain. *Handbook of Multibiometrics (International Series on Biometrics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[134] D. Roth, M. Yang, and N. Ahuja. A SNoW-based face detector. *Advances in Neural Information Processing Systems*, 2000.

[135] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[136] D. Ruderman, T. Cronin, and C. Chiao. Statistics of cone responses to natural images: Implications for visual coding. 15(8):2036–2046, 1998.

[137] E. Saber and A. Tekalp. Frontal-view face detection and facial feature extraction using color, shape and symmetry based cost functions. 19(8), 1998.

[138] T. Sakai, M. Nagao, and S. Fujibayashi. Line extraction and pattern detection in a photograph. *Pattern Recognition*, 1:233–248, 1969.

[139] F. Samaria. *Face recognition using hidden Markov models.* PhD thesis, Univerisity of Cambridge, UK, 1994.

[140] F. Samaria and S. Young. HMM-based architecture for face identification. *Interdisciplinary Systems Research*, 12(8), 1994.

[141] C. Sanderson and K. Paliwal. Information fusion and person verification using speech and face information. Technical report, IDIAP, Switzerland, September 2002.

[142] A. Sao and B. Yegnanarayana. Face verification using correlation filters and autoassociative neural networks. *International Conference on Intelligent Sensing and Information Processing*, pages 364–367, 2004.

[143] R. Schapire, Y. Freund, P. Bartlett, and W. Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. In *14th International Conference on Machine Learning*, pages 322–330, 1997.

[144] B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, pages 1299–1319, 1998.

[145] S. Shan, P. Yang, X. Chen, , and W. Gao. AdaBoost Gabor Fisher classifier for face recognition. In *AMFG'2005*, pages 279–292, 2005.

[146] A. Shashua. *Geometry and Photometry in 3D Visual Recognition.* PhD thesis, Massachusetts Insitute of Technology, USA, 1997.

[147] 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(2):129–139, 2001.

[148] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[149] T. Sim and T. Kanade. Illuminating the face. Technical report, Robotics Institute, Carnegie Mellon University, September 2001.

[150] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4:519–524, 1987.

[151] R. Smith, J. Kittler, M. Hamouz, and J. Illingworth. Face recognition using angular lda and svm ensembles. In *International Conference on Pattern Recognition*, pages Vol III: 1008–1012, 2006.

[152] K. Sobottka and I. Pitas. Extraction of facial regions and features using color and shape information. In *Int. Conf. on Pattern Recognition*, Vienna, Austria.

[153] J. Sochman and J. Matas. AdaBoost with totally corrective updates for fast face detection. In *the IEEE Computer Society Conference on Automatic Face and Gesture Recognition*, pages 445–450, 2004.

[154] Q. Song and J. Robinson. A feature space for face image processing. *the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2, 2000.

[155] Y. Song, Y. Kim, U. Chang, and H. Kwon. Face recognition robust to left/right shadows; facial symmetry. *Pattern Recognition*, 39(8):1542–1545, August 2006.

[156] Q. Tao, R. van Rootseler, R. Veldhuis, S. Gehlen, and F. Weber. Optimal decision fusion and its application on 3d face recognition. In *Special Interest Group on Biometrics and Electronic Signatures*, pages 15–24, Darmstadt, Germany, 2007.

[157] Q. Tao and R. Veldhuis. Biometric authentication for mobile personal device. In *First International Workshop on Personalized Networks*, San Jose, USA, 2006.

[158] Q. Tao and R. Veldhuis. Verifying a user in a personal face space. In *9th International Conference on Control, Automation, Robotics, and Vision*, pages 197–200, Singapore, 2006.

[159] Q. Tao and R. Veldhuis. Illumination normalization based on simplified local binary patterns for a face verification system. In *IEEE Biometrics Consortium Conference*, pages 1–7, Baltimore, USA, 2007.

[160] Q. Tao and R. Veldhuis. Optimal decision fusion for a face verification system. In *the 2nd International Conference on Biometrics*, pages 958–967, Seoul, Korea, 2007.

[161] Q. Tao and R. Veldhuis. Optimal decision fusion for verification of face sequences. In *the 28th Symposium on Information Theory in the Benelux*, pages 297–303, Enschede, the Netherlands, 2007.

[162] Q. Tao and R. Veldhuis. Hybrid fusion for biometrics: Combining score-level and decision-level fusion. In *Workshop on Biometrics, IEEE Conferene on Computer Vision and Pattern Recognition*, Alaska, US, 2008.

[163] Q. Tao and R. Veldhuis. Optimal likelihood-ratio based biometric score fusion. In *the 29th Symposium on Information Theory in the Benelux*, pages 11–18, Leuven, Belgium, 2008.

[164] Q. Tao and R. Veldhuis. A study on illumination normalization for 2D face verification. In *International Conference on Computer Vision Theory and Applications*, pages 42–49, Madeira, Portugal, 2008.

[165] Q. Tao and R. Veldhuis. Threshold-optimized decision-level fusion and its application to biometrics. *Pattern Recognition*, in press.

[166] D. Tax. One class classification. *Ph.D. thesis, Delft University of Technology*, 2001.

[167] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, 1991.

[168] P. Tuyls, A. Akkermans, T. Kevenaar, G. Schrijen, A. Bazen, R, and Veldhuis. Practical biometric authentication with template protection. In *5th International Conference on Audio- and Video-Based Perosnal Authentication*, pages 436–446, Rye Brook, NY, USA, 2005.

[169] N. Ueda and R. Nakano. Deterministic annealing EM algorithm. *Neural Networks*, 11:271–282, 1998.

[170] B. Ulery, A. Hicklin, C. Watson, W. Fellner, and P. Hallinan. Studies of biometric fusion. *NIST Techical Report IR 7346*, 2006.

[171] BioID. Bioid face database. `http://www.humanscan.de/`.

[172] FERET. Feret face database. `http://www.itl.nist.gov/iad/humanid/feret/`.

[173] FRGC. FRGC face database. `http://face.nist.gov/frgc/`.

[174] H. Van Trees. *Detection, Estimation, and Modulation Theory.* John Wiley and Sons, New York, 1969.

[175] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer Verlag, New York, 1995.

[176] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *International Journal on Computer Vision*, 62(1-2):61–81, 2005.

[177] R. Veldhuis, A. Bazen, W. Booij, and A. Hendrikse. A comparison of hand-geometry recognition methods based on low- and high-level features. In *15th Annual Workshop on Circuits, Systems and Signal Processing*, pages 326–330, Veldhoven, The Netherlands, 2004.

[178] R. Veldhuis, F. Deravi, and Q. Tao. Multibiometrics for face recognition. *Datenschutz und Datensicherheit*, 32(3):204–214, 2008.

[179] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[180] P. Viola and W. Wells. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2), 1997.

[181] H. Wang and S. Chang. A highly efficient system for automatic face detection in MPEG video. *IEEE Trans. Circuit and Systems for Video Technology*, 21(4):557–563, 1997.

[182] Y. Wang, T. Tan, and Anil K. Jain. Combining face and iris biometrics for identity verification. In *Fourth International Conference on AVBPA*, pages 805–813, 2003.

[183] F. Weber and A. Hernández. Face location by template matching with a quadratic discriminant function. In *International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, page 10, Washington, DC, USA, 1999.

[184] L. Wiskott, J. Fellous, N. Krüger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), 1997.

[185] M. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1), 2002.

[186] H. Yoshikawa and S. Ikebata. A microcomputer based personal identifictaion system. In *International Conference on Industrial Electronics*, volume 1, pages 105–109, 1984.

[187] H. Zhang. The optimality of Naive Bayes. In *17th Internat. FLAIRS Conf.*, 2004.

[188] W. Zhang, Y. Chang, and T. Chen. Optimal thresholding for key generation based on biometrics. In *International Conference on Image Processing*, 2004.

[189] W. Zhao. Performance perturbation analysis of eigen-systems. In *International Conference on Pattern Recognition*, pages Vol II: 105–108, 2000.

[190] W. Zhao, R. Chellappa, and A. Krishnaswamy. Discriminant analysis of principal components for face recognition. In *International Conference on Automatic Face and Gesture Recognition*, pages 336–341, 1998.

[191] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Comput. Surv.*, 35(4):399–458, 2003.

[192] W. Zhao, P. Phillips, R. Chellappa, and N. Nandhakumar. Linear discriminant analysis of mpf for face recognition. In *International Conference on Pattern Recognition*, pages Vol I: 185–188, 1998.

[193] S. Zhou and R. Chellapa. Illumination light field: image-based face recognition across illumination and pose. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2004.

[194] F. Zuo. *Embedded Face Recognition Using Cascaded Structures.* PhD thesis, Technische Universiteit Eindhoven, NL, 2006.