Automatic Person Verification Using Speech and Face Information

---

A dissertation presented to
the School of Microelectronic Engineering
Faculty of Engineering and Information Technology
Griffith University

Submitted in fulfillment of the requirements of
the degree of Doctor of Philosophy

by

Conrad Sanderson, BEng (Hons)

2003
(revised in 2007)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Identity verification systems are part of our every day life – one example is the Automatic Teller Machine (ATM) which employs a simple identity verification scheme: the user is asked to enter their[1] password after inserting their card. If the password matches the one prescribed to the card, the user is allowed access to their bank account. This scheme suffers from a drawback: only the validity of the combination of a certain possession (the ATM card) and certain knowledge (the password) is verified. The ATM card can be lost or stolen, and the password can be compromised. New verification methods have hence emerged, where biometrics such as the person's speech, face image or fingerprints can be used in addition to the password. Such biometric attributes cannot be lost and typically vary considerably from person to person.

Apart from the ATM example described above, biometrics can be applied to other areas, such as telephone & internet based banking, passport control (immigration checkpoints), as well as forensic work (to determine whether a biometric sample belongs to a suspect) and law enforcement applications (e.g. surveillance) [11, 35, 44, 67, 98, 105, 112, 128, 190].

While biometric systems based on face images and/or speech signals can be effective [111, 128, 149], their performance can degrade in the presence of challenging conditions. For speech based systems this is usually in the form of channel distortion and/or ambient noise. For face based systems it can be in the form of a change in the illumination direction and/or a change in the pose of the face [3, 159].

Multi-modal systems use more than one biometric at the same time. This is done for two main reasons: (i) to achieve better robustness (where the impact of a biometric affected by environmental conditions can be decreased) and (ii) to increase discrimination power (as complementary information can be used). Multi-modal systems are often comprised of several modality experts and a decision stage [22].

This work overviews relevant backgrounds and reports research aimed at increasing the robustness of single- and multi-modal biometric verification systems, in particular those based on speech and face modalities.

[1] The word 'their' is used as gender neutral replacement of the words 'his' and 'her' [125].

1.1 Acknowledgements

This work is in large part a result of my studies under Professor Kuldip K. Paliwal (Griffith University, Queensland, Australia). Other parts of this work have been created at the IDIAP Research Institute (Valais, Switzerland) with valuable input from Dr. Samy Bengio.

The ideas and work discipline of both Prof. Paliwal and Dr. Bengio have been inspirational. I am also indebted to my family for their support and understanding. My thanks further go to friends, colleagues and visiting researchers for their suggestions and many interesting discussions. Lastly, but not in the least, my thanks go to Dr Robert Davies (for his C++ matrix library) and to the numerous developers of the Linux kernel & GNU tools.

1.2 Overview

This work is comprised of three major parts: (i) verification using speech signals (Chapter 3), (ii) verification using face images (Chapters 4 and 5), (iii) verification using fused speech and face information (Chapter 6). It is supported by Chapter 2, which provides an overview of relevant pattern recognition theory. To ease reading, each chapter is largely self contained. The chapters are summarised as follows:

- Chapter 2 – Statistical Pattern Recognition – first draws distinctions between closed set identification, open set identification and verification. Relevant pattern recognition theory is then used to derive a two-class decision machine (classifier) used in the verification system. The machine is implemented using the Gaussian Mixture Model (GMM) approach. The k-means, Expectation Maximisation (EM) and maximum a-posteriori (MAP) adaptation algorithms, which are used for finding GMM parameters, are described. Two methods for finding the impostor likelihood are presented: the Background Model Set (BMS) and Universal Background Model (UBM). Next, error measures for finding the performance of a verification system, such as the Equal Error Rate (EER), are described. The chapter is concluded by a discussion on implementation issues, where practical limitations and experimental requirements are taken into account. The implementation of the decision machine is tested in the following chapter.

- Chapter 3 – Verification using Speech Signals – first describes the difference between text-dependent and text-independent systems. A review of the human speech production process is then given, followed by a review of feature extraction approaches used in a speaker verification system. Mel Frequency Cepstral Coefficients (MFCCs), delta features and Cepstral Mean Subtraction (CMS) are covered. An alternative feature set, termed Maximum Auto-Correlation Values (MACVs), which uses information from the source part of the speech signal, is also covered. A parametric Voice Activity Detector (VAD), used for disregarding silence and noise segments of the speech signal, is briefly described. The implementation of

the Gaussian Mixture Model classifier (described in Chapter 2) is tested. The use of MACVs is evaluated for reducing the performance degradation of a verification system used in noisy conditions.

- Chapter 4 – Verification using Frontal Face Images – first overviews important publications in the field of frontal face recognition. Geometric features, templates, Principal Component Analysis (PCA), pseudo-2D Hidden Markov Models (HMM), Elastic Graph Matching (EGM), as well as other points are covered. Relevant issues, such as the effects of an illumination direction change and the use of different face areas, are also covered. Several new feature extraction approaches are proposed – their robustness and performance is evaluated against three popular methods (PCA, 2D DCT and 2D Gabor wavelets) for use in an identity verification system subject to illumination direction changes. It is also shown that when using the GMM classifier with local features (such as Gabor wavelets or DCT derived features), the spatial relationships between face parts (e.g. eyes and nose) are disregarded. Such a face recognition system can surprisingly still provide good performance and provides advantages such as robustness to translations. The fragility of PCA derived features to illumination changes is addressed by introducing a pre-processing step which involves applying local feature extraction to the original face image – it is shown that the enhanced PCA technique is robust to illumination changes as well as retaining the positive aspects of traditional PCA, i.e. robustness to compression artefacts and noisy images, which might be important in forensic and law enforcement applications.

- Chapter 5 – Verification using Faces with Pose Variations – deals with faces subject to pose variations, in contrast to the previous chapter which dealt with frontal faces. A framework is developed for addressing the pose mismatch problem that occurs when there is only a single (frontal) face image available for training and non-frontal faces are presented during testing. In particular, the mismatch problem is tackled through building multi-angle models by extending each frontal face model with artificially synthesised models for non-frontal views. The synthesis methods are based on several implementations of Maximum Likelihood Linear Regression (MLLR), as well as standard multi-variate linear regression (LinReg). We stress that instead of synthesising images, model parameters are synthesised. The synthesis and extension approach is evaluated by applying it to two face verification systems: a holistic system (based on PCA-derived features) and a local feature system (based on DCT-derived features). It is also shown that the local feature system is less affected by view changes than the holistic system.

- Chapter 6 – Verification Using Fused Speech and Face Information – first provides an overview of key concepts in the information fusion area, followed by a review of important milestones in audio-visual person identification and verification. Several adaptive and non-adaptive techniques for reaching the verification decision, based on combined speech and

face information, are then evaluated in clean and noisy audio conditions on a common dataset. It is shown that in clean conditions most of the non-adaptive approaches provide similar performance and in noisy conditions most exhibit a severe deterioration in performance. It is also shown that current adaptive approaches are either inadequate or use restrictive assumptions. A new category of classifiers is then introduced, where the decision boundary is fixed but constructed to take into account how the distributions of opinions are likely to change due to noisy conditions. Compared to a previously proposed adaptive approach, the proposed classifiers do not make a direct assumption about the type of noise that causes the mismatch between training and testing conditions.

1.3 Composite Literature Review

Since this work covers several distinct yet related topics, each topic has its own literature review. The overall literature review is comprised of:

- The whole of Chapter 2, which covers relevant pattern recognition theory necessary to build a decision machine (classifier), based on Gaussian Mixture Models, for a verification system.
- Sections 3.3 and 3.4, which respectively cover the speech production process and methods for feature extraction from speech signals.
- Section 4.2, which covers important face recognition approaches (and surrounding issues) for dealing with frontal faces. Sections 4.3.1 to 4.3.3 cover several feature extraction techniques.
- Sections 5.2 and 5.3 which deal with face recognition using non-frontal faces (i.e. faces subject to pose variations).
- Sections 6.2 and 6.3, which respectively provide an overview of key concepts in the information fusion area and milestones in the field of audio-visual person recognition.

1.4 Related Topics

Apart from speech signals and face images, it is also possible to use biometrics such as the iris, fingerprints and hand geometry [41, 72, 156]. Other important aspects in biometrics include hiding biometric data as well as privacy and security issues [25, 77, 190]. Further introductory and review material about the biometrics field in general can be found in [47, 128, 185].

1.5 Acronyms and Abbreviations

This work uses the following acronyms and abbreviations:

BMS	Background Model Set
CMS	Cepstral Mean Subtraction
dB	decibel
DCF	Decision Cost Function
DCT	Discrete Cosine Transform
EER	Equal Error Rate
EGM	Elastic Graph Matching
EM	Expectation Maximisation
EPCA	Enhanced Principal Component Analysis
ERM	Empirical Risk Minimisation
FA	False Acceptance
FAR	False Acceptance Rate
FFT	Fast Fourier Transform
fps	frames per second
FR	False Rejection
FRR	False Rejection Rate
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
HTER	Half Total Error Rate
MACVs	Maximum Auto-Correlation Values
MAP	Maximum a-posteriori
MFCCs	Mel-Frequency Cepstral Coefficients
ML	Maximum Likelihood
MLP	Multi Layer Perceptron
MS-HMM	Multi-Stream Hidden Markov Model
PCA	Principal Component Analysis
PL	Piece-wise Linear
PSNR	Peak Signal-to-Noise Ratio
RBF	Radial Basis Function
SNR	Signal-to-Noise Ratio
SRM	Structural Risk Minimisation
SVM	Support Vector Machine
TE	Total Error
UBM	Universal Background Model
VAD	Voice Activity Detector

1.6 Notation

Throughout this work the following mathematical notation is mainly used:

$\boldsymbol{v}$	a column vector (lowercase, boldface)
$\boldsymbol{v}^T$	transpose of vector $\boldsymbol{v}$
$[\ v_1\ \ v_2\ \cdots\ v_D\ ] = [\ v_i\ ]_{i=1}^{D}$	contents of a D-dimensional row vector, where v_i is the i-th element
$\|\|\ \boldsymbol{v}\ \|\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_D^2}$	norm of vector $\boldsymbol{v}$, where v_i is the i-th element
$\{\ \boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_N\} = \{\ \boldsymbol{v}_i\ \}_{i=1}^{N}$	set of N vectors, where $\boldsymbol{v}_i$ is the i-th vector
$A \cup B$	union of sets A and B
$\boldsymbol{M}$	a matrix (uppercase, boldface)
$\boldsymbol{M}_{(i,j)}$	element of matrix $\boldsymbol{M}$, located at row i and column j
$\boldsymbol{M}^T$	transpose of matrix $\boldsymbol{M}$
$\boldsymbol{M}^{-1}$	inverse of matrix $\boldsymbol{M}$
$\|\boldsymbol{M}\|$	determinant of matrix $\boldsymbol{M}$
$\boldsymbol{\Sigma}$	a covariance matrix, when used in the context of a linear algebra expression (e.g., $\boldsymbol{v}^T\boldsymbol{\Sigma}\boldsymbol{v}$)
λ	a parameter set (e.g. parameters of a GMM)
$\lambda_A \sqcup \lambda_B$	concatenation of GMM parameter sets λ_A and λ_B
$\mathcal{N}(\boldsymbol{x}\|\boldsymbol{\mu}, \boldsymbol{\Sigma})$	a multi-variate Gaussian function with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$

Chapter 2

Statistical Pattern Recognition

2.1 Overview

In this chapter we first draw distinctions between closed set identification, open set identification and verification. Relevant pattern recognition theory is then used to derive a two-class decision machine (classifier) used in a verification system. The machine is implemented using the Gaussian Mixture Model (GMM) approach. The k-means, Expectation Maximisation (EM) and maximum a-posteriori (MAP) adaptation algorithms, used for finding GMM parameters, are described. Two methods for finding the impostor likelihood are presented: the Background Model Set (BMS) and Universal Background Model (UBM). Error measures for finding the performance of a verification system, such as the Equal Error Rate (EER), are then described. The chapter is concluded by a discussion on implementation issues, where practical limitations and experiment requirements are taken into account. The implementation of the decision machine is tested in the following chapter.

2.2 Recognition Types

The configuration of a biometric recognition system can be divided into three types: (i) the closed set identification task, (ii) the open set identification task, (iii) the verification task[1]. In closed set identification, the job is to assign a given sample into one of N classes (where N is the number of known persons). In open set identification, the task is to assign a given sample into one of $N+1$ classes, where the extra class represents an "unknown" or "previously unseen" person. In the verification task the classifier must assign a given sample into one of two classes: either the sample belongs to the person whose identity is being claimed (i.e. a true claimant), or it belongs to an impostor.

The verification and open set identification tasks represent operation in an uncontrolled environment [83], where any person could be encountered. In contrast, the closed set identification task assumes that all the persons to be encountered are already known. It has been suggested that while the closed set identification task has received considerable scientific interest, the verification

[1] verification is also known as authentication.

task has potentially more commercial applications [44, 49]. As such we concentrate on biometric verification systems in this work.

2.3 Bayesian Decision Theory

As mentioned above, a verification system is essentially a two-class decision machine: based on given observation vectors, the client is either an impostor or the true claimant. In this chapter we shall use Bayesian Decision Theory [15, 46, 151] to implement the decision machine.

Let us denote client specific true claimant and impostor classes as C_1 and C_2, respectively, and let $\boldsymbol{x} = [x_1\ x_2\ \cdots\ x_D]^T$ be the observation vector. Moreover, let $P(C_j)$ be the a-priori probability of class C_j, and $p(\boldsymbol{x}|C_j)$ be the conditional probability density function (pdf) of $\boldsymbol{x}$, given class C_j. We seek to find the class that $\boldsymbol{x}$ belongs to. Using the Bayes formula [15, 113], we obtain:

$$P(C_j|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_j)P(C_j)}{p(\boldsymbol{x})} \tag{2.1}$$

where

$$p(\boldsymbol{x}) = \sum_{i=1}^{2} p(\boldsymbol{x}|C_i)P(C_i) \tag{2.2}$$

Hence we obtain the a-posteriori probability of class C_j. It follows that the decision rule is then:

$$\text{choose } C_1 \text{ if } P(C_1|\boldsymbol{x}) > P(C_2|\boldsymbol{x}) \tag{2.3}$$

Or, more generally,

$$\text{index of chosen class} = \arg\max_j P(C_j|\boldsymbol{x}) \tag{2.4}$$

which is known as the maximum a-posteriori decision rule. Note that in our case $p(\boldsymbol{x})$ is not required for making the decision – the decision rule thus becomes:

$$\text{index of chosen class} = \arg\max_j p(\boldsymbol{x}|C_j)P(C_j) \tag{2.5}$$

Intuitively, the decision machine will make less mistakes when using more observations vectors. Thus in practice, multiple observation vectors are used: $X = \{\boldsymbol{x}_i\}_{i=1}^{N_V}$, where N_V is the number of vectors. Assuming that the observation vectors are independent and identically distributed[2] (i.i.d.), then the joint likelihood is:

$$p(X|C_j) = \prod_{i=1}^{N_V} p(\boldsymbol{x}_i|C_j)P(C_j) \tag{2.6}$$

[2]Due to the speech production process (see Chapter 3), feature vectors extracted from a speech signal are often correlated. However, the mathematics are greatly simplified if we assume that the observation vectors are independent.

In practice, $p(\boldsymbol{x}|C_j)$ is approximated through a parametric representation, $\widehat{p}(\boldsymbol{x}|C_j)$, which is estimated using training data. Since $\widehat{p}(\boldsymbol{x}|C_j)$ is an approximation[3], an associated "correction" function, $\varphi(\boldsymbol{x}|C_j)$, is theoretically required:

$$p(X|C_j) = \prod_{i=1}^{N_V} \widehat{p}(\boldsymbol{x}|C_j)\varphi(\boldsymbol{x}|C_j)P(C_j) \tag{2.7}$$

Taking into account the multiple observation vectors and rewriting (2.5) into a ratio test yields:

$$\text{chosen class} = \begin{cases} C_1 & \text{if} \quad \frac{\widehat{p}(X|C_1)}{\widehat{p}(X|C_2)} > \frac{\varphi(X|C_2)P(C_2)}{\varphi(X|C_1)P(C_1)} \\ \\ C_2 & \text{if} \quad \frac{\widehat{p}(X|C_1)}{\widehat{p}(X|C_2)} < \frac{\varphi(X|C_2)P(C_2)}{\varphi(X|C_1)P(C_1)} \end{cases} \tag{2.8}$$

Since the decision is undefined when $\frac{\widehat{p}(X|C_1)}{\widehat{p}(X|C_2)} = \frac{\varphi(X|C_2)P(C_2)}{\varphi(X|C_1)P(C_1)}$, for mathematical convenience we modify the above decision rule to:

$$\text{chosen class} = \begin{cases} C_1 & \text{if} \quad \frac{\widehat{p}(X|C_1)}{\widehat{p}(X|C_2)} \geq \frac{\varphi(X|C_2)P(C_2)}{\varphi(X|C_1)P(C_1)} \\ \\ C_2 & \text{otherwise} \end{cases} \tag{2.9}$$

Due to precision issues in a computational implementation, it is more convenient to use a summation rather than a series of multiplications. Since $\log(\cdot)$ is a monotonically increasing function, the decision rule can be modified to:

$$\text{chosen class} = \begin{cases} C_1 & \text{if} \quad \log\left[\frac{\widehat{p}(X|C_1)}{\widehat{p}(X|C_2)}\right] \geq \log\left[\frac{\varphi(X|C_2)P(C_2)}{\varphi(X|C_1)P(C_1)}\right] \\ \\ C_2 & \text{otherwise} \end{cases} \tag{2.10}$$

which translates to:

$$\text{chosen class} = \begin{cases} C_1 & \text{if} \quad \log \widehat{p}(X|C_1) - \log \widehat{p}(X|C_2) \geq \log\left[\frac{\varphi(X|C_2)P(C_2)}{\varphi(X|C_1)P(C_1)}\right] \\ \\ C_2 & \text{otherwise} \end{cases} \tag{2.11}$$

where, for clarity,

$$\log \widehat{p}(X|C_j) = \sum_{i=1}^{N_V} \log \widehat{p}(\boldsymbol{x}_i|C_j) \tag{2.12}$$

Due to practical considerations described later, the number of observation vectors often needs to be taken into account. Thus a normalisation factor, $\frac{1}{N_V}$ is introduced to (2.11), giving:

[3] $\widehat{p}(\boldsymbol{x}|C_j)$ is not only an approximation of $p(\boldsymbol{x}|C_j)$ due to the inherent nature of parametric representation, but also due to the limited amount of training data, resulting in a possibly poor representation.

$$\text{chosen class} = \begin{cases} C_1 & \text{if} \quad \frac{1}{N_V}\left[\log \widehat{p}(X|C_1) - \log \widehat{p}(X|C_2)\right] \geq \frac{1}{N_V} \log\left[\frac{\varphi(X|C_2)P(C_2)}{\varphi(X|C_1)P(C_1)}\right] \\ C_2 & \text{otherwise} \end{cases} \tag{2.13}$$

Let us define

$$\mathcal{L}(X|C_j) \quad = \quad \frac{1}{N_V}\sum_{i=1}^{N_V} \log \widehat{p}(X|C_j) \tag{2.14}$$

which can be interpreted as the (approximate) average log likelihood of X. Hence (2.11) can be modified accordingly:

$$\text{chosen class} = \begin{cases} C_1 & \text{if} \quad \mathcal{L}(X|C_1) - \mathcal{L}(X|C_2) \geq \frac{1}{N_V} \log\left[\frac{\varphi(X|C_2)P(C_2)}{\varphi(X|C_1)P(C_1)}\right] \\ C_2 & \text{otherwise} \end{cases} \tag{2.15}$$

We now define

$$\Lambda(X) = \mathcal{L}(X|C_1) - \mathcal{L}(X|C_2) \tag{2.16}$$

Since the true form of the pdf $p(\boldsymbol{x}|C_j)$ is unknown, the "correction" function, $\varphi(\boldsymbol{x}|C_j)$, is also unknown. Moreover, in real life situations the a-priori probabilities $P(C_1)$ and $P(C_2)$ are often not known. Hence in practice, $\frac{1}{N_V} \log\left[\frac{\varphi(X|C_2)P(C_2)}{\varphi(X|C_1)P(C_1)}\right]$ is replaced with an empirically found threshold, t. Substituting $\Lambda(X)$ and t into (2.15) yields:

$$\text{chosen class} = \begin{cases} C_1 & \text{if} \quad \Lambda(X) \geq t \\ C_2 & \text{otherwise} \end{cases} \tag{2.17}$$

Strictly speaking, the normalisation factor ($\frac{1}{N_V}$) is not necessary to make a decision. However, in practical situations variable length observations are often encountered. Since $\Lambda(X)$ is observation length independent, it allows the approximation of the distributions of $\Lambda(X)$ for true clients and known impostors, which in turn simplifies the selection of the threshold.

2.4 Gaussian Mixture Model

The approximation $\widehat{p}(\boldsymbol{x}|C_j)$ is represented by a Gaussian Mixture Model (GMM), which is capable of modelling arbitrarily complex densities [144]. For client K, $\mathcal{L}(X|C_2)$ is replaced by $\mathcal{L}(X|\lambda_{\overline{K}})$ (defined in Section 2.4.2) and $\mathcal{L}(X|C_1)$ is replaced by $\mathcal{L}(X|\lambda_K)$, defined as:

$$\mathcal{L}(X|\lambda_K) = \frac{1}{N_V} \log p(X|\lambda_K) \tag{2.18}$$

where

$$\log p(X|\lambda_K) = \sum_{i=1}^{N_V} \log p(\boldsymbol{x}_i|\lambda_K) \tag{2.19}$$

$$p(\boldsymbol{x}|\lambda) = \sum_{g=1}^{N_G} m_g \, \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \tag{2.20}$$

$$\tag{2.21}$$

and

$$\lambda = \{m_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\}_{g=1}^{N_G} \tag{2.22}$$

is the parameter set. Here, N_G is the number of Gaussians, m_g is the weight for Gaussian g (with constraints $\sum_{g=1}^{N_G} m_g = 1$ and $\forall\ g:\ m_g \geq 0$), and $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a D-dimensional Gaussian function with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$:

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right] \tag{2.23}$$

2.4.1 Maximum Likelihood Parameter Estimation

Given a set of training vectors, $X = \{\boldsymbol{x}_i\}_{i=1}^{N_V}$, the GMM parameters ($\lambda$) are estimated using the Maximum Likelihood (ML) principle:

$$\lambda = \arg\max_{\widehat{\lambda}} \left[p(X|\widehat{\lambda})\right] \tag{2.24}$$

The estimation problem can be solved using a particular instance of the Expectation Maximisation (EM) algorithm [42, 109, 116, 143]. As its name suggests, the EM algorithm is comprised of iterating two steps: the expectation step, followed by the maximisation step. GMM parameters generated by the previous iteration (λ^{old}) are used by the current iteration to generate a new set of parameters (λ^{new}), such that:

$$p(X|\lambda^{\text{new}}) \geq p(X|\lambda^{\text{old}}) \tag{2.25}$$

The process is repeated until convergence or until the increase in the likelihood after each iteration falls below a pre-defined threshold. The initial estimate is typically provided by the k-means

clustering algorithm [46] (described later). One iteration of the EM algorithm, specific to GMMs, is implemented as follows:

$$\text{for } g = 1, \cdots, N_G: \quad \text{for } i = 1, \cdots, N_V: \quad l_{g,i} = \frac{m_g \, \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)}{\sum_{n=1}^{N_G} m_n \, \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)} \tag{2.26}$$

$$\text{for } \quad g = 1, \cdots, N_G:$$

$$L_g = \sum_{i=1}^{N_V} l_{g,i} \tag{2.27}$$

$$\widehat{m}_g = \frac{L_g}{N_V} \tag{2.28}$$

$$\widehat{\boldsymbol{\mu}}_g = \frac{1}{L_g} \sum_{i=1}^{N_V} \boldsymbol{x}_i \, l_{g,i} \tag{2.29}$$

$$\widehat{\boldsymbol{\Sigma}}_g = \frac{1}{L_g} \sum_{i=1}^{N_V} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_g)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_g)^T l_{g,i} \tag{2.30}$$

$$= \frac{1}{L_g} \left[\sum_{i=1}^{N_V} \boldsymbol{x}_i \boldsymbol{x}_i^T l_{g,i} \right] - \widehat{\boldsymbol{\mu}}_g \widehat{\boldsymbol{\mu}}_g^T \tag{2.31}$$

$$\tag{2.32}$$

Once all $\widehat{m}_g, \widehat{\boldsymbol{\mu}}_g, \widehat{\boldsymbol{\Sigma}}_g$ are found, the parameters are updated:

$$\{m_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\}_{g=1}^{N_G} = \left\{ \widehat{m}_g, \widehat{\boldsymbol{\mu}}_g, \widehat{\boldsymbol{\Sigma}}_g \right\}_{g=1}^{N_G} \tag{2.33}$$

In the above, $l_{g,i} \in [0, 1]$ is the a-posteriori probability of Gaussian g given $\boldsymbol{x}_i$ and current parameters. Thus the estimates $\widehat{\boldsymbol{\mu}}_g$ and $\widehat{\boldsymbol{\Sigma}}_g$ are merely weighted versions of the sample mean and sample covariance, respectively. For a derivation of the EM algorithm for GMM parameters, the reader is directed to [24, 143, 144] or Appendix B.

Overall, the algorithm is a hill climbing procedure for maximising $p(X|\lambda)$. While it may not reach a global maximum, it is guaranteed to monotonically converge to a saddle point or a local maximum [42, 46, 114]. The above implementation can also be interpreted as an unsupervised probabilistic clustering procedure, with N_G being the assumed number of clusters.

While the initial estimate of λ can be initialised to sensible quasi-random values[4], faster convergence can be achieved when the initial estimate is provided via the k-means clustering algorithm [46], described below.

[4]By "sensible quasi-random values" we mean that the initial means are set to be equal to randomly selected data vectors, diagonal elements of covariance matrices set to 1 (with other elements set to zero) and all weights equal.

k-means

We use the Kronecker delta function, $\delta(\cdot,\cdot)$, which has a value of 1 if its two arguments match and 0 if they do not. We also use a rand(min, max) function, which generates a uniformly distributed random integer value in the $[min, max]$ interval. The essence of the k-means algorithm is described using the following pseudo-code:

```
01: for g = 1, ··· , N_G
02:     μ_g = x_rand(1,N_V)      // randomly select initial means
03: endfor
04: loop = 1
05: endloop = 10                 // empirically chosen termination condition (see Section 2.6.2)
06: finished = FALSE
07: while finished ≠ TRUE
08:     for i = 1, ··· , N_V
09:         y_i = arg min_{g=1,···,N_G} ||μ_g − x_i||     // label each vector as belonging to its closest mean
10:     endfor
11:     for g = 1, ··· , N_G
12:         n_g = Σ_{i=1}^{N_V} δ(y_i, g)                 // count the number of vectors assigned to each mean
13:         μ̂_g = (1/n_g) Σ_{i=1}^{N_V} x_i δ(y_i, g)     // find the new mean using vectors assigned to the old mean
14:     endfor
15:     same = TRUE
16:     for g = 1, ··· , N_G
17:         if μ̂_g ≠ μ_g                                  // see if the means have changed since the last iteration
18:             same = FALSE
19:         endif
20:     endfor
21:     loop = loop + 1
22:     if (same == TRUE) or (loop > endloop)
23:         finished = TRUE                               // finish if the means haven't changed since the last iteration
24:     endif                                             // or the maximum number of iterations has been reached
25:     for g = 1, ··· , N_G
26:         μ_g = μ̂_g      // update the mean vectors
27:     endfor
28: endwhile
```

In practise it is possible that while iterating at least one of the means has no vectors assigned to it, becoming a "dead" mean. This can be due to an unfortunate starting point and/or a lack of data. As such, an additional heuristic is required to attempt to resolve this situation. One example of resurrecting a "dead" mean is to make it equal to one of the vectors that has been assigned to the most "popular" mean, where the most "popular" mean is the mean that currently has the most vectors assigned to it.

Once the estimated means, $\{\boldsymbol{\mu}_g\}_{g=1}^{N_G}$, have been found, the initial weights, $\{m_g\}_{g=1}^{N_G}$, and initial covariance matrices, $\{\boldsymbol{\Sigma}_g\}_{g=1}^{N_G}$, are estimated as follows:

$$m_g = \frac{n_g}{N_V} \tag{2.34}$$

$$\boldsymbol{\Sigma}_g = \frac{1}{n_g}\sum_{i=1}^{N_V}(\boldsymbol{x}_i - \boldsymbol{\mu}_g)(\boldsymbol{x}_i - \boldsymbol{\mu}_g)^T\delta(y_i, g) \tag{2.35}$$

where n_g is defined on line 12 of the pseudo-code and y_i on line 9.

The k-means algorithm can be interpreted as a special case of the EM algorithm for GMMs. Instead of each vector being assigned a set probabilities as to which Gaussian it belongs to, each vector is assigned to only one Gaussian. Furthermore, it is assumed that the covariance matrix of each Gaussian is diagonal. We also note that the k-means algorithm can also be implemented in a somewhat different manner, for example the "splitting" LBG algorithm [97].

2.4.2 Impostor Likelihood

Background Model Set

For optimum performance, the impostor model would cover observations from all possible impostors for client K. However, by its very definition, such a requirement is ill-posed. One method to approximate the impostor model is through the use of a composite model, which is comprised of models belonging to people other than the client [146, 154] (also known as cohort models [51, 153]). In this chapter, we shall refer to such a set as the Background Model Set (BMS).

In the BMS approach, the average log likelihood that the claim for person K's identity is from an impostor is calculated using a set of models, $B = \{\lambda_b\}_{b=1}^{N_B}$:

$$\mathcal{L}(X|\lambda_{\overline{K}}) = \log\left[\frac{1}{N_B}\sum_{b=1}^{N_B}\exp\mathcal{L}(X|\lambda_b)\right] \tag{2.36}$$

where $\exp\mathcal{L}(X|\lambda_b)$ can be interpreted as $p(X|\lambda_b)$ which has been normalised to take into account the length of the observation.

In this work we have used the method described by Reynolds [146] to select the BMS for each client. The method is summarised as follows. Using training data, pair-wise distances between each client model are found. For models λ_D and λ_E with corresponding training feature vector sets X_D and X_E (which were used during the construction of the models), the distance is defined as:

$$d(\lambda_D, \lambda_E) = [\mathcal{L}(X_D|\lambda_D) - \mathcal{L}(X_D|\lambda_E)] + [\mathcal{L}(X_E|\lambda_E) - \mathcal{L}(X_E|\lambda_D)] \tag{2.37}$$

The above symmetric distance defines how similar (or close) the models λ_D and λ_E are. The background model set contains models which are the closest to as well as the farthest from the client model. While it may intuitively seem that only the close models are required (which represent the expected impostors), this would leave the system vulnerable to impostors which are very different

from the client. This is demonstrated by inspecting Eqn. (2.16) where both terms would contain similar likelihoods, leading to an unreliable opinion on the claim.

For a given client model λ_K, N_Φ closest models ($N_\Phi \geq N_B$) are placed in a temporary set Φ. Similarly, N_Ψ farthest models ($N_\Psi \geq N_B$) are placed in a temporary set Ψ. Maximally spread models from the Φ set are moved to set B_{close} using the following procedure:

1. Move the closest model from Φ to B_{close}.
2. Move λ_i from Φ to B_{close}, where λ_i is found using:

$$\lambda_i = \arg\max_{\lambda_j \in \Phi} \left[\sum_{\lambda_b \in B_{close}} \frac{d(\lambda_b, \lambda_j)}{d(\lambda_K, \lambda_j)} \right] \tag{2.38}$$

3. Repeat step (2) until $N_{B_{close}} = \frac{N_B}{2}$, where $N_{B_{close}}$ is the cardinality of B_{close}.

Next, maximally spread models from the Ψ set are moved to set B_{far} using the following procedure:

1. Move the farthest model from Ψ to B_{far}.
2. Move λ_i from Ψ to B_{far}, where λ_i is found using:

$$\lambda_i = \arg\max_{\lambda_j \in \Psi} \left[\sum_{\lambda_b \in B_{far}} d(\lambda_b, \lambda_j)\ d(\lambda_K, \lambda_j) \right] \tag{2.39}$$

3. Repeat step (2) until $N_{B_{far}} = \frac{N_B}{2}$, where $N_{B_{far}}$ is the cardinality of B_{far}.

Finally, $B = B_{close} \cup B_{far}$. The above procedures for selecting maximally spread models are required to reduce redundancy in the B set [146].

Universal Background Model

An alternative approach to approximate the impostor model is via the use of the Universal Background Model (UBM), also known as a world model [104, 149]. In this approach, pooled training data from a large number of subjects is used to construct a large mixture model as per Section 2.4.1. The average log likelihood that the claim for person K's identity is from an impostor is calculated using:

$$\mathcal{L}(X|\lambda_{\overline{K}}) = \mathcal{L}(X|\lambda_{UBM}) \tag{2.40}$$

The advantage is that the impostor likelihood is now client independent (as opposed to the BMS approach). Moreover, it has been found [148] that instead of constructing the client models directly from training data (using the ML version of the EM algorithm), lower error rates can be obtained when the models are generated by adapting the UBM using a form of maximum a-posteriori (MAP) adaptation [55, 149].

A full description of MAP adaptation is out of the scope of this chapter (the reader is referred to [54, 55, 74, 149] for details). The update equations are summarised as follows. Given UBM parameters $\lambda_{UBM} = \{\widetilde{m}_g, \widetilde{\boldsymbol{\mu}}_g, \widetilde{\boldsymbol{\Sigma}}_g\}_{g=1}^{N_G}$ and a set of training feature vectors for a specific client, $X = \{\boldsymbol{x}_i\}_{i=1}^{N_V}$, the estimated weights ($\widehat{m}_g$), means ($\widehat{\boldsymbol{\mu}}_g$), and covariances ($\widehat{\boldsymbol{\Sigma}}_g$) are found as per Eqns. (2.28)-(2.31). The parameters, $\lambda = \{m_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\}_{g=1}^{N_G}$, are updated using:

$$m_g = [\alpha \widehat{m}_g + (1-\alpha)\widetilde{m}_g]\,\gamma \tag{2.41}$$

$$\boldsymbol{\mu}_g = \alpha \widehat{\boldsymbol{\mu}}_g + (1-\alpha)\widetilde{\boldsymbol{\mu}}_g \tag{2.42}$$

$$\boldsymbol{\Sigma}_g = \left[\alpha\left(\widehat{\boldsymbol{\Sigma}}_g + \widehat{\boldsymbol{\mu}}_g\widehat{\boldsymbol{\mu}}_g^T\right) + (1-\alpha)\left(\widetilde{\boldsymbol{\Sigma}}_g + \widetilde{\boldsymbol{\mu}}_g\widetilde{\boldsymbol{\mu}}_g^T)\right)\right] - \boldsymbol{\mu}_g\boldsymbol{\mu}_g^T \tag{2.43}$$

where γ is a scale factor to make sure all weights sum to 1, while $\alpha = \frac{L_g}{L_g+r}$ is a data-dependent adaptation coefficient (L_g is found using Eqn. (2.27)) where r is a fixed relevance factor (for speech derived data, typically $r \in [8, 20]$, see [149]). It must be noted that UBM components tend to be only adapted if there is sufficient correspondence with client training data. Thus to prevent the final client models not being specific enough (leading to poor performance), the UBM must adequately represent the general client population.

2.5 Error Measures (FAR, FRR, EER)

Since the verification system is inherently a two-class decision task, it follows that the system can make two types of errors. The first type of error is a False Acceptance (FA), where an impostor is accepted. The second error is a False Rejection (FR), where a true claimant is rejected. The performance of verification systems is generally measured in terms of False Acceptance Rate (FAR) and False Rejection Rate (FRR), defined as:

$$\text{FAR} = 100\% \times \frac{\text{number of FAs}}{\text{number of impostor presentations}} \tag{2.44}$$

$$\tag{2.45}$$

$$\text{FRR} = 100\% \times \frac{\text{number of FRs}}{\text{number of true claimant presentations}} \tag{2.46}$$

Since the errors are related, minimising the FAR typically increases the FRR (and vice versa). The trade-off between FAR and FRR is adjusted by using the threshold t in Eqn. (2.17). Depending on the application, more emphasis may be placed on one error over the other. For example, in a high security environment, it may be desired to have the FAR as low as possible, even at the expense of a high FRR.

The trade-off between FAR and FRR can be graphically represented by a Receiver Operating Characteristics (ROC) plot or a Detection Error Trade-off (DET) plot [44]. The ROC plot is on a linear scale, while the DET plot is on a non-linear scale (which can improve the visual appearance of the curves).

To aid the interpretation of performance, the two error measures are often combined into one measure, called the Half Total Error Rate (HTER), which is defined as:

$$\text{HTER} = \frac{(\text{FAR}+\text{FRR})}{2} \tag{2.47}$$

The HTER can be thought of as a particular case of the Decision Cost Function (DCF) [18, 44]:

$$\text{DCF} = \text{cost(FR)} \cdot P(\text{true claimant}) \cdot \text{FRR} + \text{cost(FA)} \cdot P(\text{impostor}) \cdot \text{FAR} \tag{2.48}$$

where P(true claimant) is the prior probability that a true claimant will be presented to the system, P(impostor) is the prior probability that an impostor will be presented, cost(FR) is the cost of a false rejection and cost(FA) is the cost of a false acceptance. For the HTER, we have P(true face) $=P$(impostor face) $= 0.5$ and the costs are set to 1.

A particular case of the HTER, known as the Equal Error Rate (EER), occurs when the system is adjusted (e.g. via tuning the threshold) so that FAR = FRR on a particular dataset. We use a global threshold (common across all clients) tuned to obtain the lowest EER on the test set, following the approach often used in speaker verification [44, 51, 128]. However, we note that the posterior selection of the threshold can place an optimistic bias on the results [21].

2.6 Implementation Issues

2.6.1 EM Algorithm

Reynolds [144] suggested that the EM algorithm generally converges in 10 to 15 iterations, with further iterations resulting in only minor increases of the likelihood $p(X|\lambda)$. Since the EM algorithm can be computationally expensive, the maximum number of iterations has been limited to 10 in all experiments reported in this work.

2.6.2 k-means

The k-means algorithm is used to provide an initial estimate of the GMM parameters $\lambda = \{m_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\}_{g=1}^{N_G}$ which are used as a seed by the EM algorithm. In addition to not providing a solution optimal in the ML sense, k-means is computationally expensive (the number of operations is dependent on the size of the training set and number of Gaussians). It is the author's experience that it is only necessary to run a fixed number of iterations of the algorithm before passing the seed solution to the EM algorithm; the adequate number of iterations has been empirically found to be about 10. Letting k-means converge to its (locally) "perfect" solution usually takes a much larger number of iterations but still results in a very similar final solution by the EM algorithm.

The heart of the k-means algorithm is the $||\boldsymbol{a} - \boldsymbol{b}||$ operation which is the Euclidean distance between $\boldsymbol{a}$ and $\boldsymbol{b}$. To prevent one of the dimensions dominating the result (due to a relatively large

variance), it is necessary to first transform the training data, $X = \{\boldsymbol{x}_i\}_{i=1}^{N_V}$, so that each dimension has zero mean and unit variance:

$$\widehat{X} = \{\ \mathcal{T}(\boldsymbol{x}_i)\ \}_{i=1}^{N_V} \tag{2.49}$$

where the transformation function $\mathcal{T}(\cdot)$ is defined as:

$$\mathcal{T}(\boldsymbol{x}) \quad = \quad \Big[\ [\ (1/\sigma_d)(x_d - \mu_d)\ ]_{d=1}^{D}\Big]^T \tag{2.50}$$

and the corresponding inverse function is:

$$\mathcal{T}^{-1}(\widehat{\boldsymbol{x}}) \quad = \quad \Big[\ [\ \sigma_d \widehat{x}_d + \mu_d\ ]_{d=1}^{D}\Big]^T \tag{2.51}$$

where σ_d and μ_d are the standard deviation and the mean for the d-th dimension of D-dimensional training data X, respectively.

Once the estimated means are found through k-means, the inverse transformation $\mathcal{T}^{-1}(\cdot)$ is applied to them before the estimated covariances are calculated [Eqn. (2.35)] using the original data, X.

2.6.3 Impostor Likelihood

When using the BMS approach to calculate the impostor likelihood, one would like to use as many background persons as possible. However, as more clients are enrolled in a system, allowing the use of their models in the BMS, the slower the system would become. Due to this practical consideration, as well as the need for fixed experiment conditions, the size of the BMS is limited to 10 models.

2.6.4 Type of Covariance Matrix

The general definition of a GMM (see Section 2.4) supports full covariance matrices, i.e. a covariance matrix with all its elements. However, like many researchers, in this work we shall use diagonal covariance matrices. The reasons are explained below:

- GMMs using diagonal covariance matrices are significantly less computationally expensive to train and use than GMMs using full covariance matrices, as the inverse of a $D \times D$ matrix is not required [see Eqn. (2.23)]. Instead, only the inverse of individual diagonal elements is required.

- Density modelling using an N_G-Gaussian full covariance GMM can be well approximated using a diagonal covariance GMM with a larger number of components. Moreover, diagonal covariance GMMs with $N_G > 1$ can model distribution of feature vectors with correlated elements [149].

- Using diagonal covariance matrices reduces the number of unknown parameters. Thus less training data is required than for full covariance matrices [46].
- It has been empirically observed that diagonal covariance GMMs outperform full covariance GMMs [146, 147, 149].

Chapter 3

Verification using Speech Signals

3.1 Overview

In this chapter we first describe the difference between text-dependent and text-independent systems. A review of the human speech production process is then given, followed by a review of feature extraction approaches used in a speaker verification system. Mel Frequency Cepstral Coefficients (MFCCs), delta features and Cepstral Mean Subtraction (CMS) are covered. An alternative feature set, termed Maximum Auto-Correlation Values (MACVs), which uses information from the source part of the speech signal, is also covered. A parametric Voice Activity Detector (VAD), used for disregarding silence and noise segments of the speech signal, is briefly described.

Experiments on the telephone speech NTIMIT dataset confirm the correct implementation of the Gaussian Mixture Model classifier (described in Chapter 2) and the MFCC feature extractor by obtaining virtually the same results as presented by Reynolds in [146]. Further experiments show that the performance degradation of a verification system used in noisy conditions can be reduced by extending the feature vectors with MACV features.

3.2 Text-Dependent vs Text-Independent Systems

Speech based verification systems fall into two categories: text-dependent and text-independent. In a text-dependent system, the claimant must recite a phrase specified by the system. This is in contrast to a text-independent system, where the claimant can say whatever he or she wishes. The main advantage of a text-independent system is the general absence of idiosyncrasies in the task definition, which allows the system to be applied to many tasks[1] [44]. For this reason, in this work we concentrate on the latter category.

[1] One of the examiners of this work has also pointed out that "text-dependent systems provide lower error rates and require less enrolment data than text-independent systems. For that reason, most, if not all of the commercially deployed speaker verification systems are text-dependent. A further observation is that even if the verification system operates in a text-dependent mode, the models could still be text-independent."

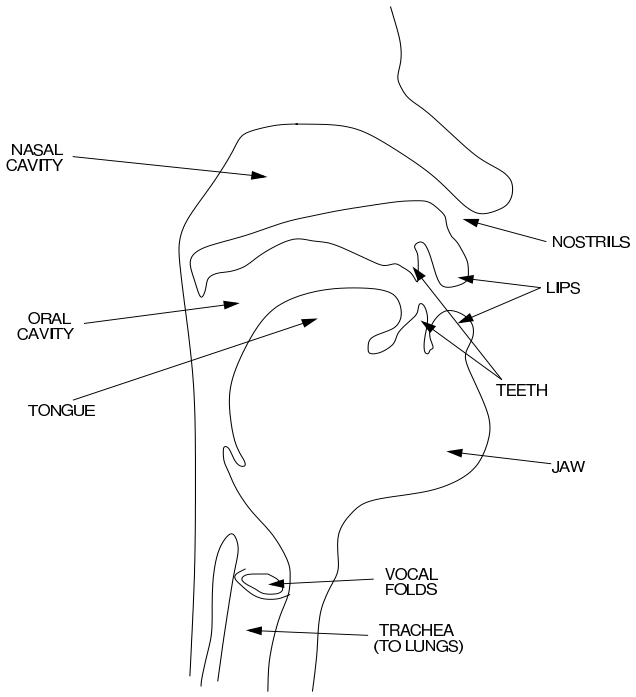

Figure 3.1: Major vocal tract components (after [162]).

3.3 Speech Production Process

Speech can be categorised into two main sound types: voiced and unvoiced. Voiced sounds are produced as follows. Quasi-periodic opening and closing of the vocal folds, measured in terms of fundamental or pitch frequency (often abbreviated as $F0$), generates a glottal wave composed of energy at $F0$ and at harmonics of $F0$ (i.e. integral multiples of $F0$). The glottal wave is then passed through the vocal tract (see Figure 3.1). The vocal tract can be modelled as an acoustic tube (starting at the vocal folds and terminating at the lips) with resonances and anti-resonances. The resonances are referred to as formants, and are abbreviated to Fi, where $F1$ is the formant with the lowest frequency. The vocal tract, in effect, amplifies energy around formant frequencies and attenuates energy at anti-resonant frequencies. Formant frequencies are changed by modifying the configuration of the articulators (such as the tongue, jaw, lips and teeth), allowing the production of different sounds (e.g. [ʌ] vs [ε][2]). In normal speech the articulators are almost constantly moving, indicating that voiced sounds are at best quasi-stationary over short periods of time (tens of milliseconds) [162].

The opening and closing of the vocal folds is accomplished by the following mechanism. At the start of the cycle the vocal folds are closed. Air pressure beneath the vocal folds is increased (due to the constriction of the lungs) and once it overcomes the resistance of the vocal fold closure, it

[2]Here we use the International Phonetic Alphabet [75]. The sound [ʌ] occurs in the underlined portion of these words: c<u>u</u>p, b<u>u</u>t, while the sound [ε] occurs in h<u>ea</u>d and b<u>e</u>t.

forces the vocal folds apart. Shortly afterwards the air pressure is temporarily equalised, and the vocal folds close again, completing the cycle. The cycle has a frequency in the approximate ranges of 60-160 Hz for males and 160-400 Hz for females [73, 131] (average values are approximately 132 Hz and 223 Hz for males and females, respectively [167]). Changes in $F0$ by the speaker are used to denote prosodic information, such as whether a spoken sentence is a statement or a question. While most speakers are capable of changing their $F0$ by two octaves, variation of $F0$ is limited in normal speech since extremes of $F0$ require increased labour.

During the production of unvoiced sounds, the vocal folds do not vibrate (they remain open). Instead, some of the articulators constrict a point in the vocal tract, causing high speed air flow, which in turn produces an aperiodic noise-like signal. The signal is then shaped by the section of the vocal tract in front of the constriction.

As a simplification, the speech signal production process can be thought of as being composed of two parts:

1. The source part. Here the source signal may be either periodic, resulting in voiced sounds, or noisy and aperiodic, resulting in unvoiced sounds.
2. The filter part. Here the source signal is filtered to produce a particular sound.

Thus for voiced sounds the source part generates a signal with spectral energy concentrated at $F0$ (the fundamental frequency) and all its harmonics. The signal is then filtered by the filter part, where the required formants are emphasised, while other parts of the signal are attenuated.

Apart from linguistic information, speech carries person dependent information due to the largely unique configuration of the vocal tract and vocal folds for each person. This causes the time course of $F0$ and the formant frequencies to be person dependent [162].

3.4 Feature Extraction From Speech Signals

Popular speech based verification systems use information from the filter part in the form of a short-time Fourier spectrum represented by Mel Frequency Cepstral Coefficients (MFCCs) [12, 44, 146, 149]. While MFCC features are quite effective for discriminating speakers, they are affected by channel distortion and/or ambient noise. This causes a degradation in the performance of a verification system due to a mismatch between training and testing conditions. There are two popular techniques to reduce the effects of channel distortion and ambient noise: the use of delta (regression) features [50, 165] and Cepstral Mean Subtraction (CMS) [50].

Wildermoth and Paliwal [189] proposed an alternative feature set, termed Maximum Auto-Correlation Values (MACVs), which utilises information from the source part of the speech signal; as will be shown, the use of MACV features reduces the performance degradation present due to mismatched conditions.

3.4.1 MFCC Features

In MFCC feature extraction, the speech signal is analysed on a frame by frame basis, with a typical frame length of 20 ms and a frame advance of 10 ms. For a frame length of 20 ms it can be assumed that the speech signal is stationary, allowing the computation of the short-time Fourier spectrum [129].

Let us denote the speech frame as $\boldsymbol{s}^T = [\ s_i\ ]_{i=1}^{N_S}$, where N_S is the number of samples (for a speech signal sampled at 8 kHz, $N_S = 160$ when using 20 ms frames). Each frame is multiplied by a Hamming window to reduce the effects of spectral leakage [140]:

$$\widehat{s}_i = s_i h_i, \quad i = 1, 2, \cdots, N_S \tag{3.1}$$

where

$$h_i = 0.54 - 0.46 \cos\left(\frac{2\pi(i-1)}{N_S - 1}\right), \quad i = 1, 2, \cdots, N_S \tag{3.2}$$

The complex spectrum of $\widehat{\boldsymbol{s}}^T = [\ \widehat{s}_i\ ]_{i=1}^{N_S}$ is then obtained using the Fast Fourier Transform (FFT) algorithm [139, 140]. The square of the magnitude of the complex spectrum is represented as $\tilde{\boldsymbol{s}}$ (in our experiments we use a 2048 point representation).

A set of triangular-shaped filters is spaced according to the Mel-scale [135], simulating the processing done by the human ear [73, 119, 120]. For filters chosen to cover the telephone bandwidth, the centre frequencies are (in Hz): 300, 400, 500, 600, 700, 800, 900, 1000, 1149, 1320, 1516, 1741, 2000, 2297, 2639, 3031 and 3482. Moreover, to simulate critical bandwidths [135], the upper and lower passband frequencies of each filter are the centre frequencies of adjacent filters. For the filter centred at 300 Hz, the lower passband frequency is 200 Hz, while the upper passband frequency for the filter centred at 3482 Hz is 4000 Hz. The responses of $N_F = 17$ filters are shown in Figure 3.2.

Let $\boldsymbol{f}_i$ be the magnitude-squared response of the i-th filter in the frequency domain. The energy output of each filter is obtained using:

$$e_i = \boldsymbol{f}_i^T \tilde{\boldsymbol{s}}, \qquad i = 1, 2, \cdots, N_F \tag{3.3}$$

The above equation can be rewritten to obtain an N_F-dimensional energy vector $\boldsymbol{e}$:

$$\boldsymbol{e} = \boldsymbol{F}^T \tilde{\boldsymbol{s}} \tag{3.4}$$

where $\boldsymbol{F} = [\ \boldsymbol{f}_1\ \boldsymbol{f}_2\ \cdots\ \boldsymbol{f}_{N_F}\ ]$. We note that Eqn. (3.4) can be interpreted as a form of dimensionality reduction. In effect, the energy vector $\boldsymbol{e}$ represents the smoothed (Mel-warped) spectrum of $\tilde{\boldsymbol{s}}$, which is a good representation of the filter part of speech [162].

In order to obtain amplitude normalisation, as well as to take into account the diagonal covariance matrix constraint in the GMM classifier (see Section 2.6.4), a form of 1D Discrete Cosine Transform (1D DCT) [59] is applied to the log version of $\boldsymbol{e}$:

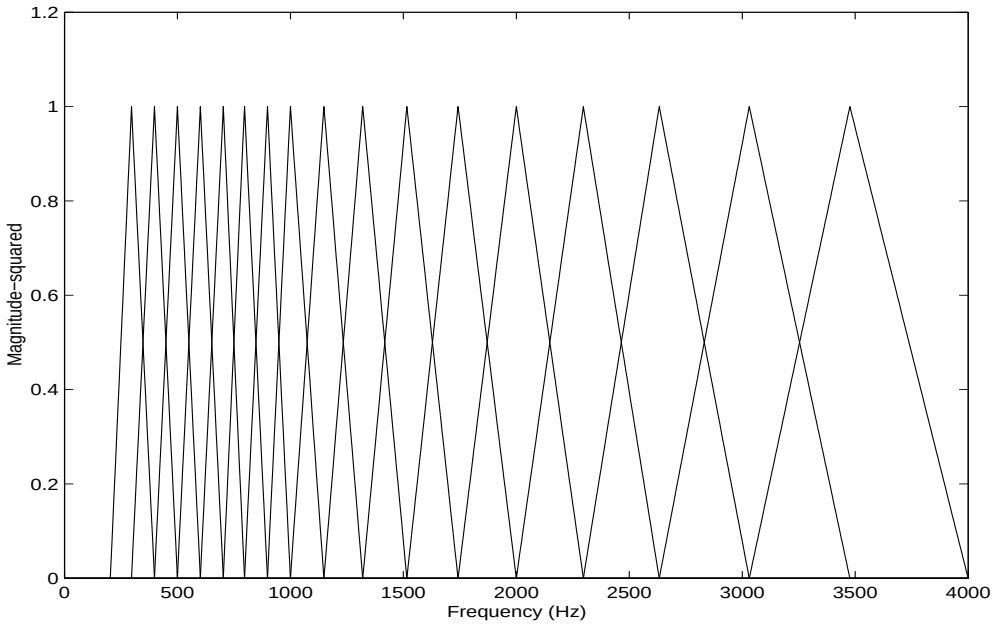

Figure 3.2: An example of a Mel-scale filter bank.

$$g_i = \frac{1}{N_F} \sum_{j=1}^{N_F} \log\left(e_j\right) \cos\left(\frac{\pi(i-1)(2j-1)}{2N_F}\right) \quad i = 1, 2, \cdots, N_F \tag{3.5}$$

One reason for using the log version of $\boldsymbol{e}$ is explained in Section 3.4.2. Eqn. (3.5) can be rewritten in matrix notation:

$$\boldsymbol{g} = \boldsymbol{C}^T \boldsymbol{e}_{\log} \tag{3.6}$$

where

$$\boldsymbol{e}_{\log}^T = \left[\ \log\left(e_i\right)\ \right]_{i=1}^{N_F} \tag{3.7}$$

and $\boldsymbol{C} = [\ \boldsymbol{c}_1\ \boldsymbol{c}_2\ \cdots\ \boldsymbol{c}_{N_F}\ ]$, where

$$\boldsymbol{c}_i = \left[\ \frac{1}{N_F} \cos\left(\frac{\pi(i-1)(2j-1)}{2N_F}\right)\ \right]_{j=1}^{N_F} \qquad i = 1, 2, \cdots, N_F \tag{3.8}$$

are the 1D DCT basis vectors.

In Eqn. (3.5), it can be seen that g_1 represents the average log energy of the spectrum. Since we prefer to use a feature set which is not susceptible to varying background noise and loudness of speech, g_1 is omitted, resulting in a $(N_F - 1)$-dimensional MFCC feature vector:

$$\boldsymbol{x} \ = \ [\ g_2\ \ g_3\ \ \cdots\ \ g_{N_F}\ ]^T \tag{3.9}$$

Disregarding g_1 can be interpreted as a form of amplitude normalisation.

Another popular speech feature extraction method is based on Linear Prediction Cepstral Coefficients (LPCC) [129], which originated from speech compression applications [8, 73, 103]. However, MFCC features are used for experiments in this work since it has been suggested that they are generally more robust than LPCC features for speaker recognition applications [145].

3.4.2 CMS Features

Let us assume that a signal z is comprised of an original speech signal a that is being filtered by a channel[3] b:

$$z = a * b \tag{3.10}$$

where $*$ denotes the convolution operation. Thus in the frequency domain the above translates to:

$$Z = AB \tag{3.11}$$

where Z, A and B are the spectra of z, a and b, respectively. Taking the logarithm of Eqn. (3.11) yields:

$$\log(Z) = \log(A) + \log(B) \tag{3.12}$$

Hence in the log domain, the speech signal and the channel are superimposed. As the energy vector $\boldsymbol{e}$ from Eqn. (3.4) represents the smoothed (Mel-warped) spectrum, Eqn. (3.11) is analogous to:

$$\boldsymbol{e}^{\,T} = [\ e_i\ ]_{i=1}^{N_F} = [\ e_i^a e_i^b\ ]_{i=1}^{N_F} \tag{3.13}$$

where $\boldsymbol{e}^{\,a}$ and $\boldsymbol{e}^{\,b}$ represent the smoothed spectrum of a and b, respectively. Taking the log of (3.13) yields:

$$\boldsymbol{e}_{\log}^{\,T} = [\ \log(e_i)\ ]_{i=1}^{N_F} = \left[\ \log(e_i^a) + \log(e_i^b)\ \right]_{i=1}^{N_F} \tag{3.14}$$

Applying 1D DCT decorrelation to $\boldsymbol{e}_{\log}$ yields:

$$\begin{aligned} \boldsymbol{g} &= \boldsymbol{C}^T \left(\boldsymbol{e}_{\log}^{\,a} + \boldsymbol{e}_{\log}^{\,b} \right) && (3.15)\\ &= \boldsymbol{C}^T \boldsymbol{e}_{\log}^{\,a} + \boldsymbol{C}^T \boldsymbol{e}_{\log}^{\,b} && (3.16)\\ &= \boldsymbol{g}^{\,a} + \boldsymbol{g}^{\,b} && (3.17) \end{aligned}$$

[3] For example, a telephone channel.

Thus the effect of the channel is an additive component on the MFCC feature vector:

$$\boldsymbol{x} = \boldsymbol{x}^{\,a} + \boldsymbol{x}^{\,b} \tag{3.18}$$

Let us define the mean MFCC feature vector for an entire utterance, $\{\boldsymbol{x}_i\}_{i=1}^{N_V}$, as:

$$\begin{aligned} \boldsymbol{x}^{\,\mu} &= \frac{1}{N_V}\sum_{i=1}^{N_V}\boldsymbol{x}_i && (3.19)\\ &= \frac{1}{N_V}\sum_{i=1}^{N_V}\left(\boldsymbol{x}_i^{\,a} + \boldsymbol{x}_i^{\,b}\right) && (3.20)\\ &= \frac{1}{N_V}\sum_{i=1}^{N_V}\boldsymbol{x}_i^{\,a} + \frac{1}{N_V}\sum_{i=1}^{N_V}\boldsymbol{x}_i^{\,b} && (3.21) \end{aligned}$$

Assuming that channel characteristics are time invariant leads to:

$$\boldsymbol{x}^{\,\mu} = \left(\frac{1}{N_V}\sum_{i=1}^{N_V}\boldsymbol{x}_i^{\,a}\right) + \boldsymbol{x}^{\,b} \tag{3.22}$$

Moreover, if we assume that speech energy is uniformly distributed over the entire spectrum for the duration of the utterance (i.e., the average speech spectrum is flat), then the term $\frac{1}{N_V}\sum_{i=1}^{N_V}\boldsymbol{x}_i^{\,a}$ tends toward zero [13]. Thus $\boldsymbol{x}^{\,b}$ can be found using Eqn. (3.19) and we can obtain channel compensated vectors using:

$$\left\{\boldsymbol{x}_i^{\,\text{comp}}\right\}_{i=1}^{N_V} = \{\boldsymbol{x}_i - \boldsymbol{x}^{\,\mu}\}_{i=1}^{N_V} \tag{3.23}$$

The above procedure is known as Cepstral Mean Subtraction (CMS) and Cepstral Mean Normalisation (CMN) [9, 13, 50, 145, 147].

As shown in Eqn. (3.22), the mean feature vector also represents the average speech spectrum; in most practical applications the length of the utterance is not long enough for the second assumption to be valid [13, 58], thus removal of the mean from MFCC features is a double-edged sword: on one hand it makes the verification system more robust to channel mismatches, while on the other it reduces the accuracy of the system in matched conditions (since the average speech spectrum contains speaker information).

In Eqn. (3.22) it is assumed that the channel characteristics are not changing over time. However, if the characteristics are time-variant, an adaptive bias removal method, such as RASTA processing [68, 69], can be used.

For the sake of convenience, we shall refer to MFCC features with CMS applied simply as CMS features.

3.4.3 Delta (Δ) Features

It has been shown that transitional spectrum information contains information which is relatively complementary to instantaneous spectral information, as well as being less affected by channel effects [165]. Given a sequence of instantaneous spectrum feature vectors, $\{\boldsymbol{x}_i\}_{i=1}^{N_V}$, the corresponding transitional spectrum feature vectors are calculated using a modified 1st order orthogonal polynomial fit [50, 79, 165]:

$$\Delta\boldsymbol{x}_i = \frac{\sum_{k=-K}^{K} h_k k \; \boldsymbol{x}_{i+k}}{\sum_{k=-K}^{K} h_k k^2} \quad \text{for} \quad i = (K+1) \;\; \text{to} \;\; (N_V - K) \tag{3.24}$$

where $\boldsymbol{h}$ is a $2K+1$ dimensional symmetric window vector. Typically, $K = 2$ and a rectangular window is used [12, 147, 149] (thus $\Delta\boldsymbol{x}_i$ is the slope of the least squares linear fit over the duration of the window).

Transitional spectrum features are better known as delta features. Consequently, instantaneous spectrum features are often referred to as static features [147].

While being more robust to channel effects, delta features do not perform as well as static features in matched conditions [165]. Thus it is general practice to combine the two feature sets by concatenating the delta feature vector with the static feature vector:

$$\boldsymbol{y} = \left[\; \boldsymbol{x}^T \; \Delta\boldsymbol{x}^T \; \right]^T \tag{3.25}$$

If we treat the delta and static features as two separate sources of information, then the above concatenation operation can be interpreted as a form of information fusion (see Chapter 6 for more information).

Since it is convenient to have the same number of delta and static feature vectors, the "missing" delta feature vectors are generated using:

$$\Delta\boldsymbol{x}_i = \Delta\boldsymbol{x}_K \quad \text{for} \quad i = 1 \;\; \text{to} \;\; K \tag{3.26}$$

$$\Delta\boldsymbol{x}_i = \Delta\boldsymbol{x}_{N_V-K} \quad \text{for} \quad i = (N_V - K + 1) \;\; \text{to} \;\; N_V \tag{3.27}$$

Delta-delta (or acceleration) feature vectors ($\Delta\Delta\boldsymbol{x}$) can be obtained by applying Eqn. (3.24) to delta feature vectors. However, it has been suggested that use of delta-delta features provides negligible performance improvements in speaker verification [44].

3.4.4 MACV Features

In MFCC features (and hence CMS and delta features) mainly the filter part of the speech signal is effectively utilised. Two possible ways of using pitch (or pitch-related) information are:

1. Using a dedicated pitch-based verification sub-system and fusing its output with that of a traditional speaker verification system before reaching the final accept/reject decision. The front-end for the dedicated sub-system can be comprised, for example, of a voiced/unvoiced frame detector, followed by a pitch frequency extractor.

2. Incorporating pitch or pitch-related information directly into the feature vector.

In this chapter we will pursue the second approach. The simplest method for detecting the pitch period is by using the auto-correlation function, which for a speech frame $\boldsymbol{s}^T = [\ s_i\ ]_{i=1}^{N_S}$ is defined as [140]:

$$R(k) = \frac{1}{N_S} \sum_{i=1}^{N_S-k} s_i\ s_{i+k} \qquad k = 0, 1, \cdots, N_S - 1 \tag{3.28}$$

If $\boldsymbol{s}$ is periodic with a period equal to P samples, then $\{R(k)\}_{k=0}^{N_S-1}$ will show a peak at a lag equal to P. The pitch frequency is typically between 60-160 Hz for males and 160-400 Hz for females [73, 131], indicating that valid pitch lags are approximately between 2.5ms and 16ms. Thus the period of $\boldsymbol{s}$ can be found by searching for the maximum of $\{R(k)\}_{k=0}^{N_S-1}$ in the 2.5ms to 16ms range. Due to the harmonic nature of the formants, this approach also allows the recovery of the pitch period when using a telephone channel (which limits the bandwidth of speech signals to between 300 and 3400 Hz).

Unfortunately the auto-correlation method (and other time-domain techniques, such as the Normalised Cross-Correlation Method [7] and the Average Magnitude Difference Method [121, 155]), suffer from pitch doubling and halving as well as other errors [73].

If the signal is periodic with period P, it is also periodic with period $2P$, $3P$, etc. Hence, $\{R(k)\}_{k=0}^{N_S-1}$ will also have maxima at lags equal to $2P$, $3P$, etc. Due to the presence of interfering signals (e.g. noise) and since the speech signal is only quasi-stationary (e.g. the pitch can drift during the duration of the frame), one of the "extra" maxima might be the global maximum. Thus the pitch period can be identified as $2P$, which is referred to as pitch halving. When the M-th formant dominates the signal's energy (which can easily occur when using a telephone channel), there will be a maximum at a lag equal to P/M; thus the pitch period can be identified as $P/2$, which is referred to as pitch doubling.

When the speech frame is unvoiced, the above mentioned pitch extraction techniques essentially provide random values [73], indicating that their output cannot be incorporated into the feature vector for each frame.

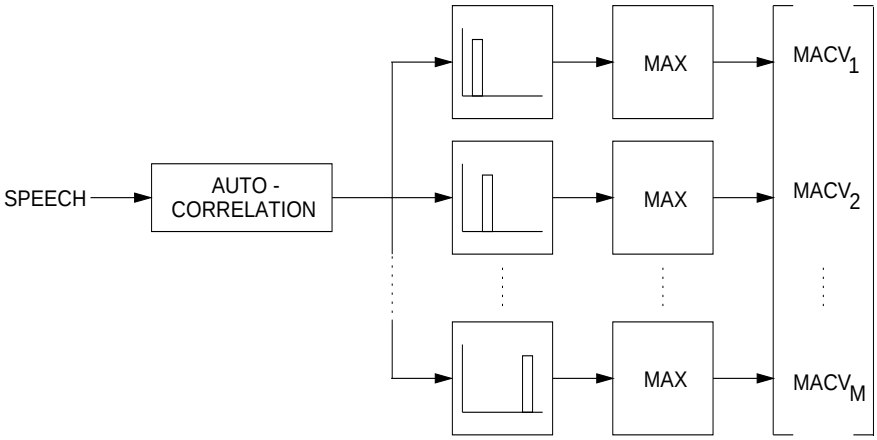

Figure 3.3: MACV feature extractor (after [189]).

A feature set known as Maximum Auto-Correlation Values (MACVs) [189] addresses the above problems by deriving pitch related information from the auto-correlation function rather than trying to find the pitch period directly. This is accomplished by dividing the auto-correlation function into several pitch-candidate regions and then finding the maximum value in each region. Formally, the MACV features are obtained as follows:

1. Compute the auto-correlation function $\{R(k)\}_{k=0}^{N_S-1}$.
2. Normalise $\{R(k)\}_{k=0}^{N_S-1}$ by its maximum, i.e., $\left\{\widehat{R}(k)\right\}_{k=0}^{N_S-1} = \left\{\frac{R(k)}{R(0)}\right\}_{k=0}^{N_S-1}$.
3. Divide the higher portion (from 2.5 ms to 16 ms) of $\{\widehat{R}(k)\}_{k=0}^{N_S-1}$ into N_M equal parts (typically $N_M = 5$, see [189]).
4. Find the maximum value of each of the N_M parts.
5. The N_M Maximum Auto-Correlation Values (MACVs) form an N_M-dimensional feature vector.

A conceptual block diagram of this process is shown in Figure 3.3. It should be noted that the MACV feature set can also be considered as a non-linear approximation of the mid-section of the auto-correlation function.

Since the MACVs for an unvoiced frame will be relatively low when compared to MACVs for a voiced frame, the MACV feature set also contains voicing information. Moreover, since the MACV feature set does not attempt to extract salient features of the spectrum for each frame (as in MFCC features) it may be less affected by background noise; this conjecture is experimentally tested in Section 3.5.2.

3.4.5 Voice Activity Detector

In addition to pauses between words, the start and the end of speech signals in many datasets often contains only background noise. Since these segments do not contain speaker dependent information, it would be advantageous to disregard them during modelling and testing. Decomposing a signal into speech and non-speech segments can be approximately accomplished via a Voice Activity Detector (VAD). Rather than using the heuristic energy based detector presented by Reynolds in [144] (seemingly used in his following work, i.e., [146, 147, 148, 149]) we have developed a parametric VAD based on the work by Haigh [64, 65].

The parametric VAD is implemented as follows. Each utterance is completely parametrised using a given feature extraction technique, resulting in a set of feature vectors, $X = \{\boldsymbol{x}_i\}_{i=1}^{N_V}$. A single Gaussian GMM (representing the background noise) is constructed using the first N_{noise} vectors[4]. Using the background noise GMM (λ_{noise}), the log-likelihood for each vector is found. If the log-likelihood for a given feature vector is below a predefined threshold (T_{VAD}), the vector is classified as containing speech. The following threshold has been empirically found to provide good discrimination ability across various parametrisation methods:

$$T_{\text{VAD}} = \frac{1}{3} l_{\text{noise}} \tag{3.29}$$

where

$$l_{\text{noise}} = \frac{1}{N_{\text{noise}}} \sum_{i=1}^{N_{\text{noise}}} \log p(\boldsymbol{x}_i | \lambda_{\text{noise}}) \tag{3.30}$$

The result of typical speech selection is shown in Figure 3.4.

A few words of caution. The VAD described here assumes that the initial part of the signal does not contain speech. Moreover, for this VAD to work well, the background noise conditions have to be stationary for the duration of the speech utterance.

3.5 Experiments

3.5.1 Test of GMM and MFCC Implementations

In this section the implementations of the Gaussian Mixture Model classifier (described in Chapter 2) and the MFCC feature extractor are tested by comparing the results obtained with the results published by Reynolds in [146].

Reynolds' experiment setup is as follows. Speech signals are taken from the test section of the telephone speech NTIMIT dataset [80], which contains 10 utterances each from 168 persons (56 female and 112 male). The utterances have an average duration of approximately 4 seconds and have been degraded by the effects of a carbon button microphone and telephone line conditions (local and long-distance). The first eight utterances (sorted alpha-numerically by filename) are used for training the models, while the last two are used for testing purposes.

[4] For the NTIMIT dataset [80], $N_{\text{noise}} = 10$.

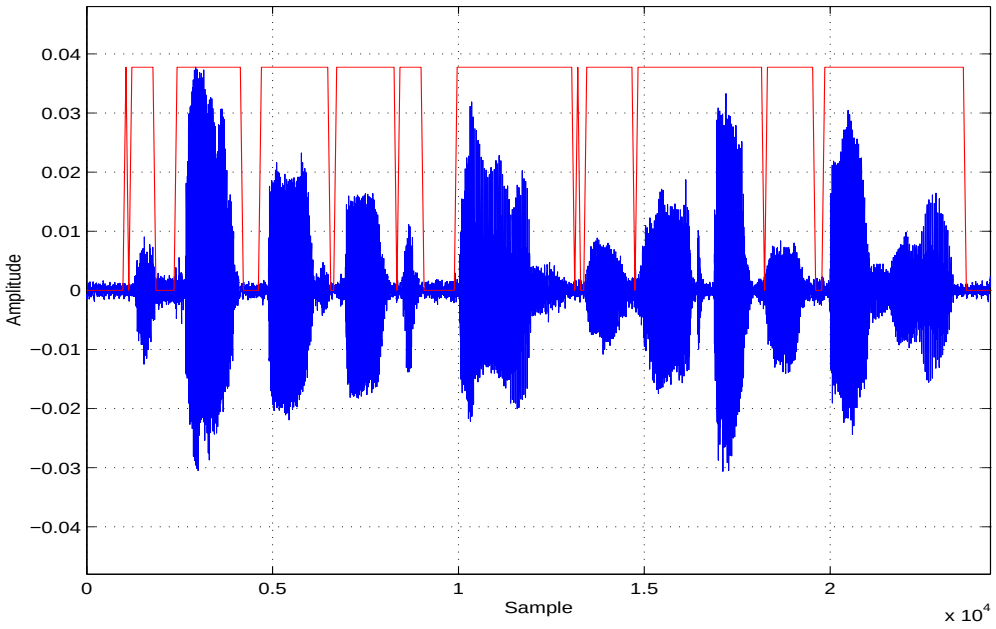

Figure 3.4: Typical result of speech selection using the parametric VAD. High level of the red line indicates the segments that have been selected as speech. The above utterance is: "before thursday's exam, review every formula".

Source	EER (%)
Reynolds [146]	7.19
This work	7.22

Table 3.1: Comparison of the EER achieved on NTIMIT, using Reynolds' [146] implementation of the MFCC feature extractor and the GMM classifier (32 Gaussians), as well as the implementation used in this work.

GMMs with 32-Gaussians were used as client models. For each client, their own test utterances were used to simulate true claimant accesses. Impostor accesses were simulated using utterances other than from the client and from the people whose models were used in the Background Model Set (BMS) for the client. The BMS for each client was comprised of 10 models (N_Φ and N_Ψ were set to 20; see Section 2.4.2). In total, there were 336 true claimant tests and 52752 impostor tests. The decision threshold was set to obtain performance as close as possible to EER.

As it can be seen in Table 3.1, the results are virtually the same, suggesting that the GMM classifier and the MFCC feature extractor were implemented correctly. The negligible difference could be attributed to the tuning of the VAD.

Number of Gaussians	1	2	4	8	16	32	64
EER (%)	14.28	12.73	11.73	9.96	9.58	9.99	11.16

Table 3.2: EER on NTIMIT for various number of Gaussians, using MFCC parametrisation of speech signals. (The experiment setup was different than the one used for Table 3.1).

3.5.2 Evaluation of MACVs in Noisy Conditions

The experiments used the NTIMIT dataset described in the previous section, though with a different experiment setup that addresses some of the shortcomings of the setup used in [146].

In particular, rather than using different impostors for each client, twenty fixed persons (first 10 females and last 10 males, alpha-numerically sorted by subject ID) were selected to be the impostors. The remaining 148 persons were used as clients. As in [146], the BMS for each client was comprised of 10 models (N_Φ and N_Ψ were set to 20; see Section 2.4.2); the BMS was constructed by considering the other 147 client models. Furthermore, the first six sentences for each client were used for model training purposes, leaving the last four sentences for simulating true claimant tests. Impostor accesses were simulated using the last four utterances from each impostor. In total there were 592 (148×4) true claimant tests and 11840 ($20 \times 4 \times 148$) impostor tests. The decision threshold was set to obtain performance as close as possible to the EER.

In the first experiment we studied the effect of the number of Gaussians on verification performance while using MFCC features. From the results shown in Table 3.2 it can be observed that the performance starts to level off at eight Gaussians. Increasing the number of Gaussians to 16 causes only minor performance gains. Further increases in the number of Gaussians reduces the performance, indicating that overfitting is occurring [46, 114]. Overfitting presents itself when the classifier is "too tuned" to the training data, resulting in poor generalisation on test data. Taking into account Occam's Razor principle [46, 114], which in effect suggests to use the simplest solution that provides adequate performance, the number of Gaussians in the second experiment was fixed at eight.

In the second experiment, the performance of each of the following feature sets was found: MFCC, CMS, MACV, MFCC+Δ, MFCC+Δ+MACV, CMS+Δ and CMS+Δ+MACV. A feature vector of type MFCC+Δ indicates that the MFCC feature vector ($\boldsymbol{x}$) has been concatenated with the feature vector containing delta versions of the MFCC features ($\Delta\boldsymbol{x}$). Similarly, MFCC+Δ+MACV indicates that the MACV feature set has also been appended.

Results were obtained for non-corrupted (clean) test utterances as well as for noisy test utterances where the SNR was varied from 28 dB to -8 dB. The utterances were corrupted by adding stationary white Gaussian noise, simulating background noise. The results are presented in Figures 3.5 to 3.7.

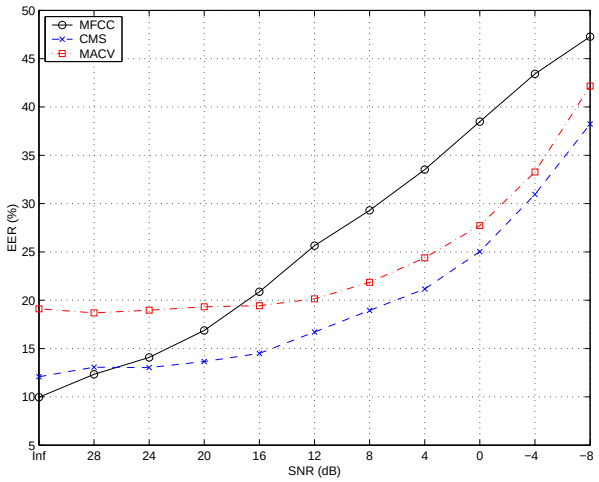

Figure 3.5: EER of baseline features (MFCC, CMS and MACV) for decreasing SNR.

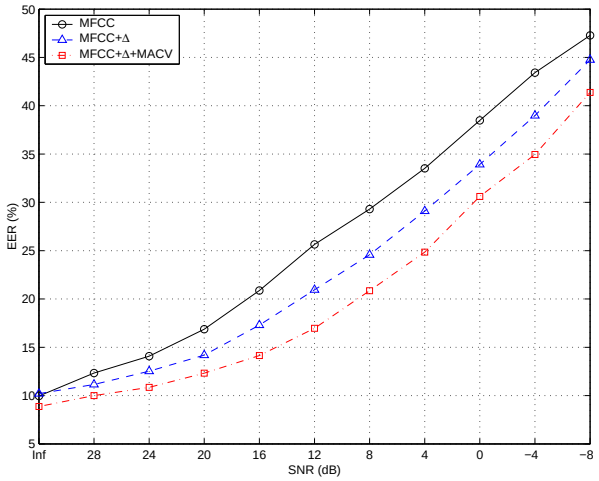

Figure 3.6: As per Figure 3.5, but using MFCC based features (MFCC, MFCC+Δ, MFCC+Δ+MACV).

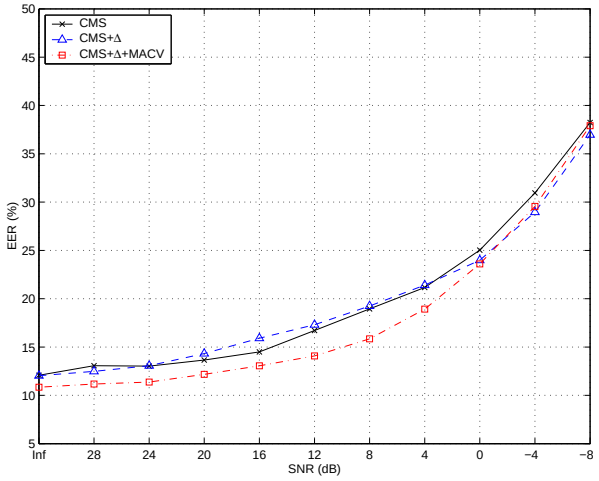

Figure 3.7: As per Figure 3.5, but using CMS based features (CMS, CMS+Δ, CMS+Δ+MACV).

In Figure 3.5 it can be seen that the CMS features are the least affected by changes in the SNR[5], at the expense of slightly worse performance than MFCC features on clean speech (as expected; see Section 3.4.2). MFCC features are the most affected by noise, with rapid degradation in performance as the SNR is lowered. Performance of MACV features in clean and low noise conditions (SNR $>$ 16 dB) is not as good as for MFCC and CMS features, indicating that pitch and voicing information is not sufficient by itself to distinguish speakers. However, as the SNR drops to 16 dB and lower, MACVs perform better than MFCCs, suggesting that MACV features are more immune to the effects of noise.

In Figure 3.6 it can be observed that extending the MFCC feature vector with delta features reduces the performance degradation as the SNR is lowered. Extending the MFCC+Δ feature vector with MACV features obtains slightly better performance on clean speech and further reduces the performance degradation. However, by comparing Figures 3.6 and 3.7 it can be seen that CMS features obtain better performance than the MFCC+Δ+MACV feature set for SNRs of 16 dB and lower.

Figure 3.7 shows that extending the CMS feature vector with corresponding delta features causes only minor differences. Extending the CMS+Δ feature vector with MACV features alleviates some of the performance loss experienced by CMS features in clean conditions, and causes the performance in noisy conditions to be visibly improved up to a SNR of 4 dB.

These results thus support the conjecture described in Section 3.4.4, and suggest that use of the MACV feature set has beneficial effects on the performance of a verification system in noisy conditions.

3.6 Summary and Future Directions

This chapter first reviewed the human speech production process and feature extraction approaches used in a speaker verification system. Mel Frequency Cepstral Coefficients (MFCCs), delta (regression) features and Cepstral Mean Subtraction (CMS) were covered. An alternative feature set, termed Maximum Auto-Correlation Values (MACVs), which utilises information from the source part of the speech signal, was also covered. A parametric Voice Activity Detector (VAD), used for disregarding silence and noise segments of the speech signal, was briefly described.

Experiments on the telephone speech NTIMIT dataset suggest that the implementation of the Gaussian Mixture Model classifier (described in Chapter 2) and the MFCC feature extractor are correct – virtually the same results as Reynolds in [146] were obtained. Further experiments show that the performance degradation of a verification system used in noisy conditions can be reduced by extending MFCC or CMS feature vectors with MACV features.

[5] We use the following definition of the Signal to Noise Ratio, expressed in decibels (dB): $\text{SNR} = 10 \log_{10}(\sum_i s_i^2 / \sum_i (s_i - n_i)^2)$, where s_i and n_i are the i-th samples from the original and noisy speech signals, respectively.

Within the area of automatic speech recognition it has been been shown that Spectral Subband Centroid (SSC) features [52, 130] and biologically inspired Zero-Crossing with Peak Amplitude (ZCPA) features [84] are quite robust to the effects of additive noise. While the speaker verification task is different from the speech recognition task, SSC and/or ZCPA features may still contain person-dependent information. As such, it would be interesting to evaluate their usefulness for robust person verification purposes. Preliminary results for SSC features are given in [172].

Chapter 4

Verification using Frontal Face Images

4.1 Overview

In this chapter we first overview important publications in the field of frontal face recognition. Geometric features, templates, Principal Component Analysis (PCA), pseudo-2D Hidden Markov Models (HMM), Elastic Graph Matching (EGM), as well as other points are covered. Relevant issues, such as the effects of an illumination direction change and the use of different face areas, are also covered.

A feature set dubbed DCT-mod2 is proposed. The feature set uses polynomial coefficients derived from 2D DCT coefficients of spatially neighbouring blocks. Its robustness and performance is evaluated against three popular feature sets for use in an identity verification system subject to illumination direction changes. Results on the multi-session VidTIMIT dataset suggest that the proposed feature set is most robust than 2D Gabor wavelets, standard 2D DCT coefficients and PCA (eigenface) derived features (with and without histogram equalisation).

As a side effect, it is also shown that when using the GMM classifier with local features (such as Gabor wavelets or 2D DCT derived features), the spatial relationships between face parts (e.g. eyes and nose) are disregarded. Such a face recognition system can surprisingly still provide good performance.

The fragility of PCA derived features to illumination changes is addressed by introducing a pre-processing step which involves applying DCT-mod2 feature extraction to the original face image. A pseudo-image is then constructed by placing all DCT-mod2 feature vectors in a matrix on which PCA feature extraction is then performed. It is shown that the enhanced PCA technique is robust to illumination changes as well as retaining the positive aspects of traditional PCA, i.e. robustness to compression artefacts and noisy images, which might be important in forensic and law enforcement applications.

4.2 Summary of Past Face Recognition Approaches

This section presents a concise overview of several important and/or popular approaches to automatic face recognition. Due to the amount of work done in this area, an in-depth review is beyond the scope. As such, the reader is also directed to the following survey articles: [36, 62, 90, 192, 193].

In general, an appearance based face recognition system can be thought of as being comprised of:

1. Face localisation and segmentation
2. Feature extraction and classification

The first stage usually provides a size normalised face image (with eyes at fixed locations). Illumination normalisation may also be performed – however, this might not be necessary if the feature extraction method is robust to illumination changes.

From here on we shall assume that the face has been located, or that images given to the system contain only one face, set against a uniform background. In other words, we shall concentrate on the second stage. Reviews of face localisation algorithms can be found in [70, 191].

There are many approaches to face recognition – ranging from the Principal Component Analysis (PCA) based approach (also known as eigenfaces) [115, 174], Elastic Graph Matching (EGM) [45, 94], Artificial Neural Networks [93, 175], to pseudo-2D Hidden Markov Models (HMM) [33, 48, 158]. All these systems differ in terms of the feature extraction procedure and/or the classification technique used. These systems, as well as others, are described in the sections below.

4.2.1 Geometric Features vs Templates

Brunelli and Poggio [28] compared the performance of a system using automatically extracted geometric features and a classifier based on the Mahalanobis distance [46] (similar to a single-Gaussian GMM) against a system using a template matching strategy. In the former system, the geometrical features included: eyebrow thickness and vertical position at the eye centre position, coarse description of the left eyebrow's arches, vertical position and width of the nose, vertical position of the mouth as well as the width and height, set of radii describing the chin shape, face width at nose position, and the face width halfway between nose tip and eyes. In the latter system, four sub-images (automatically extracted from the frontal face image), representing the eye, nose, mouth and inner face area (from eyebrows downwards), were used by a classifier based on normalised cross correlation with a set of template images. In both systems, the size of the face image was first normalised. Brunelli and Poggio found that the template matching approach obtained superior identification performance and was significantly simpler than the geometric feature based approach. Moreover, they have also found that the face areas can be sorted by discrimination ability as follows:

eyes, nose and mouth; they note that this ordering is consistent with human ability of identifying familiar people from a single facial characteristic.

4.2.2 Principal Component Analysis and Related Techniques

Inspired by the work of Kirby and Sirovich [85], Turk and Pentland [174] proposed the use of Principal Component Analysis (PCA) [117] as a holistic feature extraction method for use in face recognition.

Given a face image matrix $\boldsymbol{F}$ of size $Y \times X$, all the columns of $\boldsymbol{F}$ are concatenated to form a column vector $\boldsymbol{f}$ of dimensionality YX. A D-dimensional feature vector, $\boldsymbol{x}$, is then obtained by:

$$\boldsymbol{x} = \boldsymbol{U}^T(\boldsymbol{f} - \boldsymbol{f}_\mu) \tag{4.1}$$

where matrix $\boldsymbol{U}$ contains D eigenvectors (with largest corresponding eigenvalues) of the training data covariance matrix, and $\boldsymbol{f}_\mu$ is the mean of training face vectors. The eigenvectors are referred to as "eigenfaces" (see Section 4.3.1 for more detail).

On a dataset of 16 people and using an Euclidean distance based classifier, Turk and Pentland obtained 100% identification when using face images obtained in non-challenging conditions. However, the performance decreased when there was a change in the lighting conditions, head size or head orientation. This is not surprising, as $\boldsymbol{x}$ is in effect a dimensionality reduced version of $\boldsymbol{f}$. In addition to sensitivity to lighting conditions, head size or head orientation, the system can also be expected to be sensitive to translations and rotations. Thus prior to holistic feature extraction, it is critical that the face image is normalised (e.g. the location of the eyes must be the same for each person and any illumination changes must be compensated).

Moghaddam and Pentland [115] modified the PCA based face recognition system to use separate face areas (i.e. eyes, nose and mouth) in a similar manner to Brunelli and Poggio [28]. By disregarding the mouth area, Moghaddam and Pentland showed that the system is less affected by expression and other changes to the face (such as a beard). Moreover, an improvement in the identification rate was achieved by combining the holistic PCA system with the modular PCA system. In a separate development in the same paper, the holistic PCA system was modified to use face images processed by an edge detector, resulting in a drop in performance. The edge detector had the effect of removing most of the texture information from the face, indicating that such information is useful in recognition.

Belhumeur et al. [17] investigated the use of Linear Discriminant Analysis (LDA) as a feature extraction technique robust to changes in illumination direction. The training paradigm involved the use of face images with varying illumination. Experiments on two small datasets (the largest having 16 persons) showed that the LDA based approach is significantly more robust than the PCA approach. The experiments also showed that the PCA approach can be made more robust by disregarding the first three eigenfaces, indicating that they are primarily due to lighting variation. However, when the experiment setup was modified to use training images with constant illumination

and testing images with varying illumination, LDA derived features were shown to be still affected, although considerably less than PCA derived features.

4.2.3 Pseudo-2D Hidden Markov Model (HMM) Based Techniques

Samaria [158] extended 1D HMMs (popular in speech recognition [73, 141]) to pseudo-2D HMMs for use in face recognition. A pseudo-2D HMM for each person consists of a pseudo-2D lattice of states, each describing a distribution of feature vectors belonging to a particular area of the face. Samaria used a multivariate Gaussian as a model of the distribution of feature vectors for each state. During testing, an optimal alignment of the states was found for a given image (i.e. the likelihood of each pseudo-2D HMM was maximised). Person identification was achieved by selecting the pseudo-2D HMM which obtained the highest likelihood.

Due to the alignment stage, the pseudo-2D HMM approach is inherently robust to translations, indicating that the face normalisation stage need not be as accurate as for the PCA based approach.

Samaria showed that on a 40 person dataset the pseudo-2D HMM approach outperformed a system comprised of a nearest neighbour classifier and PCA derived feature vectors. The best pseudo-2D HMM approach used 25 states and 96 dimensional feature vectors. The face image was analysed on a block by block basis; the grey level pixel values inside each block were arranged into a feature vector. For the PCA based approach the number of eigenfaces was varied from 5 to 199; the performance generally levelled off when 40 eigenfaces were used.

In related work, Nefian and Hayes [122] proposed to use 2D Discrete Cosine Transform (2D DCT) coefficients [59] rather than the grey level pixel values. Only the coefficients which contained most of the energy were used in forming a feature vector. The same identification rate was achieved as for grey level pixel values, but the classification time was reduced by an order of magnitude.

Eickeler et al. [48] extended the pseudo-2D HMM approach to use 2D DCT coefficients directly from JPEG compressed images [186, 187]; moreover, they have also shown that utilising a three-Gaussian GMM to model for the distribution of feature vectors for each state outperforms a multivariate Gaussian model (i.e. a single-Gaussian GMM).

4.2.4 Elastic Graph Matching (EGM) Based Techniques

Lades et al. [94] proposed to use Elastic Graph Matching (EGM) for face recognition. Each face is represented by a set of feature vectors positioned on the nodes of a coarse 2D grid placed on the face. Each feature vector is comprised of a set of responses of biologically inspired 2D Gabor wavelets [95], differing in orientation and scale (see Section 4.3.2 for more information).

Comparing two faces was accomplished by matching and adapting the grid of a test image (T) to the grid of a reference image (R), where both grids have the same number of nodes; moreover, the test grid has initially the same structure as the reference grid. The elasticity of the test grid

allows accommodation of face distortions (e.g. due to expression changes) and to a lesser extent, changes in the view point. The quality of a match is evaluated using a distance function:

$$d(T,R) = \sum_{i=1}^{N_N} d_f(T_i, R_i) + \xi \sum_{i=1}^{N_N} d_s(T_i, R_i) \tag{4.2}$$

where N_N is the number of nodes, $d_f(T_i, R_i)$ describes the difference between feature vectors representing the i-th node of the test and reference grids, while $d_s(T_i, R_i)$ describes the difference between the spatial distances of node T_i to its neighbouring nodes and the spatial distances of node R_i to its neighbouring nodes. The coefficient ξ controls the stiffness of the test grid, with large values penalising distortion of the test grid with respect to the reference grid (thus $d_s(\cdot,\cdot)$ is used to preserve the topology between the test and reference grids).

$d(T,R)$ is minimised via translation of the test grid and perturbation of the locations of its nodes. Lades et al. proposed an approximate solution to the minimisation problem, comprised of two consecutive stages. First, an approximate match is found by translating the test grid while keeping it rigid [this corresponds to the limit $\xi \to \infty$ in Eqn. (4.2)]. In the second stage, ξ is set to a finite value to permit small grid distortions. Each node of the test grid is visited in a random order and its location is perturbed randomly. Each stage is deemed to have reached convergence once a predefined number of trials has failed to reduce $d(T,R)$. Once convergence is reached, the value of $d(T,R)$ is used for recognition purposes. Lades et al. reported encouraging identification results where test faces contained expression changes and small rotations.

Duc et al. [45] extended the EGM approach to include node specific weighting of the contribution of each Gabor wavelet response to the measure of the difference between feature vectors. On a dataset which had mainly expression changes, the extended system provided lower verification error rates than the standard system. Moreover, Duc et al. showed that the extended system still outperformed the standard system even if the second stage of minimisation of $d(T,R)$ is omitted (i.e. the test grid is kept rigid).

Kotropoulos et al. [91] used the outputs of multiscale morphological dilation and erosion operations [59] to yield a feature vector for each node. Compared to feature vectors based on the responses of Gabor wavelets, the advantage of the morphological operation approach is that it is significantly faster due to its relative simplicity and lack of floating point arithmetic operations. Comparative verification results in [169] show that the morphological operation based approach has slightly lower error rates than the standard approach based on Gabor wavelets.

4.2.5 Other Approaches

Matas et al. [107] proposed a face verification method based on a robust form of correlation. A search for the optimum correlation is performed in the space of all valid geometric and photometric transformations of the test image to obtain the best match with the reference image. The geometric transformations include translations, rotations and scaling, while the photometric transformation corrects the mean of pixel intensity across the face. The quality of the match between a transformed

test image and a reference image is evaluated using a sum of pixel differences, subject to a constraint: if the pixel difference is above a predefined threshold, it is ignored. This constraint is used in order to discount face regions which are subject to relatively large change (such as hair style and expression). The search technique involves the random selection of transformation parameters; each transformation is accepted only if the matching score is increased. To speed up the search, a randomly selected subset of pixels is used instead of the entire image. Verification results on a dataset which had mainly expression changes show a minor improvement over Duc's extended EGM approach (described in Section 4.2.4).

Lawrence et al. [93] proposed the use of a hybrid neural-network approach to face recognition. The system combined local image sampling, a self-organising map (SOM) [89] and a convolutional neural network. On a dataset of 40 people, the proposed approach obtained an identification error rate of 3.8%, compared to 10.5% obtained using a system comprised of the PCA based feature extractor (described Section 4.2.2) and a nearest neighbour classifier. By replacing the features obtained using local image sampling and the SOM with PCA derived features it was shown the improvement in performance can be largely attributed to the convolutional neural network (i.e. the classifier).

4.2.6 Relevant Issues

Zhang et al. [192] compared the performance of the EGM approach with a system comprised of a PCA based feature extractor and a nearest neighbour classifier. Results on a combined dataset of 100 people showed that the PCA based system was more robust to scale and rotation variations, while the EGM approach was more robust to position, illumination and expression variations. Zhang et al. contributed the robustness to illumination changes to the use of Gabor features, while the robustness to position and expression variations was contributed to the deformable matching stage.

Kotropoulos et al. [92] showed that while morphologically derived feature vectors are more sensitive to illumination changes than Gabor wavelet derived features, they are less sensitive to face size variations. They proposed a heuristic size and illumination normalisation technique, which, on a small dataset containing face images collected in real life conditions, was shown to significantly improve the performance of a EGM based system which used the morphologically derived feature vectors. (Strangely, no comparative results were reported for Gabor wavelet derived feature vectors).

Adini et al. [3] studied the suitability of several image processing techniques for reducing the effects of an illumination direction change (where one side of the face was brighter than the other). Various configurations of the following techniques were considered: filtering with 2D Gabor-like filters [95], edge maps, first & second derivatives and log transformations [59]. Several classifiers, based on pixel differences between two processed images, were also evaluated; all of the classifiers produced similar identification results. On a dataset comprised of 25 subjects, Adini et al. found

that none of the processing techniques were sufficient to completely overcome the effects of the illumination direction change; most techniques obtained an identification rate of less than 50%. However, when using unprocessed images, the identification rate was 0%. Adini et al. showed that the 2D Gabor-like filter which emphasised the differences along the vertical axis (e.g. the eyebrows and the eyes) obtained the best results. This is not surprising, considering that the illumination direction change produced the greatest pixel intensity changes along the horizontal axis. Moreover, results obtained using the vertical orientation were mostly independent of the scale of the filter; at other orientations, the size of the filter greatly affected the identification rate. These results indicate that the optimum orientation and scale of the 2D Gabor-like filter is dependent on the direction of the illumination change.

Belhumeur et al. [17] found that the recognition rate is significantly higher when using full faces (that is, containing the hair and the outline of the face) than when using closely cropped faces (that is, containing only the eyebrows, eyes, nose and mouth), indicating that the overall shape of the face is an important feature. However, Belhumeur et al. conjectured that the recognition rate would drop significantly for the full faces if the background or hairstyles were varied; moreover, it may be even lower than for closely cropped faces. Chen et al. [37] suggested that the influence of the closely cropped area on the recognition process is much smaller than that of the outside area (i.e. the hair and the outline of the face). By using synthetic full face images, where the hair and face outline of one person was combined with the closely cropped area from another person, Chen et al. successfully confused a PCA based face recognition system. Along with the results of Moghaddam and Pentland [115] (see Section 4.2.2), these results indicate that for a statistics based face recognition system, the area containing the eyebrows, eyes and the nose is the most useful. The mouth area may need to be disregarded as it is mostly affected by expression changes and beards.

4.3 Feature Extraction for Face Verification

From the review in the preceding section it is evident that PCA derived features, and to a lesser extent, 2D Gabor wavelet derived features, are affected by illumination direction changes. As will be shown, 2D DCT based features are also sensitive to changes in the illumination direction. In this section we introduce four feature sets with the aim of being less affected by illumination direction changes: DCT-delta, DCT-mod, DCT-mod-delta and DCT-mod2. We will show that out of the four, the DCT-mod2 method, which uses polynomial coefficients derived from 2D DCT coefficients of spatially neighbouring blocks, is the most suitable. We then compare the robustness and performance of the DCT-mod2 method against two popular feature extraction techniques, eigenfaces (PCA) and 2D Gabor wavelets, in addition to the standard 2D DCT approach.

We continue as follows. Feature extraction based on PCA, 2D Gabor wavelet and 2D DCT methods are concisely presented in Sections 4.3.1 to 4.3.3. The proposed methods are described in Sections 4.3.4 and 4.3.5. The performance of the described feature extraction techniques is

compared in Sections 4.3.6 and 4.3.7 on the VidTIMIT and Weizmann datasets, respectively. As a side effect of using a GMM based classifier in conjunction with local features (e.g. Gabor wavelets or 2D DCT derived features), the spatial relations between face parts are not used – surprisingly, this still leads to good results. Some properties of GMM based face representation are shown in Section 4.3.8.

To keep consistency with traditional matrix notation, pixel locations (and image sizes) are described using the row(s) first, followed by the column(s).

4.3.1 Eigenfaces (PCA)

Given a face image matrix[1] $\boldsymbol{F}$ of size $Y \times X$, we construct a vector representation by concatenating all the columns of $\boldsymbol{F}$ to form a column vector $\boldsymbol{f}$ of dimensionality YX. Given a set of training vectors $\{\boldsymbol{f_i}\}_{i=1}^{N_P}$ for all persons, we define the mean of the training set as $\boldsymbol{f_\mu}$. A new set of mean subtracted vectors is formed using:

$$\{\ \boldsymbol{g}_i\ \}_{i=1}^{N_P} = \{\ \boldsymbol{f}_i - \boldsymbol{f}_\mu\ \}_{i=1}^{N_P} \tag{4.3}$$

The mean subtracted training set is represented as matrix $\boldsymbol{G} = [\ \boldsymbol{g}_1\ \boldsymbol{g}_2\ ...\ \boldsymbol{g}_{N_P}]$. A scatter matrix is calculated using:

$$\boldsymbol{C} = \boldsymbol{G}\boldsymbol{G}^T \tag{4.4}$$

Due to the size of $\boldsymbol{C}$, calculation of the eigenvectors of $\boldsymbol{C}$ can be computationally infeasible. However, if the number of training vectors (N_P) is less than their dimensionality (YX), there will be only $N_P - 1$ meaningful eigenvectors. Turk and Pentland [174] exploit this fact to determine the eigenvectors using an alternative method, summarised as follows.

Let us denote the eigenvectors of matrix $\boldsymbol{G}^T\boldsymbol{G}$ as $\boldsymbol{v}_j$ with corresponding eigenvalues γ_j:

$$\boldsymbol{G}^T\boldsymbol{G}\boldsymbol{v}_j = \gamma_j\boldsymbol{v}_j \tag{4.5}$$

Pre-multiplying both sides by $\boldsymbol{G}$ gives us:

$$\boldsymbol{G}\boldsymbol{G}^T\boldsymbol{G}\boldsymbol{v}_j = \gamma_j\boldsymbol{G}\boldsymbol{v}_j \tag{4.6}$$

Letting $\boldsymbol{u}_j = \boldsymbol{G}\boldsymbol{v}_j$ and substituting for $\boldsymbol{C}$ from Eqn. (4.4):

$$\boldsymbol{C}\boldsymbol{u}_j = \gamma_j\boldsymbol{u}_j \tag{4.7}$$

Hence the eigenvectors of $\boldsymbol{C}$ can be found by pre-multiplying the eigenvectors of $\boldsymbol{G}^T\boldsymbol{G}$ by $\boldsymbol{G}$. To achieve dimensionality reduction, let us construct matrix $\boldsymbol{U} = [\ \boldsymbol{u}_1\ \boldsymbol{u}_2\ \cdots\ \boldsymbol{u}_D\ ]$, containing D eigenvectors of $\boldsymbol{C}$ with largest corresponding eigenvalues. Here, $D < N_P$. A feature vector $\boldsymbol{x}$ of dimensionality D is then derived from a face vector $\boldsymbol{f}$ using:

[1] The face images used in our experiments have 56 rows (Y) and 64 columns (X).

$$\boldsymbol{x} = \boldsymbol{U}^T(\boldsymbol{f} - \boldsymbol{f}_\mu) \tag{4.8}$$

i.e. face vector $\boldsymbol{f}$ decomposed in terms of D eigenvectors, known as "eigenfaces".

4.3.2 2D Gabor Wavelets

The biologically inspired family of 2D Gabor wavelets can be defined as follows [95]:

$$\Psi(y, x, \omega, \theta) = \frac{\omega}{\kappa\sqrt{2\pi}}\psi_A(y, x, \omega, \theta)\left[\psi_B(y, x, \omega, \theta) - \exp\left\{-\frac{\kappa^2}{2}\right\}\right] \tag{4.9}$$

where

$$\psi_A(y, x, \omega, \theta) = \exp\left\{-\frac{\omega^2}{8\kappa^2}\left[4(y\sin\theta + x\cos\theta)^2 + (y\cos\theta - x\sin\theta)^2\right]\right\} \tag{4.10}$$

and

$$\psi_B(y, x, \omega, \theta) = \exp\left\{i(\omega y\sin\theta + \omega x\cos\theta)\right\} \tag{4.11}$$

Here ω is the radial frequency in radians per unit length and θ is the wavelet orientation in radians. Each wavelet is centred at point $(y, x) = (0, 0)$. The family is made up of wavelets for N_ω radial frequencies, each with N_θ orientations. The radial frequencies are spaced in octave steps and cover a range from $\omega_{min} > 0$ to $\omega_{max} < \pi$, where 2π represents the Nyquist frequency. Typically $\kappa \approx \pi$ so that each wavelet has a frequency bandwidth of one octave [95].

Feature extraction can be accomplished as follows. A coarse rectangular grid is placed over given face image $\boldsymbol{F}$. At each node of the grid, the inner product of $\boldsymbol{F}$ with each member of the family is computed:

$$P_{j,k} = \int_y\int_x \Psi(y_0 - y, x_0 - x, \omega_j, \theta_k)\boldsymbol{F}_{(y,x)}\, dx\, dy \tag{4.12}$$

for $j = 1, 2, \cdots, N_\omega$ and $k = 1, 2, \cdots, N_\theta$. Here, the node is located at (y_0, x_0). An $N_\omega N_\theta$-dimensional feature vector[2] for location (y_0, x_0), is then constructed using the modulus of each inner product [94]:

$$\boldsymbol{x} = \left[\begin{array}{cccccccccc} |P_{1,1}| & |P_{1,2}| & \cdots & |P_{1,N_\omega}| & \cdots & |P_{2,1}| & |P_{2,2}| & \cdots & |P_{2,N_\omega}| & \cdots & |P_{N_\theta,N_\omega}| \end{array}\right]^T \tag{4.13}$$

Thus if there are N_G nodes in the grid, we extract N_G feature vectors from one image.

[2] Typically, $N_\omega = 3$ and $N_\theta = 6$, resulting in an 18 dimensional vector.

4.3.3 2D Discrete Cosine Transform

Here the given face image is analysed on a block by block manner. Given an image block $f(y, x)$, where $y, x = 0, 1, ..., N-1$ (typically $N = 8$), we decompose it in terms of orthogonal 2D DCT basis functions (see Figure 4.1). The result is an $N \times N$ matrix $\boldsymbol{D}$ containing 2D DCT coefficients, with the elements found using:

$$\boldsymbol{D}_{v,u} = \alpha(v)\alpha(u) \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} f(y, x)\beta(y, x, v, u) \quad \text{for} \quad v, u = 0, 1, 2, \cdots, N-1 \tag{4.14}$$

where

$$\alpha(v) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } v = 0 \\ \sqrt{\frac{2}{N}} & \text{for } v = 1, 2, \cdots, N-1 \end{cases} \tag{4.15}$$

and

$$\beta(y, x, v, u) = \cos\left[\frac{(2y+1)v\pi}{2N}\right] \cos\left[\frac{(2x+1)u\pi}{2N}\right] \tag{4.16}$$

The coefficients are ordered according to a zig-zag pattern, reflecting the amount of information stored [59] (see Figure 4.2). The ordering can be approximately described as going from low to high frequency. For a block located at (b, a), the baseline 2D DCT feature vector is composed of:

$$\boldsymbol{d}^{(b,a)} = \left[\begin{array}{cccc} d_0^{(b,a)} & d_1^{(b,a)} & ... & d_{M-1}^{(b,a)} \end{array} \right]^T \tag{4.17}$$

where $d_n^{(b,a)}$ denotes the n-th 2D DCT coefficient and M is the number of retained coefficients[3]. Following [48], each block overlaps its horizontally and vertically neighbouring blocks by 50%. Hence for an image which has Y rows and X columns, there are $N_D = (2\frac{Y}{N} - 1) \times (2\frac{X}{N} - 1)$ blocks[4].

4.3.4 Proposed DCT-delta

In speech based systems, features based on polynomial coefficients (also known as deltas), representing transitional spectral information, have been successfully used to reduce the effects of background noise and channel mismatch [165] (see also Section 3.4.3).

For images, we define the n-th horizontal delta coefficient for block located at (b, a) as a 1st order orthogonal polynomial coefficient:

$$\Delta^h d_n^{(b,a)} = \frac{\sum_{k=-K}^{K} k \; h_k \; d_n^{(b,a+k)}}{\sum_{k=-K}^{K} h_k k^2} \tag{4.18}$$

[3] In our experiments, $M = 15$.

[4] For a 56×64 image, this results in 195 2D DCT feature vectors.

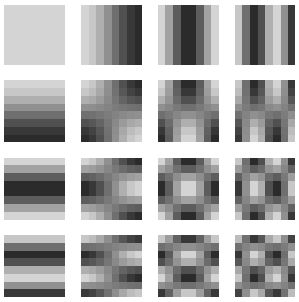

Figure 4.1: Several 2D DCT basis functions for $N = 8$. Lighter colours represent larger values.

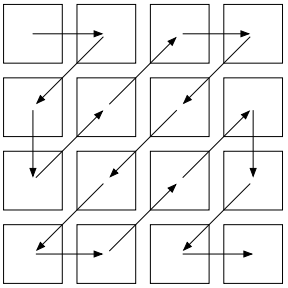

Figure 4.2: Zig-zag ordering of 2D DCT coefficients, $\boldsymbol{D}_{v,u}$, for $N = 4$.

Similarly, we define the n-th vertical delta coefficient as:

$$\Delta^v d_n^{(b,a)} = \frac{\sum_{k=-K}^{K} k\ h_k\ d_n^{(b+k,a)}}{\sum_{k=-K}^{K} h_k k^2} \tag{4.19}$$

where $\boldsymbol{h}$ is a $2K+1$ dimensional symmetric window vector. In this section we shall use $K = 1$ and a rectangular window (i.e. $\boldsymbol{h} = [\ 1\ 1\ 1\ ]^T$).

Let us assume that we have three horizontally consecutive blocks X, Y and Z. Each block is composed of two components: face information and additive noise, e.g. $X = I_X + I_N$. Moreover, let us also suppose that all of the blocks are corrupted with the same noise (a reasonable assumption if the blocks are small and close or overlapping). To find the deltas for block Y, we apply Eqn. (4.18) to obtain (ignoring the denominator):

$$\begin{aligned} \Delta^h Y &= -X + Z && (4.20)\\ &= -(I_X + I_N) + (I_Z + I_N) && (4.21)\\ &= I_Z - I_X && (4.22) \end{aligned}$$

i.e. the noise component is removed.

By combining the horizontal and vertical delta coefficients an overall delta feature vector is formed. Hence, given that we extract M 2D DCT coefficients from each block, the delta vector is $2M$ dimensional. We shall term this feature extraction method as DCT-delta.

DCT-delta feature extraction for a given block is only possible when the block has vertical and horizontal neighbours. A graphical example of the spatial area used by this feature extraction is shown in Figure 4.3. Table 4.1 shows the number of feature vectors extracted from a 56×64 face using $N = 8$ and varying overlap.

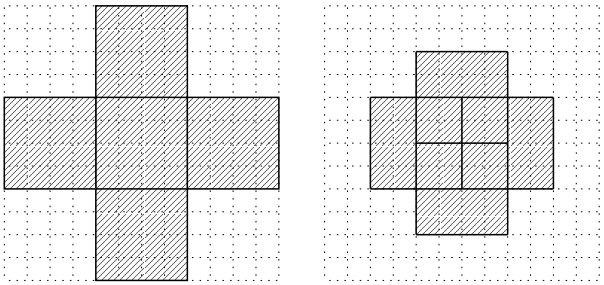

Figure 4.3: Graphical example of the spatial area (shaded) used in DCT-delta feature extraction for $N = 4$. Left: 0% overlap. Right: 50% overlap.

Overlap	Vectors	Spatial width
0	30	24
1	35	22
2	56	20
3	80	18
4	143	16
5	255	14
6	621	12
7	2585	10

Table 4.1: Number of DCT-delta feature vectors extracted from a 56×64 face using $N = 8$ and varying overlap. It also shows the effective spatial width (and height) in pixels for each feature vector. (Note that the effective area used for each feature vector is not equivalent to width × height).

4.3.5 Proposed DCT-mod, DCT-mod2 and DCT-mod-delta

By inspecting Eqns. (4.14) and (4.16), it is evident that the 0-th 2D DCT coefficient will reflect the average pixel value inside each block and hence will be the most affected by any illumination change. Moreover, by inspecting Figure 4.1 it is evident that the first and second coefficients represent the average horizontal and vertical pixel intensity change. As such, they may also be considerably affected by an illumination change. Hence we shall study three additional feature extraction approaches (in all cases we assume the baseline 2D DCT feature vector is M dimensional):

1. Discard the first three coefficients from the baseline 2D DCT feature vector. We shall term this modified feature extraction method as DCT-mod.

2. Discard the first three coefficients from the baseline 2D DCT feature vector and concatenate the resulting vector with the corresponding DCT-delta feature vector. We shall refer to this method as DCT-mod-delta.

3. Replace the first three coefficients with their horizontal and vertical deltas, and form a feature vector representing a given block as follows:

$$\boldsymbol{x} = \Big[\; [\; \Delta^h d_0 \;\; \Delta^v d_0 \;\; \Delta^h d_1 \;\; \Delta^v d_1 \;\; \Delta^h d_2 \;\; \Delta^v d_2 \;] \;\; [\;\; d_3 \;\; d_4 \;\; \cdots \;\; d_{M-1} \;] \; \Big]^T \tag{4.23}$$

   where the (b, a) superscript was omitted for clarity. Let us term this modified approach as DCT-mod2.

Each DCT-mod-delta and DCT-mod2 feature vector represents not only local texture information, but also how a subset of the texture information changes across neighbouring blocks.

4.3.6 Experiments on the VidTIMIT Dataset

Dataset

The VidTIMIT dataset is comprised of video and corresponding audio recordings of 43 people (19 female and 24 male), reciting short sentences selected from the NTIMIT corpus [80]. It was recorded in 3 sessions, with a mean delay of 7 days between Session 1 and 2, and 6 days between Session 2 and 3. There are 10 sentences per person. The first six sentences are assigned to Session 1. The next two sentences are assigned to Session 2 with the remaining two to Session 3. The first two sentences for all persons are the same, with the remaining eight generally different for each person. The mean duration of each sentence is 4.25 seconds, or approximately 106 video frames. Appendix A contains more detailed description of the dataset.

Setup

Before feature extraction can occur, the face must first be located. Furthermore, to account for varying distances to the camera, a geometrical normalisation must be performed. Here we treat the problem of face location and normalisation as separate from feature extraction.

To find the face, we use template matching with several prototype faces[5] of varying dimensions. Using the distance between the eyes as a size measure, an affine transformation is used [59] to adjust the size of the image, resulting in the distance between the eyes to be the same for each person. Finally a 56×64 pixel face matrix, $\boldsymbol{F}$, containing the eyes and the nose (the most invariant face area to changes in the expression and hair style) is extracted from the image.

For PCA, the dimensionality of the face matrix is reduced to 40 (choice based on the works by Kirby and Sirovich [85], Samaria [158] and Belhumeur et al. [17]). For 2D DCT and 2D DCT derived methods, each block is 8×8 pixels. Moreover, each block overlaps with horizontally and vertically adjacent blocks by 50%. For Gabor wavelet features, we heed the choice of Duc et al. [45] with $N_\omega = 3$, $N_\theta = 6$, $\omega_1 = \frac{\pi}{2}$, $\omega_2 = \frac{\pi}{4}$, $\omega_3 = \frac{\pi}{8}$ and $\theta_k = \frac{\pi(k-1)}{N_\theta}$ (where $k = 1, 2, ..., N_\theta$). Hence the dimensionality of the Gabor feature vectors is 18. The location of the wavelet centres was chosen to be as close as possible to the centres of the blocks used in DCT-mod2 feature extraction.

In our experiments we use sequences of images (video) from the VidTIMIT dataset. If the sequence has N_I images, then $N_V = N_I$ for PCA derived feature vectors, $N_V = N_I N_G$ for Gabor features, $N_V = N_I N_D$ for 2D DCT and DCT-mod and $N_V = N_I N_{D2}$ for DCT-delta, DCT-mod-delta and DCT-mod2.

For classification, we use the GMM based classifier described in Chapter 2. To recap, a set of test feature vectors, $X = \{\boldsymbol{x}_i\}_{i=1}^{N_V}$, extracted from a given person is classified as belonging to person K when

[5] A "mother" prototype face was constructed by averaging manually extracted and size normalised faces from all people in the VidTIMIT dataset. Prototype faces of various sizes were constructed by applying an affine transform to the "mother" prototype face.

$$\mathcal{L}(X|\lambda_K) - \mathcal{L}(X|\lambda_{\overline{K}}) \geq t \tag{4.24}$$

where t is the decision threshold, while

$$\mathcal{L}(X|\lambda_K) = \frac{1}{N_V} \log \sum_{i=1}^{N_V} p(\boldsymbol{x}_i|\lambda_K) \tag{4.25}$$

is the average log likelihood of the feature vectors in set X given parameter set λ_K. The GMM parameter set λ_K is optimised through the EM algorithm, using training data for person K. In the experiments reported in this section, the impostor average log likelihood $\mathcal{L}(X|\lambda_{\overline{K}})$ is found using the BMS approach (see Section 2.4.2).

To reduce the computational burden during modelling and testing, every second video frame was used. For each feature extraction method, 8 Gaussian client models[6] (GMMs) were generated from features extracted from face matrices in Session 1. Sessions 2 and 3 were used for testing. Thus for each person an average of 318 frames were used for training and 212 for testing.

Ignoring any edges created by shadows, the main effect of an illumination direction change is that one part of the face is brighter than the rest[7]. Taking this into account, an illumination direction change was introduced to face matrices extracted from Sessions 2 and 3. To simulate more illumination on the left side of the face and less on the right, a new face matrix $\widehat{\boldsymbol{F}}$ is created by transforming $\boldsymbol{F}$ using:

$$\begin{aligned} &\widehat{\boldsymbol{F}}_{(y,x)} = \boldsymbol{F}_{(y,x)} + mx + \delta \\ &\text{for} \quad y = 0, 1, \cdots, N_Y - 1 \quad \text{and} \quad x = 0, 1, \cdots, N_X - 1 \end{aligned} \tag{4.26}$$

where

$$\begin{aligned} m &= \frac{-\delta}{(N_X - 1)/2} \\ \delta &= \text{illumination delta (in pixels)} \end{aligned}$$

Example face matrices for various δ are shown in Figure 4.4. We note that while this model of illumination direction change is artificial as it does not cover all the effects possible in real life (shadows[8], etc.), it should still be useful for providing suggestive results.

To find the performance, Sessions 2 and 3 were used for obtaining example opinions of known impostor and true claims. Four utterances, each from 8 fixed persons (4 male and 4 female), were

[6] Preliminary experiments suggested that there was little performance gain when using more than 8 Gaussians on the VidTIMIT dataset.

[7] As evidenced by the images presented in [91], which were obtained under real-life conditions.

[8] However, the face images presented in [17] show that only extreme illumination direction conditions produce significant shadows, where even humans are likely to have trouble recognising faces.

Figure 4.4: Examples of the artificial illumination change. Left: $\delta = 0$ (no change); middle: $\delta = 40$; right: $\delta = 80$.

used for simulating impostor accesses against the remaining 35 persons. As in [146], 10 background person models were used for the impostor likelihood calculation. For each of the remaining 35 persons, their four utterances were used separately as true claims. In total there were 1120 impostor and 140 true claims. The decision threshold was then set so the a-posteriori performance is as close as possible to the EER.

Evaluation and Discussion

In the first experiment, we found the performance of the 2D DCT approach on face matrices with $\delta = 0$ (i.e. no illumination change) while varying the dimensionality of the feature vectors. The results are presented in Figure 4.5. As can be observed, the performance improves immensely as the number of dimensions is increased from 1 to 3. Increasing the dimensionality from 15 to 21 provides only a relatively small improvement, while increasing the amount of computation time required to generate the models. Based on this we have chosen 15 as the dimensionality of baseline 2D DCT feature vectors. Hence the dimensionality of DCT-delta feature vectors is 30, DCT-mod is 12, DCT-mod-delta is 42 and DCT-mod2 is 18.

In the second experiment we compared the performance of 2D DCT and all of the proposed techniques for increasing δ. Results are shown in Figure 4.6.

In the third experiment we compared the performance of PCA, PCA with histogram equalisation pre-processing[9], DCT, Gabor and DCT-mod2 features for varying δ. Results are presented in Figure 4.7.

In the fourth experiment, we have evaluated the effects of varying block overlap used during DCT-mod2 feature extraction (in all other experiments, the overlap was fixed at 50%). Varying the overlap has two effects: the first is that as overlap is increased the spatial area used to derive one feature vector is decreased; the second effect is that the number of feature vectors extracted from an image grows quickly as the overlap is increased (see Table 4.1). Results are shown in Figure 4.8.

As can be observed in Figure 4.5, the first three 2D DCT coefficients contain a significant amount of person dependent information. Hence ignoring them (as in DCT-mod) implies a reduction in

[9]Histogram equalisation [34, 59] is often used in an attempt to reduce the effects of varying illumination conditions [88, 118].

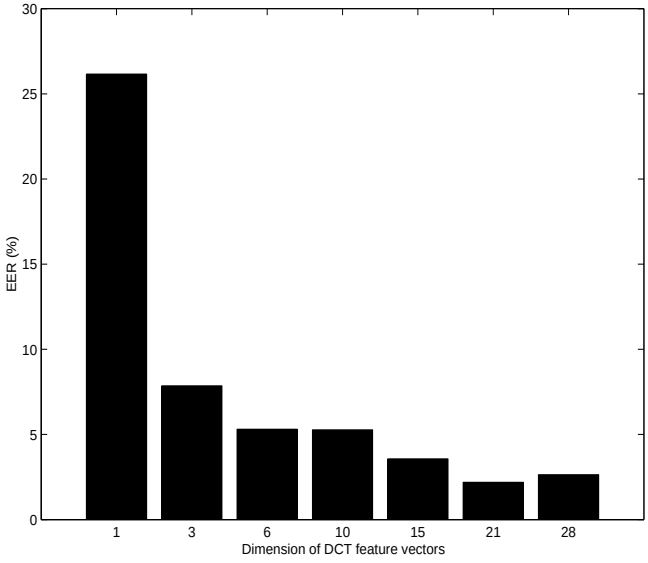

Figure 4.5: EER for increasing dimensionality of 2D DCT feature vectors.

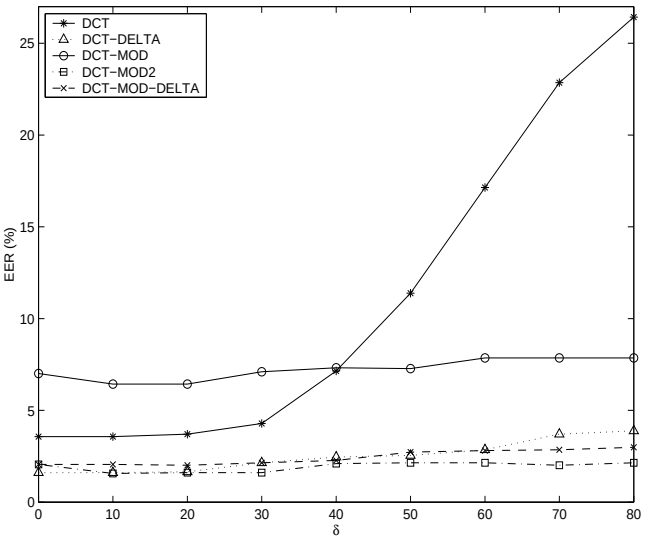

Figure 4.6: EER of 2D DCT and proposed feature sets for increasing illumination change.

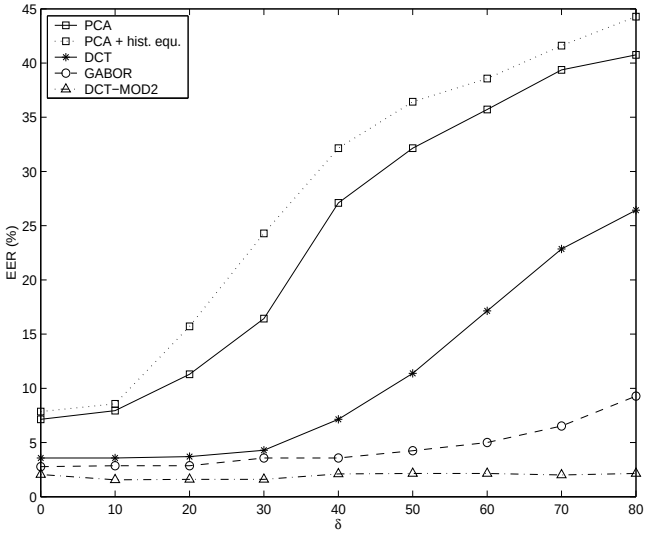

Figure 4.7: EER for PCA, PCA with histogram equalisation pre-processing, DCT, Gabor and DCT-mod2 feature sets.

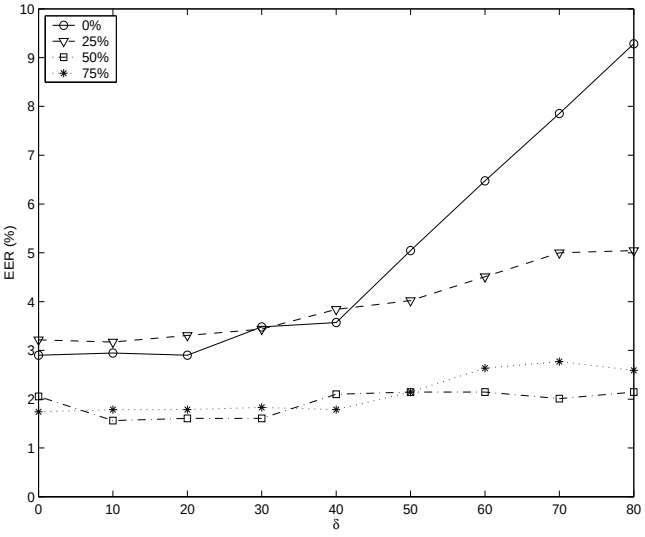

Figure 4.8: EER for DCT-mod2 for varying overlap.

performance. This is verified in Figure 4.6 where the DCT-mod features have worse performance than 2D DCT features when there is little or no illumination direction change ($\delta \leq 30$). We can also see that the performance of DCT features is fairly stable for small illumination direction changes but rapidly degrades for $\delta \geq 40$. This is in contrast to DCT-mod features which have a relatively static performance.

The remaining feature sets (DCT-delta, DCT-mod-delta and DCT-mod2) do not have the performance penalty associated with the DCT-mod feature set. Moreover, all of them have similarly better performance than 2D DCT features. We conjecture that the increase in performance can be attributed to the effectively larger spatial area used when obtaining the features. DCT-mod2 edges out DCT-delta and DCT-mod-delta in terms of stability for large illumination direction changes ($\delta \geq 50$). Additionally, the dimensionality of DCT-mod2 (18) is lower than DCT-delta (30) and DCT-mod-delta (42).

The results suggest that delta features make the system more robust as well as improve performance. The results also suggest that it is only necessary to use deltas of coefficients representing the average pixel intensity and low frequency features (i.e. the 0-th, first and second 2D DCT coefficients) while keeping the remaining DCT coefficients unchanged. Hence out of the four proposed feature extraction techniques, the DCT-mod2 approach is the most suitable.

Using 0% or 25% block overlap in DCT-mod2 feature extraction (Figure 4.8) results in a performance degradation as δ is increased, implying that the assumption that the blocks are corrupted with the same noise has been violated (see Section 4.3.4). Increasing the overlap from 50% to 75% had little effect on the performance at the expense of extracting considerably more feature vectors.

By comparing the performance of PCA, PCA with histogram equalisation pre-processing, 2D DCT, 2D Gabor and DCT-mod2 feature sets (Figure 4.7), it can be seen that the DCT-mod2 approach is the most immune to illumination direction changes (the performance is virtually flat for varying δ). The performance of PCA derived features rapidly degrades as δ increases, while the performance of 2D Gabor features is stable for $\delta \leq 40$ and then gently deteriorates as δ increases. We can also see that use of histogram equalisation as pre-processing for PCA increases the error rate in all cases, and most notably offers no help against illumination changes.

We note that using the introduced illumination change, the centre portion of the face (column wise) is largely unaffected; the size of the portion decreases as δ increases. In the PCA approach one feature vector describes the entire face, hence any change to the face would alter the features obtained. This is in contrast to the other approaches (2D Gabor, 2D DCT and DCT-mod2), where one feature vector describes only a small part of the face. Thus a significant percentage (dependent on δ) of the feature vectors is largely unchanged, automatically leading to a degree of robustness.

Method	Illumination direction		
	uniform	left	right
DCT	3.49	48.15	48.15
Gabor	0.36	33.34	33.34
DCT-mod2	0	25.93	22.65

Table 4.2: Results on the Weizmann Dataset, quoted in terms of approximate EER (%).

4.3.7 Experiments on the Weizmann Dataset

The experiments described in Section 4.3.6 used an artificial illumination direction change. In this Section we compare the performance of 2D DCT, 2D Gabor wavelet and DCT-mod2 feature sets on the Weizmann dataset [3], which has more realistic illumination direction changes.

The Weizmann dataset is rather small, as it is comprised of images of 27 people; moreover, for the direct frontal view, there is only one image per person with uniform illumination (the training image) and two test images where the illumination is either from the left or right; all three images were taken in the same session. As such, the dataset is not the best for verification experiments, but some suggestive results can still be obtained.

The experiment setup is similar to that described in Section 4.3.6. However, due to the small amount of training data, an alternative GMM training strategy is used. Rather than training the client models directly using the EM algorithm, each model is derived from a Universal Background Model (UBM) through MAP adaptation (see Section 2.4.2). The UBM is trained via the EM algorithm using pooled training data from all clients. Moreover, due to the small number of persons in the dataset, the UBM is also used to calculate the impostor likelihood (rather than using a set of background models). A detailed description of this training and testing strategy is presented in Section 2.4.2.

For each illumination type, the client's own training image was used to simulate a true claim. Images from all other people were used to simulate impostor claims. In total, for each illumination type, there were 27 true claims and 702 impostor claims. The decision threshold was set to obtain performance as close as possible to the EER. In the results shown in Table 4.2 it can be observed that no method is immune to the changes in the illumination direction. However DCT-mod2 features appear to be the least affected, followed by Gabor features and DCT features.

4.3.8 Face Areas Modelled by the GMM

When using the GMM classifier in conjunction with local features such as Gabor wavelets or 2D DCT derived features, the spatial relations between major face features (e.g. eyes and nose) are in effect lost – this is due to the assumption of independence of each feature vector in Eqn. (4.25). However, good performance is still obtained, implying that the use of more complex classifiers which explicitly preserve spatial relations, such as EGM and pseudo-2D HMM, may not be strictly

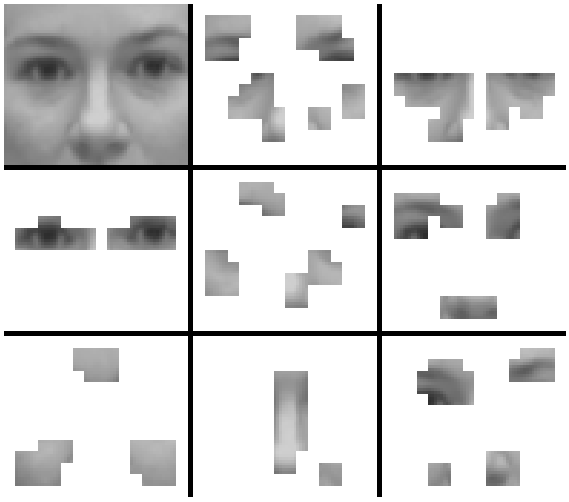

Figure 4.9: An example of 8 Gaussian GMM face modelling. Top left: original image of subject fdrd1. Other squares: areas modelled by each Gaussian in fdrd1's model (DCT-mod2 feature extraction).

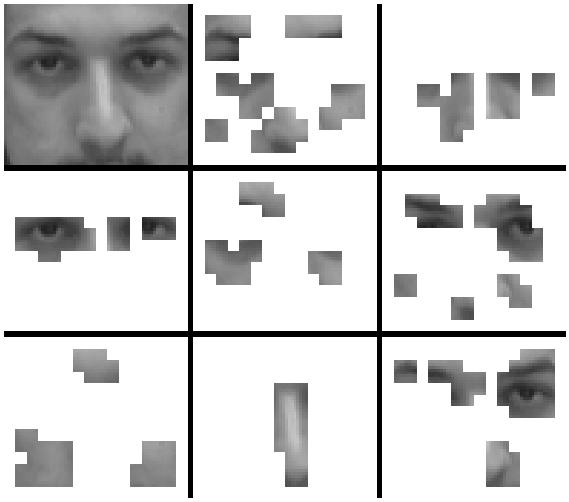

Figure 4.10: Top left: original image of subject mbdg0. Other squares: areas selected by fdrd1's Gaussians.

necessary. Moreover, due to the loss of the spatial relations, the GMM classifier theoretically has inbuilt robustness to translations[10] (which may be caused by inaccurate face localisation).

An example of the face areas modelled by each Gaussian (in an 8 Gaussian GMM) is shown in Figure 4.9, where DCT-mod2 feature extraction was used. Images from a video sequence of the face were used to train the model. The selected areas represent the centre blocks used in DCT-mod2 feature extraction (see Section 4.3.5). Some overlap between the areas for different Gaussians is present since a 50% block overlap was used.

As can be observed, the type of area modelled by each Gaussian is generally guided by the degree of smoothness of the area. This can lead to automatic selection of physically meaningful areas, such as the eyes and the nose. This is expected, since the EM algorithm used to train each GMM (see Section 2.4.1) is in effect a probabilistic clustering procedure, where similar features are represented by each Gaussian.

Figure 4.10 shows a typical example of the effect of decomposing a face image in terms of a different person's model. In this case, fdrd1's model was used to decompose mbdg0's face image. By comparing Figures 4.9 and 4.10 it can be seen that fdrd1's model selects similar areas in fdrd1's and mbdg0's face images. Hence if we assume that, in a verification scenario, subject mbdg0 claims to be subject fdrd1, the GMM-based face verification system, in effect, compares fdrd1's eyes against mbdg0's eyes.

[10] Empirical evaluations presented in [152] support this claim.

4.4 Enhanced PCA

As described in Section 4.3.6, the main effect of an illumination direction change is that one part of the face is brighter than the rest. Since the pixel intensity for that part is larger than usual, the dot product obtained by projecting the face onto an eigenface [see Eqn. (4.8)] is now different from the usual result. Because of this, use of PCA derived features results in poor performance under varying illumination conditions.

We propose to address the fragility of PCA derived features to illumination changes by introducing a pre-processing step. A given face image is first processed using DCT-mod2 feature extraction to produce a pseudo-image $\widetilde{\boldsymbol{F}}$, which is then used in place of $\boldsymbol{F}$ by traditional PCA feature extraction (described in Section 4.3.1). Since the DCT-mod2 feature vectors can be robust to illumination changes, features obtained through this method also have the potential to be robust. We shall refer to this method as enhanced PCA (EPCA). This approach differs to that of Belhumeur et al. [17] where training images in varying illumination conditions are required. It also differs from using an edge detector as the pre-processor (as used by Moghaddam and Pentland [115], resulting in a drop in performance) since local texture information is retained.

Formally, a given image is analysed on a block by block basis, with the blocks overlapping by 50%. Each block has N rows and N columns, where $N = 8$. For a 50% overlap, processing an image which has Y rows and X columns with DCT-mod2 feature extraction results in $(2\frac{Y}{N} - 3) \times (2\frac{X}{N} - 3)$ vectors (due to the feature extraction being only possible for blocks which have vertical and horizontal neighbours). Hence for a 56×64 image, there are 11×13 feature vectors. Let us now construct the pseudo image:

$$\widetilde{\boldsymbol{F}} = \begin{bmatrix} \boldsymbol{d}^{(1\Delta b,\ 1\Delta a)} & \boldsymbol{d}^{(1\Delta b,\ 2\Delta a)} & \boldsymbol{d}^{(1\Delta b,\ 3\Delta a)} & \dots \\ \boldsymbol{d}^{(2\Delta b,\ 1\Delta a)} & \boldsymbol{d}^{(2\Delta b,\ 2\Delta a)} & \boldsymbol{d}^{(2\Delta b,\ 3\Delta a)} & \dots \\ \boldsymbol{d}^{(3\Delta b,\ 1\Delta a)} & \boldsymbol{d}^{(3\Delta b,\ 2\Delta a)} & \boldsymbol{d}^{(3\Delta b,\ 3\Delta a)} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{4.27}$$

where $\boldsymbol{d}^{(n\Delta b, n\Delta a)}$ denotes the DCT-mod2 feature vector for block located at $(n\Delta b, n\Delta a)$, while Δb and Δa are block location advancement constants for rows and columns respectively. Since $N = 8$ and we are using a 50% overlap, Δb and Δa are equal to 4. As each DCT-mod2 feature vector is $M + 3$ dimensional, matrix $\widetilde{\boldsymbol{F}}$ has $(M + 3)(2\frac{Y}{N} - 3)$ rows and $(2\frac{X}{N} - 3)$ columns.

4.4.1 Experiments and Discussion

Experiments are done with four types of images: (i) high quality, (ii) corrupted with an illumination direction change, (iii) corrupted with compression artefacts, (iv) corrupted with white Gaussian noise. While the illumination direction change may be of most concern in security systems, in forensic applications [98] all three types of image corruption can be important. Here, face images may be obtained in various illumination conditions from various sources: digitally stored video,

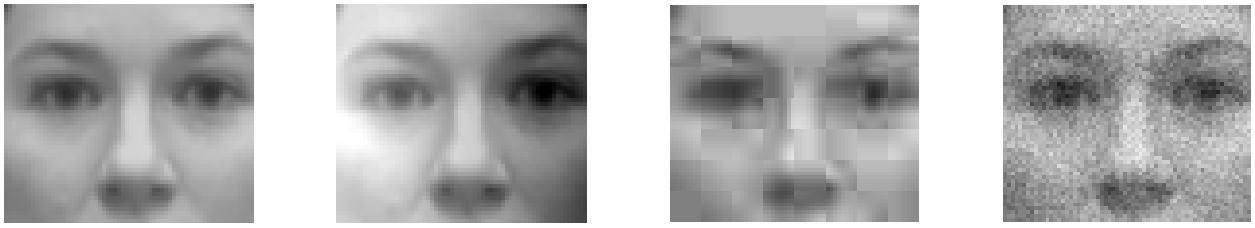

Figure 4.11: From left to right: original image, corrupted with the artificial illumination change (δ=80), corrupted with compression artefacts (PSNR=31.7 dB), corrupted with white Gaussian noise (PSNR=26 dB).

possibly damaged and/or low quality analogue video tape or a TV signal corrupted with "static" noise.

The experiment setup is similar to that of Section 4.3.6. The changes are as follows. For experiments involving compression artefacts, face matrices extracted from Sessions 2 and 3 were processed by a JPEG codec [186, 187] (simulating compressed digital video). The JPEG codec reduces the bitrate of a given image at the expense of introducing distortion in the form of compression artefacts. The distortion is measured in terms of Peak Signal to Noise Ratio[11] (PSNR). The average PSNR of the corrupted face matrices ranged from 45.66 to 31.13 dB. In a similar manner, for TV "static" noise experiments, face matrices extracted from Sessions 2 and 3 were corrupted by additive white Gaussian noise, with the PSNR ranging from 40 to 15.5 dB. Example face matrices are shown in Figure 4.11. For PCA and EPCA based systems, MAP based training was used.

In the first experiment we compared the performance of EPCA derived features with that of the traditional PCA derived features and DCT-mod2 features, using faces corrupted by the illumination change. Results, shown in Figure 4.12, suggest that EPCA derived features are largely immune to the illumination change and on clean data obtain the same performance as traditional PCA based features.

The second experiment was similar to the first one, with the change that the faces were corrupted by the JPEG codec. Results are shown in Figure 4.13. Both PCA and EPCA based systems are virtually unaffected by the compression artefacts and obtain almost exactly the same performance. DCT-mod2 features have relatively stable performance upto a PSNR of 35.89 dB. Their performance then rapidly degrades as the PSNR is lowered, becoming worse than the PCA and EPCA approaches at a PSNR of 33.3 dB.

In the third experiment the faces were corrupted by additive white Gaussian noise (simulating TV "static" noise). The results are presented in Figure 4.14. Once again, both PCA approaches are virtually immune and obtain very similar performance. Performance of DCT-mod2 features quickly degrades as the PSNR is lowered – it becomes worse than PCA and EPCA at a PSNR of

[11] We use the following definition of the Peak Signal to Noise Ratio, expressed in decibels (dB): $\text{PSNR} = 10\log_{10}(peakval^2/(\frac{1}{YX}\sum_{i,j}(\boldsymbol{S}_{(i,j)} - \boldsymbol{N}_{(i,j)})^2))$, where $peakval = 255$ (for 8 bit images), $\boldsymbol{S}$ is the original image and $\boldsymbol{N}$ is the corrupted image. Both images have a size of Y rows and X columns.

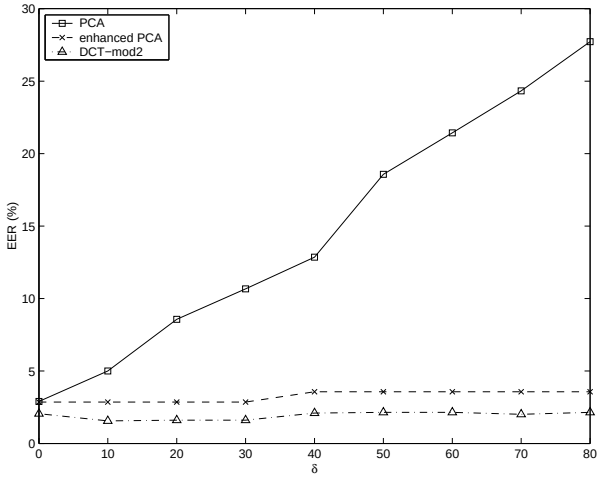

Figure 4.12: EER for faces corrupted with the artificial illumination change, using PCA, Enhanced PCA (EPCA), and DCT-mod2 based approaches.

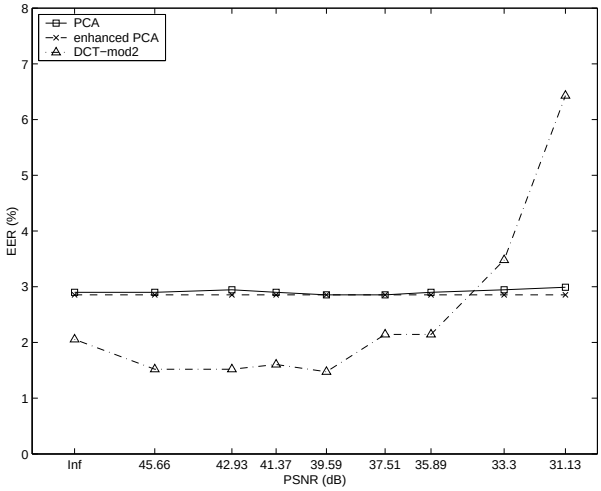

Figure 4.13: As per Figure 4.12, but for faces corrupted with compression artefacts.

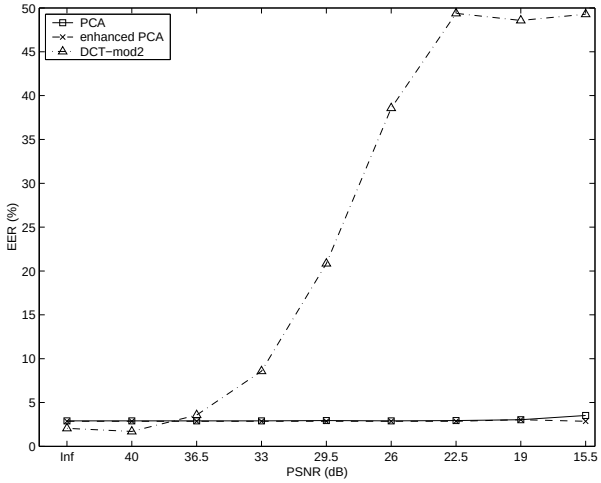

Figure 4.14: As per Figure 4.12, but for faces corrupted with white Gaussian noise.

36.5 dB and becomes unusable at a PSNR of 22.5 dB.

We have observed that while the additive noise greatly distorts the image, the average pixel intensity for the whole face remains largely the same (similar reasoning can be applied for the case of images corrupted with compression artefacts). Hence we conjecture that the robustness of PCA and EPCA approaches stems from extracting information from the entire face in one hit (i.e. holistic feature extraction). This is in contrast to the DCT-mod2 based system, which is a type of a local feature system. Here each feature vector describes only a small section of the face and hence is easily affected by additive noise.

4.5 Summary and Future Directions

In this chapter we first overviewed important approaches in the field of face recognition: Geometric features, templates, Principal Component Analysis (PCA), pseudo-2D Hidden Markov Models (HMM), Elastic Graph Matching (EGM), as well as other points were covered. Important issues, such as the effects of an illumination direction change and the use of different face areas, were also covered.

A new feature set (termed DCT-mod2) was proposed; it uses polynomial coefficients derived from 2D DCT coefficients of spatially neighbouring blocks. Its robustness and performance was evaluated against three popular feature sets for use in an identity verification system subject to illumination direction changes. Results on the multi-session VidTIMIT dataset suggest that the proposed feature set is considerably more robust than standard PCA, standard 2D DCT and somewhat more robust than Gabor wavelets.

We proposed to address the fragility of PCA derived features to illumination direction changes by introducing a pre-processing step which involves applying the DCT-mod2 feature extraction to the original face image. A pseudo-image is then constructed by placing all DCT-mod2 feature vectors in a matrix on which traditional PCA feature extraction is performed. It was shown that the enhanced PCA technique retains all the positive aspects of traditional PCA (that is, robustness to compression artefacts and white noise), while also being robust to changes in the illumination direction.

The local feature system (comprised of DCT based feature extraction and a GMM based classifier) used in this chapter has a number of advantages and disadvantages. While it has been shown to be relatively robust to the effects of imperfect face localisation, such as translations [33, 152] (as well as being robust to pose changes, as shown in the following chapter), it may have issues with scalability. In an identification scenario (e.g. for surveillance purposes), a given face may need to be compared to a large number of reference faces (i.e. a watch list). If each reference face is represented by a GMM, the number of evaluations of the exponential function in Eqn. (2.23) [via Eqn. (4.25)] may quickly become prohibitive, especially if a large number of feature vectors is extracted or a large number of components is used in each reference GMM.

To address the scalability issue, an alternative approach based on the so-called "bag of words" technique [40, 127, 164] could be used. In this technique, a given face would be described as a histogram of "visual words" occurrences. In particular, each face part (e.g. a block) would be represented in terms of a large number of pre-set face parts ("words"). The representation mechanism can be through Vector Quantisation [56], where each face part is assigned to exactly one word, or through a GMM, where each face part is assigned probabilities of belonging to a number of words. In the case of the GMM representation, the dictionary of pre-set face parts would in effect be the UBM.

The "bag of words" technique would be scalable as a given face would only need to be processed through one GMM, instead of a large number of reference GMMs. Preliminary work in this direction shows promising results [159]. Furthermore, there might be additional benefits: the histogram of "visual word" occurrences can be classified by a kernel based technique such as the Support Vector Machine (SVM) [46, 188]. Due to regularisation as well as the margin constraint, SVMs may have the advantage of being able to use (in effect) only the most stable and/or discriminative features (where, in our case, the features would be face parts). A demonstration of this ability is shown in [16, 63], in the context of cancer classification.

Chapter 5

Verification using Faces with Pose Variations

5.1 Overview

In contrast to the previous chapter where we dealt with frontal faces, in this chapter we deal with faces subject to pose variations. Specifically, we address the pose mismatch problem that occurs when there is only a single (frontal) face image available for training and non-frontal faces are presented during testing.

The problem is tackled through building multi-angle models, by extending each frontal face model with artificially synthesised models for non-frontal views. The synthesis methods are based on several implementations of Maximum Likelihood Linear Regression (MLLR), as well as standard multi-variate linear regression (LinReg). All synthesis techniques rely on prior information and learn how face models for the frontal view are related to face models for non-frontal views. We stress that instead of synthesising images, model parameters are synthesised.

The synthesis and extension approach is evaluated by applying it to two face verification systems: a holistic system (based on PCA-derived features) and a local feature system (based on DCT-derived features). Experiments on the FERET dataset suggest that for the holistic system, the LinReg based technique is more suited than the MLLR based techniques. For the local feature system, the results show that synthesis via a new MLLR implementation obtains better performance than synthesis based on traditional MLLR. The results further suggest that multi-angle models are able to considerably reduce errors due to pose variations.

It is also shown that the local feature system is less affected by view changes than the holistic system. This can be attributed to the parts based representation of the face, and, due to the classifier based on mixtures of Gaussians, the lack of constraints on spatial relations between the face parts, allowing for deformations and movements of face areas.

5.2 Introduction

Many contemporary approaches to face recognition are able to achieve low error rates when dealing with frontal faces (see for example [39, 111]). In order to handle non-frontal faces, previously proposed extensions to 2D approaches include the use of training images (for the person to be recognised) at multiple views [60, 61, 133]. In some applications, such as surveillance, there may be only one reference image (e.g. a passport photograph) for the person to be spotted. In a surveillance video (e.g. at an airport), the pose of the face is usually uncontrolled, thus causing a problem in the form of a mismatch between the training and the test poses.

While true 3D based approaches in theory allow face matching at various poses, current 3D sensing hardware has too many limitations [26], including cost and range. Moreover unlike 2D recognition, 3D technology cannot be retrofitted to existing surveillance systems. Quasi-3D approaches [10, 27] (where the 3D shape is inferred from 2D images) can also be used to deal with pose variations, though we do not pursue this direction here.

In this chapter we concentrate on extending two 2D based techniques. Specifically, we extend the local feature approach presented in Chapter 4 (based on DCT-derived features) and a holistic approach (based on PCA-derived features [85, 174]). In both cases we employ a Bayesian classifier based on Gaussian Mixture Models (GMMs) [46, 149], which is central to our extensions. We exclusively deal with the classification problem, and postulate that the face localisation step has been performed correctly.

The PCA/GMM system is an extreme example of a holistic system where the spatial relations between face characteristics (such as the eyes and nose) are rigidly kept. Contrarily, the DCT/GMM approach is an extreme example of a local feature approach (also known as a parts based approach [99]). Here, the spatial relations between face parts are largely not used, resulting in robustness to translations of the face which can be caused by an automatic face localisation algorithm [33, 152]. In between the two extremes are systems based on multiple template matching [28], modular PCA [133], Pseudo 2D Hidden Markov Models [33, 48, 158] and approaches based on Elastic Graph Matching [45, 94].

We propose to address the single training pose problem by extending each statistical frontal face model with artificially synthesised models for non-frontal views. We propose to synthesise the non-frontal models via methods based on several implementations of Maximum Likelihood Linear Regression (MLLR), as well as standard multi-variate linear regression (LinReg).

In the proposed MLLR-based approach, prior information is used to construct generic face models for different views. A generic GMM does not represent a specific person's face – instead it represents a population of faces, or interpreted alternatively, a "generic" face. In the field of speech based identity verification, an analogous generic model is known as a world model and as a Universal Background Model [104, 149] (see also Section 2.4.2). Each non-frontal generic model is constructed by learning and applying a MLLR-based transformation to the frontal generic model. When we wish to obtain a person's non-frontal model, we first obtain the person's frontal model via

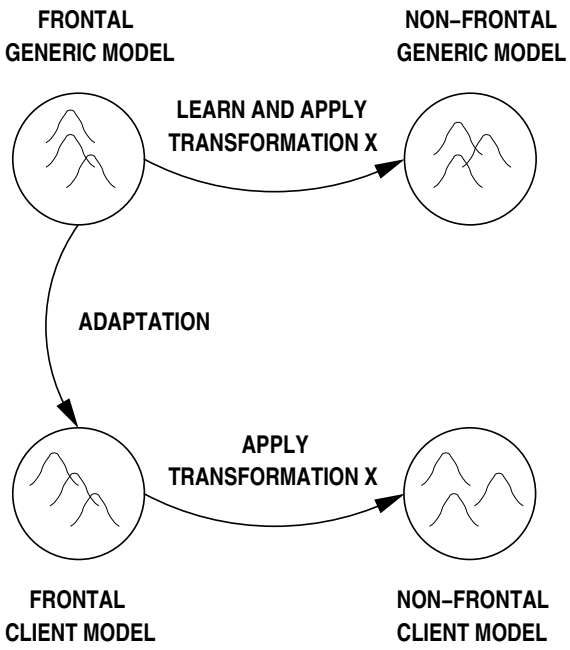

Figure 5.1: An interpretation of synthesising a non-frontal client model based on how the frontal generic model is transformed to a non-frontal generic model.

adapting [149] the frontal generic model. A non-frontal face model is then synthesised by applying the previously learned transformation to the person's frontal model. In order for the system to automatically handle the two views, a person's frontal model is extended by concatenating it with the newly synthesised model (i.e. we're building a multi-angle model). The procedure is then repeated for other views. An interpretation of this procedure is shown in Figure 5.1.

The LinReg approach is similar to the MLLR-based approach described above. The main difference is that it learns a common relation between two sets of feature vectors, instead of learning the transformation between generic models. In our case the LinReg technique is applicable only to the holistic system, while the MLLR-based methods are applicable to both holistic and local feature based systems.

The chapter is continued as follows. An overview of related work is given in Section 5.3. In Section 5.4 we briefly describe the dataset used in the experiments and the pre-processing of images. An overview the DCT- and PCA-based feature extraction techniques is given in Section 5.5. Section 5.6 provides a concise description of the GMM based classifier and the different training strategies used when dealing with DCT and PCA derived features. In Section 5.7 we summarise MLLR, while in Section 5.8 we describe model synthesis techniques based on MLLR and standard multi-variate linear regression. Section 5.9 details the process of extending a frontal model with synthesised non-frontal models. Section 5.10 is devoted to experiments evaluating the proposed

synthesis techniques and the use of extended models. The main findings and future directions are presented in Section 5.11.

5.3 Related Work

Previous approaches to addressing single view problems include the synthesis of new images at previously unseen views. Some examples are optical flow based methods [23, 126], linear object classes [180] and methods based on Active Appearance Models [161]. To handle views for which there are no training images, an appearance based face recognition system could then use the synthesised images. The proposed model synthesis and extension approach is inherently more efficient, as the intermediary steps of image synthesis and feature extraction (from synthesised images) are omitted.

The model extension part of the proposed approach is somewhat similar to [61], where features from many real images were used to extend a person's face model. This is in contrast to the proposed approach, where the models are synthesised to represent the face of a person for various non-frontal views, without having access to the person's real images. The synthesis part is somewhat related to [108] where the "jets" in the nodes an elastic graph are transformed according to a geometric framework. Apart from the inherent differences in the structure of classifiers (i.e. Elastic Graph Matching compared to a Bayesian classifier), the proposed synthesis approach differs in that it is based on a statistical framework.

5.4 Dataset Setup and Pre-Processing

In our experiments we used a subset of face images from the FERET dataset [134], rather than the VidTIMIT dataset. While VidTIMIT has images with pose variations, the FERET dataset has more subjects and can be considered to be a de-facto dataset for experiments on pose variations. We used images from the ba, bb, bc, bd, be, bf, bg, bh and bi portions, which represent views of 200 persons for approximately 0^o (frontal), $+60^o$, $+40^o$, $+25^o$, $+15^o$, -15^o, -25^o, -40^o and -60^o, respectively.

The 200 persons were split into three groups: group A, group B and an impostor group[1]. There are 90 people each in group A and B, and 20 people in the impostor group. Example images are shown in Figure 5.2. Throughout the experiments, group A is used as a source of prior information while the impostor group and group B are used for verification tests. For most experiments there are

[1]FERET subject IDs for group A: 00019, 00029, 00268, 00647, 00700, 00761, 01013 to 01018, 01020 to 01032, 01034 to 01048, 01050, 01052, 01054 to 01066, 01068 to 01076, 01078 to 01081, 01083, 01084, 01085, 01086, 01088 to 01092, 01094, 01098, 01101, 01103, 01106, 01108, 01111, 01117, 01124, 01125, 01156, 01162, 01172. Group B: 01095 to 01097, 01099, 01100, 01102, 01104, 01105, 01107, 01109, 01110, 01112 to 01116, 01118 to 01120, 01122, 01127 to 01136, 01138 to 01142, 01144, 01146 to 01150, 01152 to 01155, 01157 to 01161, 01163 to 01168, 01170, 01171, 01173 to 01178, 01180 to 01202, 01204 to 01206. Impostor group: 01019, 01033, 01049, 01051, 01053, 01067, 01077, 01082, 01087, 01093, 01121, 01123, 01126, 01137, 01143, 01145, 01151, 01169, 01179, 01203.

Figure 5.2: Example images from the FERET dataset for 0^o (frontal), $+15^o$, $+25^o$, $+40^o$ and $+60^o$ views. The angles are approximate.

Figure 5.3: Extracted face windows from images in Figure 5.2.

90 true claimant accesses and 90×20=1800 impostor attacks per angle (with the view of impostor faces matching the testing view). This restriction is relaxed in later experiments.

To reduce the effects of facial expressions and hair styles, closely cropped faces are used [37]. Face windows, with a size of 56 rows and 64 columns, are extracted based on manually found eye locations. As we are proposing extensions to existing 2D approaches, we obtain normalised face windows for non-frontal views in the same way as for the frontal view (i.e. the location of the eyes is the same in each face window). This has a significant side effect: for large deviations from the frontal view (such as -60^o and $+60^o$) the effective size of facial characteristics is significantly larger than for the frontal view. The non-frontal face windows hence differ from the frontal face windows not only due to out-of-plane rotations, but also scale. Example face windows are shown in Figure 5.3.

5.5 Feature Extraction

For convenience, in this section DCT- and PCA-based feature extraction techniques are overviewed below. It must be emphasised that in the PCA based approach, one feature vector represents the entire face (i.e. it is a holistic representation), while in the DCT approach one feature vector represents only a small portion of the face (i.e. it is a local feature representation).

5.5.1 DCT Based System

We use the DCT-mod2 feature extraction technique (described in detail in Chapter 4), which is a modified form of DCT based feature extraction. A given face image is first analysed on a block by block basis. Each block is $N_P \times N_P$ (here we use N_P=8) and overlaps neighbouring blocks by N_O pixels. Each block is decomposed in terms of orthogonal 2D Discrete Cosine Transform (DCT) basis functions [59]. A feature vector for a given block is then constructed as:

$$\boldsymbol{x} = \Big[\; [\; \Delta^h d_0 \;\; \Delta^v d_0 \;\; \Delta^h d_1 \;\; \Delta^v d_1 \;\; \Delta^h d_2 \;\; \Delta^v d_2 \;] \;\; [\; d_3 \;\; d_4 \;\; \cdots \;\; d_{M-1} \;] \; \Big]^T \tag{5.1}$$

where d_n represents the n-th DCT coefficient, while $\Delta^h d_n$ and $\Delta^v d_n$ represent the horizontal and vertical delta coefficients respectively. The deltas are computed using DCT coefficients extracted from neighbouring blocks. Compared to standard DCT feature extraction [48], the first three DCT coefficients are replaced by their respective horizontal and vertical deltas as a way of preserving discriminative information while alleviating the effects of illumination changes. Note that this feature extraction is only possible when a given block has vertical and horizontal neighbours. In this study we use $M = 15$ (choice based on the results presented in Chapter 4), resulting in an 18 dimensional feature vector for each block.

The degree of overlap (N_O) has two effects: (i) as overlap is increased the spatial area used to derive one feature vector is decreased (see Figure 4.3 for an example); (ii) as the overlap is increased the number of feature vectors extracted from an image grows quickly (see Table 4.1). As will be shown later, the larger the overlap (and hence the smaller the spatial area for each feature vector), the more the system is robust to view changes.

5.5.2 PCA Based System

In PCA based feature extraction [85, 174], a given face image is represented by a matrix containing grey level pixel values. The matrix is then converted to a face vector, $\boldsymbol{f}$, by concatenating all the columns. A D-dimensional feature vector, $\boldsymbol{x}$, is then obtained by:

$$\boldsymbol{x} = \boldsymbol{U}^T(\boldsymbol{f} - \boldsymbol{f}_\mu) \tag{5.2}$$

where $\boldsymbol{U}$ contains D eigenvectors (corresponding to the D largest eigenvalues) of the training data covariance matrix, and $\boldsymbol{f}_\mu$ is the mean of training face vectors. In our experiments we use frontal faces from group A to find $\boldsymbol{U}$ and $\boldsymbol{f}_\mu$.

5.6 GMM Based Classifier

As the GMM classifier is central to the methods proposed in this chapter, we overview it below. The distribution of training feature vectors for each person's face is modelled by a GMM. There is

also a secondary model, the generic model, which models the distribution of a population of faces, or interpreted alternatively, it represents "generic" face.

In the verification task we wish to find out whether a set of (test) feature vectors, $X = \{\boldsymbol{x}_i\}_{i=1}^{N_V}$, extracted from an unknown person's face, belongs to person C (which we will refer to as client C) or someone else (i.e. this is a two class recognition task). We first find the likelihood of set X belonging to client C with[2]:

$$p(X|\lambda_C) = \prod_{i=1}^{N_V} p(\boldsymbol{x}_i|\lambda_C) \tag{5.3}$$

where $p(\boldsymbol{x}|\lambda) = \sum_{g=1}^{N_G} w_g \ \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ and $\lambda = \{w_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\}_{g=1}^{N_G}$. Here, $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a D-dimensional Gaussian function with mean $\boldsymbol{\mu}$ and diagonal covariance matrix $\boldsymbol{\Sigma}$:

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \tag{5.4}$$

λ_C is the parameter set for client C, N_G is the number of Gaussians and w_g is the weight for Gaussian g (with constraints $\sum_{g=1}^{N_G} w_g = 1$ and $\forall\ g : w_g \geq 0$). Secondly, we obtain $P(X|\lambda_{generic})$, which is the likelihood of set X describing someone else's face (which we shall refer to as an impostor face). A log-likelihood ratio is then found using

$$\Lambda(X|\lambda_C, \lambda_{generic}) = \log p(X|\lambda_C) - \log p(X|\lambda_{generic}) \tag{5.5}$$

The verification decision is reached as follows: given a threshold t, the set X (i.e. the face in question) is classified as belonging to client C when $\Lambda(X|\lambda_C, \lambda_{generic}) \geq t$ or to an impostor when $\Lambda(X|\lambda_C, \lambda_{generic}) < t$. Note that $\Lambda(X|\lambda_C, \lambda_{generic})$ can be interpreted as an opinion of how more likely set X represents client C's face than an impostor's face, and hence can also be used in an open set identification system. Methods for obtaining the parameter set for the generic model and each client model are described in the following sections.

Note that in (5.3) each vector in the set $X = \{\boldsymbol{x}_i\}_{i=1}^{N_V}$ was assumed to be independent and identically distributed [46, 188]. When using local features, this results in the spatial relations between face parts to be not used, resulting in robustness to translations of the face [33, 152] (see also Section 4.3.8).

5.6.1 Classifier Training for the DCT Based System

First, the parameters for the generic model are obtained via the Expectation Maximisation (EM) algorithm [46, 42, 109], using all 0^o (frontal) data from group A. Here, the EM algorithm tunes the model parameters to optimise the maximum likelihood criterion. The parameters (λ) for each client model are then found by using the client's training data and adapting the generic model. The adaptation is traditionally done using a form of maximum a-posteriori (MAP) estimation (see

[2] In this chapter we have omitted the normalisation factor $\frac{1}{N_V}$ as the number of feature vectors extracted by a particular method is always the same.

Section 2.4.2). In this work we shall also employ the MLLR model transformation approaches as adaptation methods. The choice of the adaptation technique depends on the non-frontal model synthesis method used later (Section 5.8).

5.6.2 Classifier Training for the PCA Based System

The subset of the FERET dataset that is used in this work has only one frontal image per person. In PCA-based feature extraction, this results in only one training vector, leading to necessary constraints in the structure of the classifier and the classifier's training paradigm.

The generic model and all client models for frontal faces are constrained to have only one component (i.e. one Gaussian), with a diagonal covariance matrix[3]. The mean and the covariance matrix of the generic model are taken to be the mean and the covariance covariance matrix of feature vectors from group A, respectively. Instead of adaptation (as done in the DCT based system), each client model inherits the covariance matrix from the generic model. Moreover, the mean of each client model is taken to be the single training vector for that client.

5.7 Maximum Likelihood Linear Regression

In the Maximum Likelihood Linear Regression (MLLR) framework [53, 96], the adaptation of a given model is performed in two steps. In the first step the means are updated while in the second step the covariance matrices are updated, such that:

$$p(X|\widetilde{\lambda}) \geq p(X|\widehat{\lambda}) \geq p(X|\lambda) \tag{5.6}$$

where $\widetilde{\lambda}$ has both means and covariances updated while $\widehat{\lambda}$ has only means updated. The weights are not adapted as the main differences are assumed to be reflected in the means and covariances.

5.7.1 Adaptation of Means

Each adapted mean is obtained by applying a transformation matrix $\boldsymbol{W}_S$ to each original mean:

$$\widehat{\boldsymbol{\mu}}_g = \boldsymbol{W}_S \boldsymbol{\nu}_g \tag{5.7}$$

where $\boldsymbol{\nu}_g = [\ 1\ \boldsymbol{\mu}_g^T\ ]^T$ and W_S is a $D \times (D+1)$ transformation matrix which maximises the likelihood of given training data. For $\boldsymbol{W}_S$ shared by N_S Gaussians $\{g_r\}_{r=1}^{N_S}$ (see Section 5.7.3 below), the general form for finding $\boldsymbol{W}_S$ is:

[3]The assumption of a diagonal covariance matrix is supported by the fact that PCA derived feature vectors are decorrelated [46, 188].

$$\sum_{i=1}^{N_V}\sum_{r=1}^{N_S} p(g_r|\boldsymbol{x}_i,\lambda)\boldsymbol{\Sigma}_{g_r}^{-1}\boldsymbol{x}_i\boldsymbol{\nu}_{g_r}^T = \sum_{i=1}^{N_V}\sum_{r=1}^{N_S} p(g_r|\boldsymbol{x}_i,\lambda)\boldsymbol{\Sigma}_{g_r}^{-1}\boldsymbol{W}_S\boldsymbol{\nu}_{g_r}\boldsymbol{\nu}_{g_r}^T \tag{5.8}$$

where

$$p(g|\boldsymbol{x}_i,\lambda) = \frac{w_g\,\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g)}{\sum_{n=1}^{N_G} w_n\,\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_n,\boldsymbol{\Sigma}_n)} \tag{5.9}$$

As further elucidation is quite tedious, the reader is referred to [96] for the full solution of $\boldsymbol{W}_S$.

Two forms of $\boldsymbol{W}_S$ were originally proposed: full and "diagonal" [96]. We shall refer to MLLR transformation with a full transformation matrix as full-MLLR. When the transformation matrix is forced to be "diagonal", it has the following form:

$$\boldsymbol{W}_S = \begin{bmatrix} w_{1,1} & w_{1,2} & 0 & \cdots & 0 \\ w_{2,1} & 0 & w_{2,3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{D,1} & 0 & 0 & \cdots & w_{D,D+1} \end{bmatrix} \tag{5.10}$$

We shall refer to MLLR transformation with a "diagonal" transformation matrix as diag-MLLR. We propose a third form of MLLR, where the "diagonal" elements are set to one, i.e.:

$$\boldsymbol{W}_S = \begin{bmatrix} w_{1,1} & 1 & 0 & \cdots & 0 \\ w_{2,1} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{D,1} & 0 & 0 & \cdots & 1 \end{bmatrix} \tag{5.11}$$

In other words, each mean is transformed by adding an offset.
As such, Eqn. (5.7) can be rewritten as:

$$\widehat{\boldsymbol{\mu}}_g = \boldsymbol{\mu}_g + \boldsymbol{\Delta}_S \tag{5.12}$$

where $\boldsymbol{\Delta}_S$ maximises the likelihood of given training data. This leads to the following solution:

$$\boldsymbol{\Delta}_S = \left[\sum_{r=1}^{N_S}\sum_{i=1}^{N_V} p(g_r|\boldsymbol{x}_i,\lambda)\boldsymbol{\Sigma}_{g_r}^{-1}\right]^{-1}\left[\sum_{r=1}^{N_S}\sum_{i=1}^{N_V} p(g_r|\boldsymbol{x}_i,\lambda)\boldsymbol{\Sigma}_{g_r}^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}_{g_r})\right] \tag{5.13}$$

The derivation for the above solution is given in Appendix C. We shall refer to this form of MLLR as offset-MLLR.

5.7.2 Adaptation of Covariance Matrices

Once the new means are obtained, each new covariance matrix is found using [53]:

$$\widetilde{\boldsymbol{\Sigma}}_g = \boldsymbol{B}_g^T \boldsymbol{H}_S \boldsymbol{B}_g \tag{5.14}$$

where

$$\boldsymbol{B}_g = \boldsymbol{C}_g^{-1} \tag{5.15}$$

$$\boldsymbol{C}_g \boldsymbol{C}_g^T = \boldsymbol{\Sigma}_g^{-1} \tag{5.16}$$

Here, Eqn. (5.16) is a form of Cholesky decomposition [150]. $\boldsymbol{H}_S$, shared by N_S Gaussians $\{g_r\}_{r=1}^{N_S}$, is found with:

$$\boldsymbol{H}_S = \frac{\sum_{r=1}^{N_S} \left\{ \boldsymbol{C}_{g_r}^T \left[\sum_{i=1}^{N_V} p(g_r|\boldsymbol{x}_i, \lambda)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_{g_r})(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_{g_r})^T \right] \boldsymbol{C}_{g_r} \right\}}{\sum_{i=1}^{N_V} \sum_{r=1}^{N_S} p(g_r|\boldsymbol{x}_i, \lambda)} \tag{5.17}$$

The covariance transformation may be either full or diagonal. When the full transformation is used, full covariance matrices can be produced even if the original covariances were diagonal to begin with. To avoid this, the off-diagonal elements of $\boldsymbol{H}_S$ can be set to zero. In this work we restrict ourselves to the use of diagonal covariance matrices to reduce the number of parameters that need to be estimated. For full covariance matrices the dataset may not be large enough to robustly estimate the transformation parameters, which could result in the transformed covariance matrices being ill-conditioned [53].

5.7.3 Regression Classes

If each Gaussian is transformed individually, then for full-MLLR there are $D^2 + 2D$ parameters to estimate per Gaussian (i.e. $D \times (D + 1)$ parameters for each mean and D parameters for each covariance matrix); for diag-MLLR, there are $D + D + D = 3D$ parameters and for offset-MLLR there are $D + D = 2D$ parameters. Ideally each Gaussian would have its own transform, however in practical applications the training dataset may not be large enough to reliably estimate the required number of parameters. One way of working around the small training dataset problem is to share a transform across two or more Gaussians [53, 96]. We define which Gaussians are to share a transform by clustering the Gaussians based on the distance between their means.

We define a regression class as $\{g_r\}_{r=1}^{N_S}$ where g_r is the r-th Gaussian in the class. All Gaussians in a regression class share the same mean and covariance transforms. In our experiments we vary the number of regression classes from one (all Gaussians share one mean and one covariance transform) to 32 (each Gaussian has its own transform). The number of regression classes is denoted as N_R.

5.8 Synthesising Client Models for Non-Frontal Views

5.8.1 DCT Based System

In the MLLR based model synthesis technique, we first transform, using prior information, the frontal generic model into a non-frontal generic model for angle Θ. For full-MLLR and diag-MLLR, the parameters which describe the transformation of the means and covariances are $\Psi = \{\boldsymbol{W}_g, \boldsymbol{H}_g\}_{g=1}^{N_G}$, while for offset-MLLR the parameters are $\Psi = \{\boldsymbol{\Delta}_g, \boldsymbol{H}_g\}_{g=1}^{N_G}$. Parameters $\boldsymbol{W}_g$, $\boldsymbol{\Delta}_g$ and $\boldsymbol{H}_g$ are found as described in Section 5.7. When several Gaussians share the same transformation parameters, the shared parameters are replicated for each Gaussian in question. To synthesise a client model for angle Θ (i.e. a non-frontal view) the previously learned transformations are applied to the client's frontal model. The weights are kept the same as for the frontal model. Moreover, each frontal client model is derived from the frontal generic model by MLLR.

5.8.2 PCA Based System

For the PCA based system, we use MLLR based model synthesis in a similar way as described in the section above. The only difference is that each non-frontal client model inherits the covariance matrix from the corresponding non-frontal generic model. Moreover, as each client model has only one Gaussian, we note that the MLLR transformations are "single point to single point" transformations, where the points are the old and new mean vectors.

As described in Section 5.6.2, the mean of each client model is taken to be the single training vector available. Thus in this case a transformation in the feature domain is equivalent to a transformation in the model domain. It is therefore possible to use transformations which are not of the "single point to single point" type. Let us suppose that we have the following multi-variate linear regression model:

$$\boldsymbol{B} = \boldsymbol{A}\,\boldsymbol{W} \tag{5.18}$$

$$\begin{bmatrix} \boldsymbol{b}_1^T \\ \boldsymbol{b}_2^T \\ \vdots \\ \boldsymbol{b}_N^T \end{bmatrix} = \begin{bmatrix} 1 & \boldsymbol{a}_1^T \\ 1 & \boldsymbol{a}_2^T \\ & \vdots \\ 1 & \boldsymbol{a}_N^T \end{bmatrix} \begin{bmatrix} w_{1,1} & \cdots & w_{1,D} \\ w_{2,1} & \cdots & w_{2,D} \\ \vdots & \vdots & \vdots \\ w_{D+1,1} & \cdots & w_{D+1,D} \end{bmatrix} \tag{5.19}$$

where $N > D+1$, with D being the dimensionality of $\boldsymbol{a}$ and $\boldsymbol{b}$. $\boldsymbol{W}$ is a matrix of unknown regression parameters. Under the sum-of-least-squares regression criterion, $\boldsymbol{W}$ can be found using [150]:

$$\boldsymbol{W} = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{B} \tag{5.20}$$

Compared to MLLR, this type of regression finds a common relation between two sets of points – as such it may be more accurate than MLLR. Given a set of PCA-derived feature vectors from

group A, representing faces at 0^o (frontal view) and Θ (a non-frontal view), we find $\boldsymbol{W}$. We can then synthesise the single mean for Θ from client C's 0^o mean using:

$$\boldsymbol{\mu}^{\Theta} = \left[\ 1 \ \ (\boldsymbol{\mu}^{0^o})^T \ \right] \ \boldsymbol{W} \tag{5.21}$$

We shall refer to this PCA-specific linear regression based technique as LinReg. We note that for this synthesis technique, $(D+1)\times D = D^2+D$ parameters need to be estimated.

5.9 Multi-Angle Models

In order for the system to automatically handle non-frontal views, each client's frontal model is extended by concatenating it with synthesised non-frontal models. The frontal generic model is also extended with non-frontal generic models. Formally, an extended (or multi-angle) model is created using:

$$\begin{aligned} \lambda^{extended} &= \lambda^{0^o} \sqcup \lambda^{+60^o} \sqcup \lambda^{+40^o} \cdots \sqcup \lambda^{\text{-}40^o} \sqcup \lambda^{\text{-}60^o} \\ &= \sqcup_{i\in\Phi} \lambda^i \end{aligned} \tag{5.22}$$

where λ^{0^o} represents a frontal model, Φ is a set of angles, e.g. $\Phi = \{\ 0^o, +60^o, \cdots, +15^o, -15^o, \cdots, -60^o\ \}$, and $\sqcup$ is an operator for joining GMM parameter sets, defined below.

Let us suppose we have two GMM parameter sets, λ^x and λ^y, comprised of parameters for N_G^x and N_G^y Gaussians, respectively. The $\sqcup$ operator is defined as follows:

$$\begin{aligned} \lambda^z &= \lambda^x \sqcup \lambda^y \\ &= \left\{\alpha w_g^x,\ \boldsymbol{\mu}_g^x,\ \boldsymbol{\Sigma}_g^x\right\}_{g=1}^{N_G^x} \cup \left\{\beta w_g^y,\ \boldsymbol{\mu}_g^y,\ \boldsymbol{\Sigma}_g^y\right\}_{g=1}^{N_G^y} \end{aligned} \tag{5.23}$$

where $\alpha = N_G^x/(N_G^x + N_G^y)$ and $\beta = 1-\alpha$.

5.10 Experiments and Discussion

5.10.1 DCT Based System

In the first experiment we studied how the overlap setting in the DCT-mod2 feature extractor and number of Gaussians in the classifier affects performance and robustness. Client models were trained on frontal faces and tested on faces at 0^o and $+40^o$ views; impostor faces matched the testing view. Traditional MAP adaptation was used to obtain the client models. Results, in terms of EER, are shown in Figures 5.4 and 5.5.

When testing with frontal faces, the overall trend is that as the overlap increases more Gaussians are needed to decrease the error rate. This can be interpreted as follows: the smaller the area used in the derivation of each feature vector, the more Gaussians are required to adequately model the face. When testing with non-frontal faces, the overall trend is that as the overlap increases, the

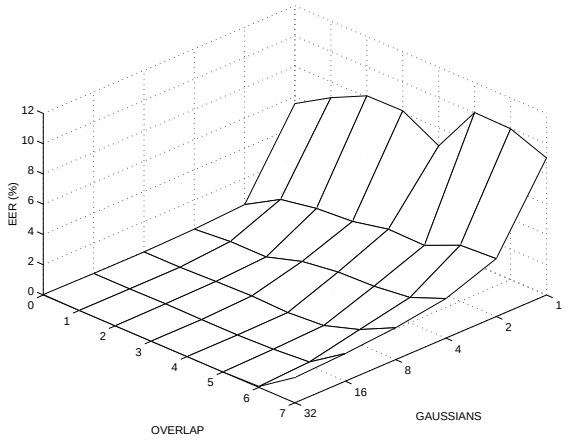

Figure 5.4: EER of the DCT-based system trained and tested on frontal faces, for varying degrees of overlap and number of Gaussians. Traditional MAP based training was used.

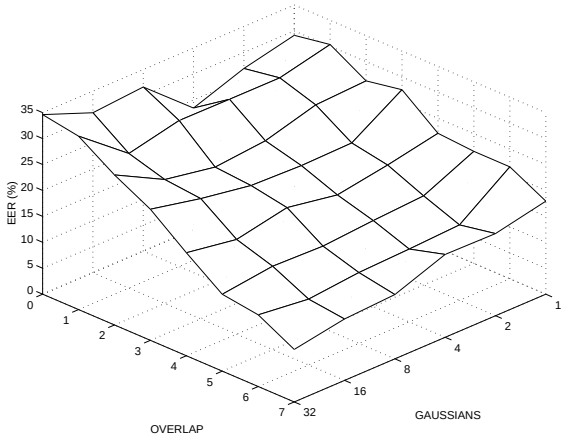

Figure 5.5: EER of the DCT based system trained on frontal faces and tested on $+40^o$ faces, for varying degrees of overlap and number of Gaussians. Traditional MAP based training was used.

lower the error rate. There is also a less defined trend when the overlap is 4 pixels or greater: the more Gaussians, the lower the error rate[4]. While not shown here, the DCT based system obtained similar trends for non-frontal views other than $+40^o$. The best performance for $+40^o$ faces is achieved with an overlap of 7 pixels and 32 Gaussians, resulting in an EER close to 10%. We chose this configuration for further experiments.

In the second experiment we evaluated the performance of models synthesised via the full-MLLR, diag-MLLR and offset-MLLR techniques, for varying number of regression classes. Results are presented in Tables 5.1 to 5.4. As can be observed, the full-MLLR technique falls apart when there are two or more regression classes. Its best results (obtained for one regression class) are in some cases worse than for standard frontal models. Frontal client models, obtained by using full-MLLR as an adaptation method, resulted in an EER of 0% for frontal faces for all configurations of regression classes. Thus while the full-MLLR transformation is adequate for adapting the frontal generic model to frontal client models, the synthesis results suggest that the transformation is only reliable when applied to the specific model it was trained to transform. A further investigation of the sensitivity of the full-MLLR transform is given in the following section.

Compared to full-MLLR, the diag-MLLR technique obtains lower EERs (Table 5.2). We note that the number of transformation parameters for diag-MLLR is significantly less than for full-MLLR. The overall error rate (across all angles) decreases as the number of regression classes increases from one to eight. The performance then deteriorates for higher numbers of

[4]This is true up to a point: eventually the error rate will go up as there will be too many Gaussians to train adequately with the small size of the training dataset. Preliminary experiments showed that there was little performance gain when using more than 32 Gaussians.

regression classes. The results are consistent with the scenario that once the number of regression classes reaches a certain point, the training dataset is too small to obtain robust transformation parameters. The best performance, obtained at eight regression classes, is for all angles better than the performance of standard frontal models.

The offset-MLLR technique (Table 5.3) has the lowest EERs when compared to full-MLLR and diag-MLLR. It must be noted that it also has the least number of transformation parameters. The overall error rate consistently decreases as the number of regression classes increases from one to 32. The best performance, obtained at 32 regression classes, is for all angles better than the performance of standard frontal models.

Angle	N_R=1	N_R=2	N_R=4	N_R=8	N_R=16	N_R=32
-60^o	23.58	48.83	49.50	49.56	49.94	49.81
-40^o	13.11	49.61	49.58	49.50	49.47	49.56
-25^o	5.81	50.39	49.56	49.56	49.97	49.64
-15^o	1.58	49.83	49.47	49.67	49.75	49.69
$+15^o$	1.28	50.19	49.58	49.61	49.81	49.58
$+25^o$	4.69	50.17	49.67	49.69	49.97	49.56
$+40^o$	9.39	49.25	49.67	49.67	49.64	49.53
$+60^o$	19.53	49.81	49.64	49.81	49.75	49.64

Table 5.1: EER of the full-MLLR synthesis technique for varying number of regression classes (DCT-based system).

Angle	N_R=1	N_R=2	N_R=4	N_R=8	N_R=16	N_R=32
-60^o	23.56	22.69	22.11	18.33	23.67	32.61
-40^o	11.86	11.97	11.14	11.19	15.28	25.17
-25^o	5.25	5.72	4.75	3.86	8.06	16.75
-15^o	1.64	1.58	1.56	1.50	3.53	16.81
$+15^o$	1.36	1.36	1.33	1.36	2.50	15.67
$+25^o$	4.97	4.42	4.36	3.69	5.92	20.72
$+40^o$	8.97	8.33	7.86	8.78	17.14	29.28
$+60^o$	19.81	16.97	16.86	15.31	31.22	31.25

Table 5.2: EER of the diag-MLLR synthesis technique for varying number of regression classes (DCT-based system).

Angle	N_R=1	N_R=2	N_R=4	N_R=8	N_R=16	N_R=32
-60^o	23.31	22.78	22.47	19.67	16.97	17.94
-40^o	12.28	11.00	10.06	10.83	9.25	7.94
-25^o	4.89	5.31	4.64	3.72	3.33	3.44
-15^o	1.58	1.58	1.56	1.53	1.44	1.44
$+15^o$	1.36	1.36	1.33	1.33	1.42	1.42
$+25^o$	4.94	4.67	4.42	3.33	3.08	3.28
$+40^o$	9.00	7.42	7.08	7.42	6.81	6.67
$+60^o$	19.86	18.94	18.81	17.11	15.44	14.33

Table 5.3: EER of the offset-MLLR synthesis technique for varying number of regression classes (DCT-based system).

Angle	standard (frontal models)	full-MLLR (N_R=1)	diag-MLLR (N_R=8)	offset-MLLR (N_R=32)
-60^o	22.72	23.58	18.33	$*$ 17.94
-40^o	11.47	13.11	11.19	$*$ 7.94
-25^o	5.72	5.81	3.86	$*$ 3.44
-15^o	2.83	1.58	1.50	$*$ 1.44
$+15^o$	2.64	$*$ 1.28	1.36	1.42
$+25^o$	5.94	4.69	3.69	$*$ 3.28
$+40^o$	10.11	9.39	8.78	$*$ 6.67
$+60^o$	24.72	19.53	15.31	$*$ 14.33

Table 5.4: EER for standard frontal models (obtained via traditional MAP based training) and models synthesised for non-frontal angles via MLLR based techniques (DCT-based system). Best result for a given angle is indicated by an asterisk.

5.10.2 Analysis of MLLR Sensitivity

The results presented in the previous section show that the full-MLLR technique is only reliable when applied directly to the specific model it was trained to transform, making the full-MLLR transform unsuitable for model synthesis (where a related model is transformed, instead of the model for which the transformation was learned). In this section we explore this observation further by measuring how sensitive the full-MLLR, diag-MLLR and offset-MLLR transforms are to perturbations of the model they were trained to transform.

The sensitivity is measured as follows. The transformation of the frontal generic model to a $+60^o$ generic model is learned (using 32 regression classes) and the average log-likelihood of $+60^o$ data from group A is found:

$$\mathcal{A}(X|\lambda_{generic}^{+60^o}) = \frac{1}{N_V} \log p(X|\lambda_{generic}^{+60^o}) \tag{5.24}$$

The mean vectors of the frontal generic model are then "corrupted" by adding Gaussian noise with zero mean and various levels of variance. Formally:

$$\left[\boldsymbol{\mu}_g^{corrupted}\right]^T = \left[\mu_{g,d}^{original} + \mathcal{R}(0,\sigma^2)\right]_{d=1}^{D} \tag{5.25}$$

where $\mu_{g,d}$ is the d-th element of $\boldsymbol{\mu}_g$ and $\mathcal{R}(0,\sigma)$ is a Gaussian distributed random variable with zero mean and variance σ^2. The previously learned transformation is applied to the "corrupted" frontal generic model to obtain a "corrupted" $+60^o$ generic model. The average log-likelihood of $+60^o$ data from group A is then found as per Eqn. (5.24). This process is repeated ten times for each variance setting and the mean of the average log-likelihood is taken. The mean value represents how well the transformed model represents the $+60^o$ data; the lower the value, the worse the representation. Results are presented in Table 5.5.

By treating the mean vectors of frontal client models as noisy instances of the frontal generic model mean vectors (where the frontal client models were derived from the original frontal generic model), it is possible to measure the overall "variance" of the frontal mean vectors. This is the variance that a synthesis technique must handle. While the frontal client models also differ from the frontal generic model in their covariance matrices, we believe this approach nevertheless provides suggestive results.

The full-MLLR, diag-MLLR and offset-MLLR approaches for deriving frontal client models (from the original frontal generic model) obtained similar overall "variance" of frontal client means of around 90. From the results shown in Table 5.5 it can be observed that the full-MLLR transformation is easily affected by small perturbations of the frontal generic model. Close to level of the required variance (i.e. at 100), the full-MLLR approach produces a $+60^o$ generic model which very poorly represents the data on which the transform was originally trained. In comparison, the diag-MLLR and offset-MLLR transforms are largely robust to perturbations of the frontal generic model, with the offset-MLLR approach the most stable.

noise variance	full-MLLR	diag-MLLR	offset-MLLR
0	-74.81	-74.81	-74.81
1×10^{-7}	-76.51	-74.81	-74.81
1×10^{-6}	-78.76	-74.81	-74.81
1×10^{-5}	-83.34	-74.81	-74.81
1×10^{-4}	-91.63	-74.82	-74.81
1×10^{-3}	-119.95	-74.85	-74.81
1×10^{-2}	-367.01	-75.14	-74.81
1×10^{-1}	-246.57×10^{1}	-75.55	-74.82
1	-313.49×10^{2}	-76.80	-74.92
$1 \times 10^{+1}$	-205.79×10^{3}	-78.29	-75.96
$1 \times 10^{+2}$	-172.71×10^{4}	-84.32	-81.59
$1 \times 10^{+3}$	-283.12×10^{5}	-104.29	-95.81

Table 5.5: Mean of the average log-likelihood [Eqn. (5.24)] computed using $+60^o$ generic model; the $+60^o$ generic model was derived from a noise corrupted frontal generic model using a fixed transform (either full-MLLR, diag-MLLR or offset-MLLR).

5.10.3 PCA Based System

In the first experiment we studied how the dimensionality of the feature vectors used in the PCA based system affects robustness to varying pose. Client models were trained on frontal faces and tested on faces from -60^o to $+60^o$ views; impostor faces matched the testing view. Results for -60^o to 0^o are shown in Figure 5.6 (results for $+15^o$ to $+60^o$, not shown, have very similar trends).

As can be observed, a dimensionality of at least 40 is required to achieve perfect verification on frontal faces (this is consistent with the results presented in [158]). For non-frontal faces at $\pm 60^o$ and $\pm 40^o$, the error rate generally increases as the dimensionality increases, and saturates when the dimensionality is about 15. Hence there is somewhat of a trade-off between the error rates on frontal faces and non-frontal faces, controlled by the dimensionality. Since in this work we are pursuing extensions to standard 2D approaches, the dimensionality has been fixed at 40 for further experiments. Using a lower dimensionality of, say 4, offers better performance for non-frontal faces, however it comes at the cost of an EER of about 10% on frontal faces.

We note that the PCA based system (which is holistic in nature) is much more affected by view changes than the DCT based system. This can be attributed to the rigid preservation of spatial relations between face areas, which is in contrast to the DCT/GMM based approach, where the spatial relations between face parts are very loose. The loose spatial relations allow for the deformations and movements of face areas, which can occur due to view changes. Interestingly, empirical evidence suggests that humans may recognise faces by parts rather than in a holistic manner [106].

In the second experiment we evaluated the performance of models synthesised using LinReg and MLLR-based techniques. As there is only one Gaussian per client model, there was only one regression class for MLLR techniques. Results in Table 5.6 show that model synthesis with full-MLLR and diag-MLLR was unsuccessful. Since the LinReg technique works quite well and has a similar number of free parameters as full-MLLR, we attribute the failure of full-MLLR and diag-MLLR to their sensitivity to the starting point, which was described in the preceding section.

While models synthesised by offset-MLLR exhibit better performance than standard frontal models, they are easily outperformed by models synthesised via the LinReg technique. This supports the view that "single point to single point" type transformations (such as MLLR) are less useful for the PCA-based system.

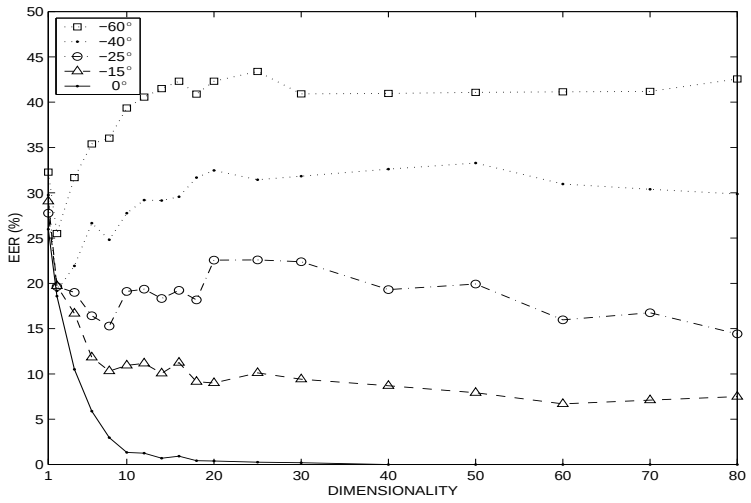

Figure 5.6: EER of the PCA based system (trained on frontal faces) for increasing dimensionality and the following angles: -60^o, -40^o, -25^o, -15^o and 0^o (frontal).

Angle	frontal	full-MLLR	diag-MLLR	offset-MLLR	LinReg
-60^o	40.97	49.67	50.00	38.56	$*$ 14.92
-40^o	32.61	50.00	49.97	25.75	$*$ 17.19
-25^o	19.31	49.69	49.75	$*$ 13.81	15.78
-15^o	8.69	49.58	49.72	6.86	$*$ 6.44
$+15^o$	10.39	49.67	49.69	8.36	$*$ 5.72
$+25^o$	20.83	49.58	49.97	14.00	$*$ 7.78
$+40^o$	34.36	49.78	50.00	28.97	$*$ 15.00
$+60^o$	44.92	49.83	49.47	38.44	$*$ 14.89

Table 5.6: Performance comparison (in terms of EER) between frontal models and synthesised non-frontal models for the PCA based system. Best result for a given angle is indicated by an asterisk.

5.10.4 Performance of Multi-Angle Models

In the experiments described in Sections 5.10.1 and 5.10.3, it was assumed that the angle of the face is known. In this section we progressively remove this constraint and propose to handle varying pose by extending each client's frontal model with the client's synthesised non-frontal models. We shall refer to these extended models as multi-angle models.

In the first experiment we compare the performance of multi-angle models to frontal models and models synthesised for a specific angle; impostor faces matched the test view. For the DCT based system, each client's frontal model is extended with models synthesised by the offset-MLLR technique (with 32 regression classes) for the following angles: $\pm 60^o$, $\pm 40^o$ and $\pm 25^o$. Synthesised models for $\pm 15^o$ were not used since they provided little performance benefit over the 0^o model (see Table 5.4). The frontal generic model is also extended with non-frontal generic models. Since each frontal model had 32 Gaussians, each multi-angle model has 224 Gaussians. Following the offset-MLLR based model synthesis paradigm, each frontal client model is derived from the frontal generic model using offset-MLLR. The results are presented in Table 5.7 and Figure 5.7.

For the PCA based system, model synthesis is accomplished using LinReg. Each client's frontal model is extended for the following angles: $\pm 60^o$, $\pm 40^o$, $\pm 25^o$ and $\pm 15^o$. The frontal generic model is also extended with non-frontal generic models. Since each frontal model had one Gaussian, each extended model has nine Gaussians. The results are given in Table 5.8 and Figure 5.8.

The results show that for most angles the multi-angle models have only a small reduction in performance when compared to models synthesised for a specific angle. This also suggests that the multi-angle approach could be used instead of selecting the most appropriate synthesised model (via detection of the face angle), thus reducing the complexity of a multi-view face verification system.

In the first experiment impostor attacks and true claims were evaluated for each angle separately. In the second experiment we relax this restriction and allow true claims and impostor attacks to come from all angles, resulting in $90 \times 9 = 810$ true claims and $90 \times 20 \times 9 = 16200$ impostor attacks. An overall EER is then found. For both DCT and PCA based systems two types of models are used: frontal and multi-angle. For the DCT based system, frontal models were derived from the generic model using offset-MLLR. From the results presented in Table 5.9, it can be observed that the multi-angle approach reduces the error rate in both PCA and DCT based systems, with the DCT based system achieving the lowest EER. The largest error reduction is present in the PCA based system, where the EER is reduced by a remarkable 57.9%. For the DCT based system the EER is reduced by a notable 26.1%.

Angle	Frontal	Synth'd	Multi-Angle
-60^o	28.22	17.94	18.25
-40^o	15.17	7.94	9.36
-25^o	6.06	3.44	3.28
-15^o	1.61	1.44	1.64
$+15^o$	1.44	1.42	1.67
$+25^o$	5.67	3.28	3.53
$+40^o$	9.39	6.67	5.94
$+60^o$	23.75	14.33	16.56

Table 5.7: EER performance of the DCT based system using frontal, synthesised (for a specific angle) and multi-angle models. Offset-MLLR based training (frontal models) and synthesis (non-frontal models) was used.

Angle	Frontal	Synth'd	Multi-Angle
-60^o	40.97	14.92	15.33
-40^o	32.61	17.19	17.56
-25^o	19.31	15.78	14.94
-15^o	8.69	6.44	9.17
$+15^o$	10.39	5.72	3.67
$+25^o$	20.83	7.78	8.11
$+40^o$	34.36	15.00	15.67
$+60^o$	44.92	14.89	16.08

Table 5.8: As per Table 5.7 but using the PCA based system. LinReg model synthesis was used.

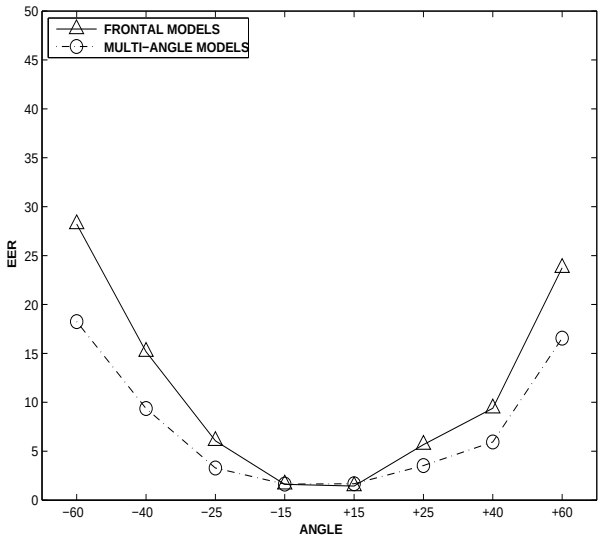

Figure 5.7: EER performance of the DCT based system using frontal and multi-angle models (data from Table 5.7).

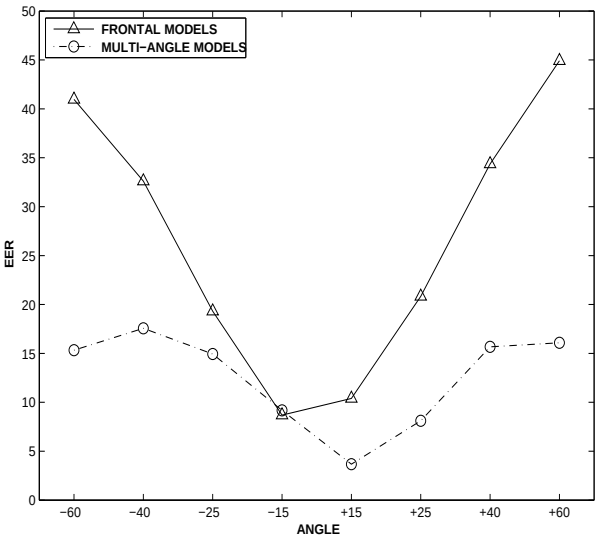

Figure 5.8: EER performance of the PCA based system using frontal and multi-angle models (data from Table 5.8).

System	Model type	
	Frontal	Multi-angle
PCA based	27.34	11.51
DCT based	14.82	10.96

Table 5.9: Overall EER performance of frontal and multi-angle models, where true claims and impostor attacks come from all available face angles.

5.11 Summary and Future Directions

In this chapter we addressed the pose mismatch problem which can occur in face verification systems that have only a single (frontal) face image available for training. In the framework of a Bayesian classifier based on mixtures of Gaussians, the problem was tackled through extending each frontal face model with artificially synthesised models for non-frontal views. The synthesis was accomplished via methods based on several implementations of Maximum Likelihood Linear Regression (MLLR) (originally developed for tuning speech recognition systems), and standard multi-variate linear regression (LinReg). To our knowledge this is the first time MLLR has been adapted for face verification.

All synthesis techniques rely on prior information and learn how face models for the frontal view are related to face models at non-frontal views. The synthesis and extension approach was evaluated by applying it to two face verification systems: a holistic system (utilising PCA derived features) and a local feature system (using DCT derived features).

Experiments on the FERET dataset suggest that for the PCA based system, the LinReg technique (which is based on a common relation between two sets of points) is more suited than the MLLR based techniques (which are "single point to single point" transforms in the PCA based system). For the DCT based system, the results show that synthesis via a new MLLR implementation obtains better performance than synthesis based on traditional MLLR (mainly due to a lower number of parameters). The results further suggest that extending frontal models considerably reduces errors in both systems.

The results also show that the standard DCT based system (trained on frontal faces) is less affected by view changes than the PCA based system. This can be attributed to the parts based representation of the face (via local features) and, due to the classifier based on mixtures of Gaussians, the lack of constraints on spatial relations between face parts. The lack of constraints allows for deformations and movements of face areas, which can occur due to view changes. This is in contrast to the PCA based system, where, due to the holistic representation, the spatial relations are rigidly kept. Interestingly, empirical evidence suggests that humans may recognise faces by parts rather than in a holistic manner [106].

Future areas of exploration, specific to dealing with pose variations, include whether it is possible to interpolate between two synthesised models to generate a third model for a view for which there is no prior information. A related question is how many discrete views are necessary to adequately cover a wide range of poses. The dimensionality reduction matrix $\boldsymbol{U}$ in the PCA approach was defined using only frontal faces; higher performance may be obtained by incorporating non-frontal faces.

The local feature/GMM approach can be extended by embedding position information into each feature vector [33], thus placing a weak constraint on the face areas each Gaussian can model (as opposed to the current absence of constraints). This in turn could make the transformation of frontal models to non-frontal models more accurate, as different face areas effectively "move" in different ways when there is a view change. Alternatively, the GMM based classifier can be replaced with a (more complex) pseudo-2D Hidden Markov Model based classifier [33, 48] (see also Section 4.2.3), where there is a more stringent constraint on the face areas modelled by each Gaussian. Lastly, it would be useful to devise alternative size normalisation approaches in order to address the face scaling problem mentioned in Section 5.4.

Chapter 6

Verification Using Fused Speech and Face Information

6.1 Overview

This chapter first provides an overview of key concepts in the information fusion area, followed by a review of important milestones in audio-visual person identification and verification. Several adaptive and non-adaptive techniques for reaching the verification decision, based on combined speech and face information, are then evaluated in clean and noisy audio conditions on a common dataset. It is shown that in clean conditions most of the non-adaptive approaches provide similar performance and in noisy conditions most exhibit a severe deterioration in performance. It is also shown that current adaptive approaches are either inadequate or use restrictive assumptions. A new category of classifiers is then introduced, where the decision boundary is fixed but constructed to take into account how the distributions of opinions are likely to change due to noisy conditions. Compared to a previously proposed adaptive approach, the proposed classifiers do not make a direct assumption about the type of noise that causes the mismatch between training and testing conditions.

6.2 Information Fusion Background

Broadly speaking, the term information fusion encompasses areas which deal with using a combination of different sources of information, either to generate one representational format, or to reach a decision. This includes: consensus building, team decision theory, committee machines, integration of multiple sensors, multi-modal data fusion, combination of multiple experts/classifiers, distributed detection and distributed decision making. Some pioneering publications can be traced back to early 1980s [14, 132, 170, 171].

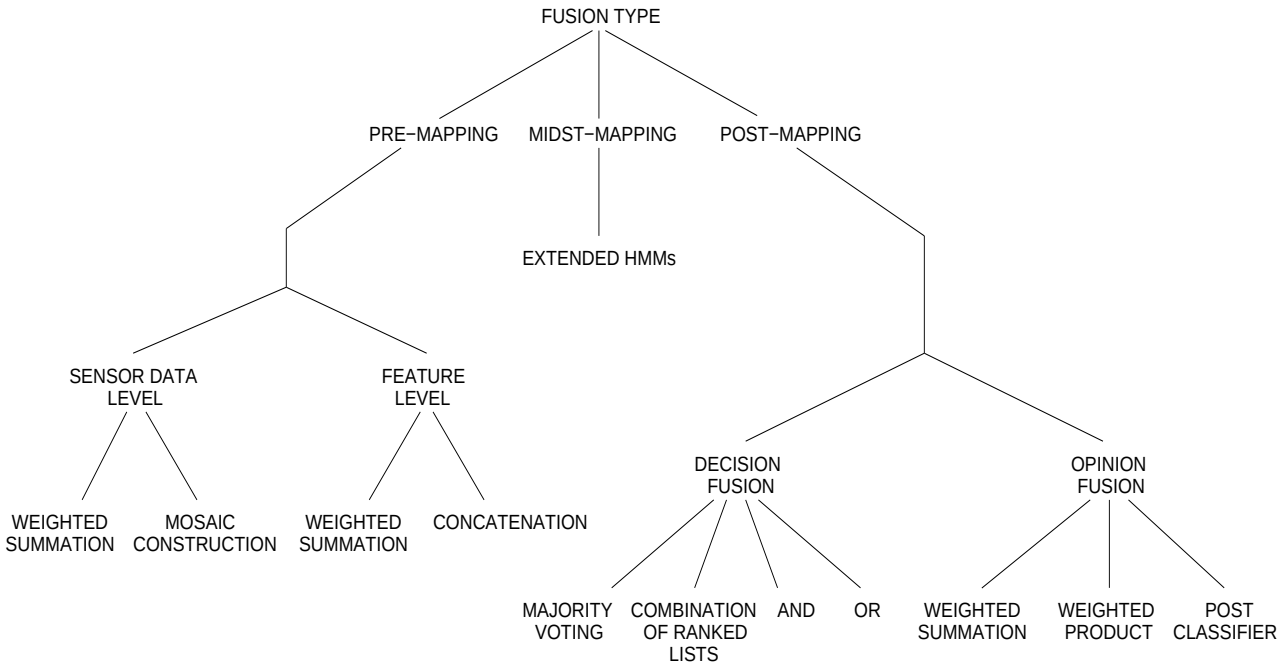

Figure 6.1: Tree of fusion types.

When looking from the point of decision making, there are several motivations for using information fusion:

- Using complementary information (e.g. audio and video) can reduce error rates.
- Use of multiple sensors (i.e. redundancy) can increase reliability.
- Cost of implementation can be reduced by using several cheap sensors rather than one expensive sensor.
- Sensors can be physically separated, allowing the acquisition of information from different points of view.

Humans use information fusion every day; some examples are: use of both eyes, seeing and touching the same object, or seeing and hearing a person talk (which improves intelligibility in noisy situations [166]). Several species of snakes combine infrared information with visual information [76, 101].

This section is an overview of key approaches to information fusion. In the literature information fusion is often divided into several categories: sensor data level fusion, feature level fusion, score fusion and decision fusion [66, 76, 156]. However, it is more intuitive to classify it into three main categories: pre-mapping fusion, midst-mapping fusion and post-mapping fusion, as shown in Figure 6.1. In pre-mapping fusion, information is combined before any use of classifiers or experts. In midst-mapping fusion, information is combined during mapping from sensor-data/feature space into opinion/decision space, while in post-mapping fusion, information is combined after mapping

from sensor-data/feature space into opinion/decision space (here the mapping is accomplished by an ensemble of experts or classifiers. While a classifier provides a hard decision, an expert provides an opinion (e.g. in the [0,1] interval) on each possible decision).

In pre-mapping fusion, there are two main sub-categories: sensor data level fusion and feature level fusion. In post-mapping fusion, there are also two main sub-categories: decision fusion and opinion fusion. It must be noted that in some literature (e.g. [66, 76, 179]) the term "decision fusion" also encompasses opinion fusion. However, since each expert provides an opinion and not a decision, sub-typing opinion fusion under "decision fusion" is questionable.

Silsbee and Bovik [166] refer to pre-mapping fusion and post-mapping fusion as pre-categorical integration and post-categorical integration, respectively. Wark [184] refers to pre-mapping fusion as input level or early fusion and post-mapping fusion as classifier level or late fusion. Ross and Jain [156] refer to opinion fusion as score fusion.

In order to aid understanding, the following description of fusion methods is presented in the general context of class identification. Wherever necessary, comments are included to elucidate a fusion approach in terms of the verification application. This section leads onto the review of important milestones in the field of information fusion in audio-visual person recognition (Section 6.3).

6.2.1 Pre-mapping Fusion: Sensor Data Level

In sensor data level fusion [66], the raw data from sensors is combined. Depending on the application, there are two main methods to accomplish this: weighted summation and mosaic construction. For example, weighted summation can be employed to combine visual and infra-red images into one image, or, in the form of an average operation, to combine the data from two microphones (to reduce the effects of noise). The data must first be commensurate, which can be accomplished by mapping to a common interval. Mosaic construction can be employed to create one image out of images provided by several cameras, where each camera is observing a different part of the same object [76].

6.2.2 Pre-mapping Fusion: Feature Level

In feature level fusion, features extracted from data provided by several sensors (or from one sensor but using different feature extraction techniques [137]) are combined. If the features are commensurate, the combination can be accomplished by a weighted summation (e.g. features extracted from data provided by two microphones). If the features are not commensurate, feature vector concatenation can be employed [4, 66, 100, 156], where a new feature vector can be constructed by concatenating two or more feature vectors (e.g. to combine audio and visual features).

There are three downsides to the feature vector concatenation approach. The first is that there is no explicit control over how much each vector contributes to the final decision. The second

downside is that the separate feature vectors must be available at the same frame rate (i.e. the feature extraction must be synchronous), which can be a problem when combining speech and visual feature vectors[1]. The third downside is the dimensionality of the resulting feature vector, which can lead to the "curse of dimensionality" problem [46]. Due to the above problems, in many cases the post-mapping fusion approach is preferred (described in Sections 6.2.4 and 6.2.5).

6.2.3 Midst-Mapping Fusion

Compared to other fusion techniques presented in this chapter, midst-mapping fusion is a relatively new and more complex concept. Here several information streams are processed concurrently while mapping from feature space into opinion/decision space. Midst-mapping fusion can be used for exploitation of temporal synergies between the streams (e.g. a speech signal and the corresponding video of lip movements), with the ability to avoid problems present in vector concatenation. Examples of this type of fusion are extended Hidden Markov Models (adapted to handle multiple streams of data [19, 20, 138, 141]), which have been shown useful for text-dependent person verification [19, 123, 183].

6.2.4 Post-Mapping Fusion: Decision Fusion

In decision fusion [66, 76], each classifier in an ensemble of classifiers provides a hard decision. The classifiers can be of the same type but working with different features (e.g. audio and video data), non-homogeneous classifiers working with the same features, or a hybrid of the previous two types. The decisions can be combined by majority voting, combination of ranked lists, or using and & or operators.

The inspiration behind the use of non-homogeneous classifiers with the same features stems from the belief that each classifier (due to different internal representation) may be "good" at recognising a particular set of classes while being "bad" at recognising a different set of classes. Hence a combination of classifiers may overcome the "bad" properties of each classifier [71, 87].

Majority Voting

In majority voting [57, 76, 142], a consensus is reached on the decision by having a majority of the classifiers declaring the same decision. There are two downsides to the voting approach: (i) an odd number of classifiers is required to prevent ties; (ii) the number of classifiers must be greater than the number of classes (possible decisions) to ensure a decision is reached.

[1] For example, speech feature vectors are usually extracted at a rate of 100 per second [135], while visual features are constrained by the video camera's frame rate (25 fps in the PAL standard and 30 fps in the NTSC standard [173]).

Ranked List Combination

In ranked list combination [2, 71, 142], each classifier provides a ranked list of class labels, with the top entry indicating the most preferred class and the bottom entry indicating the least preferred class. The ranked lists can then be combined via various means [71], possibly taking into account the reliability and discrimination ability of each classifier. The decision is then usually reached by selecting the top entry in the combined ranked list.

AND Fusion

In AND fusion [101, 178], a decision is reached only when all the classifiers agree. As such, this type of fusion is quite restrictive. For multi-class problems no decision may be reached, thus it is mainly useful in situations where one would like to detect the presence of an event/object, with a low false acceptance bias. In a person verification scenario, where the task is to detect the presence of a true claimant, this translates to a low False Acceptance Rate (FAR) at the inconvenience of a high False Rejection Rate (FRR). Such an operating characteristic might be useful in high-security applications.

OR Fusion

In OR fusion [101, 178], a decision is made as soon as one of the classifiers makes a decision. In comparison to AND fusion, this type of fusion is very relaxed, providing multiple possible decisions in multi-class problems. Since in most multi-class problems this is undesirable, OR fusion is mainly useful where one would like to detect the presence of an event/object with a low false rejection bias. For the verification task this translates to a low FRR and high FAR.

6.2.5 Post-Mapping Fusion: Opinion Fusion

In opinion fusion [66, 76, 156, 179] (also referred to as score fusion), an ensemble of experts provides an opinion on each possible decision. Since non-homogeneous experts can be used (e.g. where one expert provides its opinion in terms of distances while another in terms of a likelihood measure), the opinions are usually required to be commensurate before further processing. This can be accomplished by mapping the output of each expert to the $[0, 1]$ interval[2], where 0 indicates the lowest opinion and 1 the highest opinion. While the term non-homogeneous usually implies a different expert structure, it is sufficient for a set of experts to be considered non-homogeneous if they are using different features (e.g. audio and video features, or different features extracted from one modality [137]).

In ranked list combination fusion (which doesn't require the mapping step) the rank itself could be considered to indicate the opinion of the classifier. However, compared to opinion fusion, some information regarding the "goodness" of each possible decision is lost.

[2]The mapping can be performed via a sigmoid. See Section 6.4.4 for more information.

Opinions can be combined using weighted summation or weighted product approaches before using a classification criterion, such as the max operator (which selects the class with the highest opinion), to reach a decision. Alternatively, a post-classifier can be used to directly reach a decision. In the former approach, each expert can be considered to be an elaborate discriminant function, working on its own section of the feature space [46].

The inherent advantage of weighted summation and product fusion over feature vector concatenation and decision fusion is that the opinions from each expert can be weighted. The weights can be selected to reflect the reliability and discrimination ability of each expert. Hence when fusing opinions from a speech and a face expert, it is possible to decrease the contribution of the speech expert when working in low audio SNR conditions (this type of fusion is known as adaptive fusion). The weights can also be optimised to satisfy a given criterion (e.g. to obtain EER performance).

Weighted Summation Fusion

In weighted summation, the opinions regarding class j from N_E experts are combined using:

$$f_j = \sum_{i=1}^{N_E} w_i o_{i,j} \tag{6.1}$$

where $o_{i,j}$ is the opinion from the i-th expert and w_i is the corresponding weight in the $[0, 1]$ interval, with the constraints $\sum_{i=1}^{N_E} w_i = 1$ and $\forall i : w_i \geq 0$. When all the weights are equal, Eqn. (6.1) reduces to an arithmetic mean operation. The weighted summation approach is also known as the sum rule [5, 87].

Weighted Product Fusion

The opinions can be interpreted as posterior probabilities in the Bayesian framework [30]. Assuming the experts are independent, the opinions regarding class j from N_E experts can be combined using a product rule:

$$f_j = \prod_{i=1}^{N_E} o_{i,j} \tag{6.2}$$

To account for varying discrimination ability and reliability of each expert, the above method is modified by introducing weighting:

$$f_j = \prod_{i=1}^{N_E} (o_{i,j})^{w_i} \tag{6.3}$$

The weighted product approach is also known as the product rule [5, 87]. There are two downsides to weighted product fusion: (i) one expert can have a large influence over the fused opinion - for example, an opinion close to zero from one expert sets the fused opinion also close to zero; (ii) the independence assumption is only strictly valid when each expert is using independent features.

Post-Classifier

Since the opinions produced by the experts indicate the "likelihood" of a particular class, the opinions can be considered as features in "likelihood space". The opinions from N_E experts regarding N_C classes form a $N_E N_C$-dimensional opinion vector, which is used by a classifier to make the final decision. We shall refer to such a classifier as a post-classifier[3]. It must be noted that the opinions do not necessarily have to be commensurate, as it is the post-classifier's job to provide adequate mapping from the "likelihood space" to class label space.

The obvious downside of this approach is that the resultant dimensionality of the opinion vector is dependent on the number of experts as well as the number of classes, which can be quite large in some applications. However, in a verification application, the dimensionality of the opinion vector is usually only dependent on the number of experts [22]. Each expert provides only one opinion, indicating the likelihood that a given claimant is the true claimant. The post-classifier then provides a decision boundary in N_E-dimensional space, separating the impostor and true claimant classes[4].

Special Case of Equivalence of Weighted Summation and Post-Classifier Approaches

In a normal verification application, there are only two classes (i.e. true claimants and impostors) and each expert provides only one opinion. Once the fused score is obtained using the weighted summation approach the accept/reject decision can be reached as follows: given a threshold t, the claim is accepted when $f \geq t$ (i.e. true claimant). The claim is rejected when $f < t$ (i.e. impostor). Eqn. (6.1) can thus be modified to:

$$F(\boldsymbol{o}) = \boldsymbol{w}^T \boldsymbol{o} - t \tag{6.4}$$

where $\boldsymbol{w}^T = [\ w_i\ ]_{i=1}^{N_E}$ and $\boldsymbol{o}^T = [\ o_i\ ]_{i=1}^{N_E}$. The decision is accordingly modified to: the claim is accepted when $F(\boldsymbol{o}) \geq 0$. The claim is rejected when $F(\boldsymbol{o}) < 0$.

It can be seen that Eqn. (6.4) is a form of a linear discriminant function [46], indicating that the procedure of weighted summation followed by thresholding creates a linear decision boundary in N_E-dimensional space. Thus in the verification application, weighted summation fusion is equivalent to a post-classifier which uses a linear decision boundary to separate the true claimant and impostor classes.

6.2.6 Hybrid Fusion

For certain applications, it may be necessary to combine various fusion techniques due to practical considerations. For example, Hong and Jain [72] used a fingerprint expert and a frontal face expert. A hybrid fusion scheme involving a ranked list and opinion fusion was used: opinions of the face expert for the top n identities were combined with the opinions of the fingerprint expert for the

[3] In the identification scenario, the described post-classifier is a natural extension of the approach presented in [6]. In the verification scenario it has been implemented by Ben-Yacoub et al. [22] as a binary classifier.

[4] see Figure 6.6 for example decision boundaries.

corresponding identities using a form of the product approach. This hybrid approach was used to take into account the relative computational complexity of the fingerprint expert, which was considerably slower than the face expert.

6.3 Milestones in Audio-Visual Person Recognition

This section provides an overview of the most important contributions in the field of audio-visual person recognition. It is assumed that the reader is familiar with the concepts presented in Section 6.2. We concentrate on the verification task while briefly touching on the identification task. Almost all of the work reviewed here used different datasets and/or different experiment setups (e.g. experts and performance measures), thus any direct comparison between the numerical results would be meaningless. Numerical figures are only shown in the first few cases to demonstrate that using fusion increases performance. Moreover, no thorough description of the various experts used is provided, as it is beyond the scope of this section.

The overview is split into two areas: non-adaptive (Section 6.3.1) and adaptive (Section 6.3.2) approaches. In non-adaptive approaches, the contribution of each expert is fixed a-priori. In adaptive approaches, the contribution of at least one expert is varied according to its reliability and discrimination ability in the presence of some environmental condition. For example, the contribution of a speech expert can be decreased when the audio SNR is lowered.

6.3.1 Non-Adaptive Approaches

Fusion of audio and visual information has been applied to automatic person recognition in pioneering papers by Chibelushi et al. [38] in 1993 and Brunelli et al. [29, 30] in 1995.

Chibelushi et al. [38] combined information from speech and still face profile images using a form of weighted summation fusion:

$$f = w_1 o_1 + w_2 o_2 \tag{6.5}$$

where o_1 and o_2 are the opinions from the speech and face profile experts, respectively, with corresponding weights w_1 and w_2. Each opinion reflects the likelihood that a given claimant is the true claimant. Since there are constraints on the weights ($\sum_{i=1}^{2} w_i = 1$ and $\forall i : w_i \geq 0$), Eqn. (6.5) reduces to:

$$f = w_1 o_1 + (1 - w_1) o_2 \tag{6.6}$$

The verification decision was reached via thresholding the fused opinion, f. When using the speech expert alone (i.e. $w_1 = 1$), an Equal Error Rate (EER) of 3.4% was achieved, while when using the face profile expert alone (i.e. $w_1 = 0$), an EER of 3.0% was obtained. Using an optimal weight and threshold (in the EER sense) the EER was reduced to 1.5%.

Brunelli et al. [29] combined the opinions from a face expert (which utilised geometric features obtained from static frontal face images) and a speech expert using the weighted product approach:

$$f = (o_1)^{w_1} \times (o_2)^{(1-w_1)} \tag{6.7}$$

When the speech expert was used alone (i.e. $w_1 = 1$), an identification rate of 51% was obtained, while when the face expert was used alone (i.e. $w_1 = 0$), an identification rate of 92% was achieved. Using an optimal weight, the identification rate increased to 95%.

In [30], two speech experts (for static and delta features) and three face experts (for the eye, nose and mouth areas of the face) were used for person identification. The weighted product approach was used to fuse the opinions, with the weights found automatically through a heuristic approach. The static and dynamic feature experts obtained an identification rate of 77% and 71%, respectively. Combining the two speech experts increased the identification rate to 88%. The eye, nose and mouth experts obtained an identification rate of 80%, 77% and 83%, respectively. Combining the three facial experts increased the identification rate to 91%. When all five experts were used, the identification rate increased to 98%.

Dieckmann et al. [43] used three experts (frontal face expert, dynamic lip image expert and text-dependent speech expert). A hybrid fusion scheme involving majority voting and opinion fusion was used. Two of the experts had to agree on the decision and the combined opinion had to exceed a pre-set threshold. The hybrid fusion scheme provided better performance than using the underlying experts alone.

Kittler et al. [86] used one frontal face expert which provided one opinion for one face image. Multiple images of one person were used to generate multiple opinions, which were then fused by various means, including averaging (a special case of weighted summation fusion). It was shown that error rates were reduced by up to 40% and that performance gains tended to saturate after using five images. The results suggest that using a video sequence of the face, rather than one image, provides superior performance. In further work, Kittler et al. [87] provided theoretical foundations for common fusion approaches such as the summation and product methods. However, the authors also note that the underlying assumptions can be unrealistic. Empirical results for combining the opinions from three experts (two face experts (frontal and profile) and a text-dependent speech expert) showed that the summation approach outperformed the product approach.

Luettin [100] investigated the combination of speech and (visual) lip information using feature vector concatenation. In order to match the frame rates of both feature sets, speech information was extracted at 30 fps instead of the usual 100 fps. In text-dependent configuration, the fusion process resulted in a minor performance improvement, however, in text-independent configuration, the performance slightly decreased. This suggests that feature vector concatenation in this case is unreliable.

Jourlin et al. [81, 82] used a form of weighted summation fusion to combine the opinions of two experts: a text-dependent speech expert and a text-dependent lip expert. Using an optimal weight, fusion led to better performance than using the underlying experts alone.

Abdeljaoued [1] proposed to use a Bayesian post-classifier to reach the verification decision. Formally, the decision rule is expressed as:

$$\text{chosen class} = \begin{cases} C_1 & \text{if} \quad \prod_{i=1}^{N_E} p(o_i|\lambda_{i,\text{true}}) > \prod_{i=1}^{N_E} p(o_i|\lambda_{i,\text{imp}}) \\ \\ C_2 & \text{otherwise} \end{cases} \tag{6.8}$$

where C_1 and C_2 are true claimant and impostor classes, respectively, N_E is the number of experts, while $\lambda_{i,\text{true}}$ and $\lambda_{i,\text{imp}}$ are, for the i-th expert, the parametric models of the distribution of opinions for true claimant and impostor claims, respectively[5]. Due to precision issues in a computational implementation, it is more convenient to use a summation rather than a series of multiplications. Since $\log(\cdot)$ is a monotonically increasing function, the decision rule can be modified to:

$$\text{chosen class} = \begin{cases} C_1 & \text{if} \quad \sum_{i=1}^{N_E} \log p(o_i|\lambda_{i,\text{true}}) > \sum_{i=1}^{N_E} \log p(o_i|\lambda_{i,\text{imp}}) \\ \\ C_2 & \text{otherwise} \end{cases} \tag{6.9}$$

To allow adjustment of FAR and FRR, the above decision rule is in practice modified by introducing a threshold:

$$\text{chosen class} = \begin{cases} C_1 & \text{if} \quad \sum_{i=1}^{N_E} \log p(o_i|\lambda_{i,\text{true}}) - \sum_{i=1}^{N_E} \log p(o_i|\lambda_{i,\text{imp}}) > t \\ \\ C_2 & \text{otherwise} \end{cases} \tag{6.10}$$

Abdeljaoued used three experts and showed that use of the above classifier (with Beta distributions) provided lower error rates than when using the experts alone.

Ben-Yacoub et al. [22] investigated the use of several binary classifiers for opinion fusion using a post-classifier. The investigated classifiers were: Support Vector Machine (SVM), Bayesian classifier (using Beta distributions), Fisher's Linear Discriminant, Decision Tree and Multi Layer Perceptron (MLP). Three experts were used: a frontal face expert and two speech based experts (text-dependent and text-independent). It was found that the SVM classifier (using a polynomial kernel) and the Bayesian classifier provided the best results.

Verlinde [179] also investigated various binary classifiers for opinion fusion as well as the majority voting and the and & or fusion methods (which fall in the decision fusion category). Three experts were used: frontal face expert, face profile expert and a text-independent speech expert. In the case of decision fusion, each expert acted like a classifier and provided a hard decision rather than an opinion. The investigated classifiers were: Decision Tree, MLP, Logistic Regression (LR) based classifier, Bayesian classifier using Gaussian distributions, Fisher's Linear Discriminant and various forms of the k-Nearest Neighbour classifier. Verlinde found that the LR based classifier (which created a linear decision surface) provided the lowest overall error rates as well as being the easiest to train.

[5] In our experiments we use Gaussian Mixture Models to model the distribution of opinions. See Section 6.4.2 for more information.

Wark et al. [181] used the weighted summation approach to combine the opinions of a speech expert and a lip expert (both text-independent). The performance of the speech expert was deliberately decreased by adding varying amounts of white noise to speech data (where the SNR varied from 50 to 10 dB). Empirical results showed that although the performance of the system was always better than using the speech expert alone, it significantly decreased as the noise level increased. Depending on the values of the weights (which were selected a-priori), the performance in high noise levels was actually worse than using the lip expert alone (a condition referred to as catastrophic fusion [184]). The authors proposed a statistically inspired method of selecting weights a-priori (described below) which resulted in good performance in clean conditions and never fell below the performance of the lip expert in noisy conditions. However, the performance in noisy conditions was shown not to be optimal. Moreover, the performance (for each noise level) was found using only 30 true claimant tests and 210 impostor tests.

The weight for the speech expert was found as follows:

$$w_1 = \frac{\zeta_2}{\zeta_1 + \zeta_2} \tag{6.11}$$

where

$$\zeta_i = \sqrt{\frac{\sigma^2_{i,true}}{N_{true}} + \frac{\sigma^2_{i,imp}}{N_{imp}}} \tag{6.12}$$

where, for the i-th expert, ζ_i is the standard error [32] of the difference between sample means $\mu_{i,true}$ and $\mu_{i,imp}$ of opinions for true and impostor claims, respectively, $\sigma^2_{i,true}$ and $\sigma^2_{i,imp}$ are the corresponding variances, while N_{true} and N_{imp} is the number of opinions for true and impostor claims, respectively. Wark et al. referred to ζ_i as an a-priori confidence. Since there are constraints on the weights ($\sum_{i=1}^{2} w_i = 1$ and $\forall i : w_i \geq 0$), the weight for the lip expert is $1 - w_1$.

Wark et al. assumed that the standard error gives relative indication of the discrimination ability of an expert. The less variation there is in the opinions for known true and impostor claims, the lower the standard error. Hence a low standard error indicates better performance.

Multi-Stream Hidden Markov Models (MS-HMMs) (a form of midst-mapping fusion) were evaluated for the task of text-dependent audio-visual person identification in [183]. The audio stream was comprised of a sequence of vectors containing Mel Frequency Cepstral Coefficients (MFCCs) [145] and their deltas [165] (see Chapter 3), while the video stream was comprised of a sequence of feature vectors describing lip contours. Due to the nature of the MS-HMM implementation the frame rate of the video features had to match the frame rate of the audio features (accomplished by up-sampling). Experiments on a small audio-visual dataset showed that for high SNRs the performance was comparable to that of an audio-only HMM system (which outperformed the video-only HMM system), while at low SNRs the multi-stream system obtained considerably better performance than the audio-only system and exceeded the performance of the video-only system.

Bengio [19] addressed several limitations of previous MS-HMM systems, allowing the two streams to be temporarily desynchronised (since related events in the streams may start and/or

end at different points, e.g. lip movement can start before speech is heard) and have different frame rates (thus up-sampling is no longer required). Experiments on a small audio-visual dataset (using two feature streams similar to the audio and video streams described for [183], above) showed that while at a relatively high SNR the performance was worse than a text-independent audio-only system, the performance was better at lower SNRs. Moreover, the proposed system had higher performance (and was more robust) than a text-dependent HMM system based on feature vector concatenation.

6.3.2 Adaptive Approaches

Wark et al. [182] extended the work presented in [181] (see above) by proposing a heuristic method to adjust the weights. Empirical results showed that although the performance significantly decreased as the noise level increased, it was always better than using the speech expert alone. However, in high noise levels, equal weights (non-adaptive) were shown to provide better performance. A major disadvantage of the method is that the calculation of the weights involved finding the opinion of the speech expert for all possible claims (i.e. for all persons enrolled in the system), thus limiting the approach to systems with a small number of clients due to practical considerations (i.e. time taken to verify a claim). Moreover, similar experimental limitations were present as described for [181] (above).

In further work [184], Wark proposed another heuristic technique of weight adjustment (described below). In a text-dependent configuration, the system provided performance which was always better than using the lip expert alone. However, in a text-independent configuration, the performance in low SNR conditions was worse than using the lip expert alone.

The weight for the speech expert was found as follows:

$$w_1 = \left[\frac{\zeta_2}{\zeta_1 + \zeta_2}\right]\left[\frac{\kappa_1}{\kappa_1 + \kappa_2}\right] \tag{6.13}$$

where $\frac{\zeta_2}{\zeta_1+\zeta_2}$ was found using Eqn. (6.12) during training and

$$\kappa_i = \frac{|\mathcal{M}(o_i)_{i,true} - \mathcal{M}(o_i)_{i,imp}|}{\mu_{i,true}} \tag{6.14}$$

was found during testing. Wark referred to κ_i as the posterior confidence. For the i-th expert, $\mathcal{M}(o_i)_{i,true} = \frac{(o_i-\mu_{i,true})^2}{\sigma^2_{i,true}}$ is the one dimensional squared Mahalanobis distance [46] between opinion o_i and the model of opinions for true claims. Here, $\mu_{i,true}$ and $\sigma^2_{i,true}$ are the mean and variance of opinions for true claims, respectively; they are found during training. Similarly, $\mathcal{M}(o_i)_{i,imp} = \frac{(o_i-\mu_{i,imp})^2}{\sigma^2_{i,imp}}$ is the distance between opinion o_i and the model of opinions for impostor claims. Here, $\mu_{i,imp}$ and $\sigma^2_{i,imp}$ are the mean and variance of opinions for impostor claims, respectively.

Under clean conditions, the distance between a given opinion for a true claim and the model of opinions for true claims should be small. Similarly, the distance between a given opinion for a

true claim and the model of opinions for impostor claims should be large. Vice versa applies for a given opinion for an impostor claim. Hence under clean conditions, κ_i should be large. Wark used empirical evidence to argue that under noisy conditions, the distances should decrease, hence κ_i should decrease.

In [160] a weight adjustment method was proposed that can be summarised as follows. Every time a speech utterance is recorded, it is usually preceded by a short segment which contains only ambient noise. From each training utterance, Mel Frequency Cepstral Coefficients (MFCCs) [135, 145] from the noise segment are used to construct a global noise Gaussian Mixture Model (GMM), λ_{noise}. Given a test speech utterance, N_{noise} MFCC feature vectors, $\{\boldsymbol{x}_i\}_{i=1}^{N_{\text{noise}}}$, representing the noise segment, are used to estimate the utterance's quality by measuring the mismatch from λ_{noise} as follows:

$$q = \frac{1}{N_{\text{noise}}} \sum_{i=1}^{N_{\text{noise}}} \log p(\boldsymbol{x}_i|\lambda_{\text{noise}}) \tag{6.15}$$

The larger the difference between the training and testing conditions, the lower q is going to be. The value of q is then mapped to the $[0, 1]$ interval using a sigmoid:

$$q_{\text{map}} = \frac{1}{1 + \exp[-a(q - b)]} \tag{6.16}$$

where a and b describe the shape of the sigmoid. The values of a and b are manually selected so that q_{map} is close to one for clean training utterances and close to zero for training utterances artificially corrupted with noise. As such, this adaptation method is dependent on the noise type that caused the mismatch.

Let us assume that the face expert is the first expert and that the speech expert is the second expert. Given an a-priori weight $w_{2,\text{prior}}$ for the speech expert (which is found on clean data [to achieve, for example, EER performance]), the adapted weight for the speech expert is found using:

$$w_2 = q_{\text{map}} w_{2,\text{prior}} \tag{6.17}$$

For a two modal system the corresponding weight for the face expert is found using $w_1 = 1 - w_2$. We shall refer to this weight adjustment method as the mismatch detection method.

6.4 Performance of Non-Adaptive Approaches in Noisy Conditions

In this section we evaluate the performance of feature vector concatenation fusion and several non-adaptive opinion fusion methods (weighted summation fusion, Bayesian and SVM post-classifiers), for combining face and speech information under the presence of audio noise.

6.4.1 VidTIMIT Audio-Visual Dataset

The VidTIMIT dataset [160] is comprised of video and corresponding audio recordings of 43 people (19 female and 24 male), reciting short sentences selected from the NTIMIT corpus [80]. It was

recorded in 3 sessions, with a mean delay of 7 days between Session 1 and 2, and 6 days between Session 2 and 3.

There are 10 sentences per person. The first six sentences are assigned to Session 1. The next two sentences are assigned to Session 2 with the remaining two to Session 3. The first two sentences for all persons are the same, with the remaining eight generally different for each person. The mean duration of each sentence is 4.25 seconds, or approximately 106 video frames. The recording was done in a noisy office environment using a broadcast quality digital video camera. Appendix A contains more detailed description of the dataset.

6.4.2 Speech Expert

The speech expert is comprised of two main components: speech feature extraction and a Gaussian Mixture Model (GMM) opinion generator. The latter is described in detail in Chapter 2, while the former in Chapter 3. For convenience we summarise the speech expert as follows.

The speech signal is analysed on a frame by frame basis, with a typical frame length of 20 ms and a frame advance of 10 ms. For each frame, a 37-dimensional feature vector is extracted, comprised of Mel Frequency Cepstral Coefficients (MFCC), which reflect the instantaneous Fourier spectrum [135, 145], their corresponding deltas (which represent transitional spectral information) [165] and Maximum Auto-Correlation Values (which represent pitch and voicing information) [189]. Cepstral mean subtraction was applied to MFCCs [50, 145]. The sequence of feature vectors is then processed by a parametric Voice Activity Detector (VAD) [64, 65], which removes feature vectors that are considered to represent silence or background noise.

The distribution of feature vectors for each person is modelled by a GMM. Given a claim for person C's identity and a set of feature vectors $X = \{\boldsymbol{x}_i\}_{i=1}^{N_V}$ supporting the claim, the average log-likelihood of the claimant being the true claimant is found with:

$$\mathcal{L}(X|\lambda_C) = \frac{1}{N_V}\sum_{i=1}^{N_V}\log p(\boldsymbol{x}_i|\lambda_C) \tag{6.18}$$

where

$$p(\boldsymbol{x}|\lambda) = \sum_{j=1}^{N_G} m_j\, \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \tag{6.19}$$

$$\lambda = \{m_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}_{j=1}^{N_G} \tag{6.20}$$

Here λ_C is the parameter set[6] for client C, N_G is the number of Gaussians, m_j is the weight for Gaussian j (with constraints $\sum_{j=1}^{N_G} m_j = 1$ and $\forall\, j : m_j \geq 0$). $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multi-variate Gaussian function with mean $\boldsymbol{\mu}$ and diagonal covariance matrix $\boldsymbol{\Sigma}$:

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] \tag{6.21}$$

[6] For convenience, we use the terms parameter set and model interchangeably.

where D is the dimensionality of $\boldsymbol{x}$. Given the average log-likelihood of the claimant being an impostor, $\mathcal{L}(X|\lambda_{\overline{C}})$, an opinion on the claim is found using:

$$\mathcal{O}(X|\lambda_C, \lambda_{\overline{C}}) = \mathcal{L}(X|\lambda_C) - \mathcal{L}(X|\lambda_{\overline{C}}) \tag{6.22}$$

The verification decision is reached as follows: given a threshold t, the claim is accepted when $\mathcal{O}(X|\lambda_C, \lambda_{\overline{C}}) \geq t$ and rejected when $\mathcal{O}(X|\lambda_C, \lambda_{\overline{C}}) < t$. The opinion reflects the likelihood that a given claimant is the true claimant (i.e. a low opinion suggests that the claimant is an impostor, while a high opinion suggests that the claimant is the true claimant). In mono-modal systems, the opinion can be thresholded to achieve the final verification decision.

Estimation of Model Parameters (Training)

First, a Universal Background Model (UBM) is trained using the Expectation Maximisation (EM) algorithm [42, 46]. As it is a good representation of the general population [149], it is also used to find the average log-likelihood of the claimant being an impostor, i.e.:

$$\mathcal{L}(X|\lambda_{\overline{C}}) = \mathcal{L}(X|\lambda_{ubm}) \tag{6.23}$$

The parameters (λ) for each client model are then found by using the client's training data and adapting the UBM using a form of maximum a-posteriori (MAP) adaptation (see Section 2.4.2).

6.4.3 Face Expert

The face expert is similar to the speech expert, with the feature extraction method being the main difference. Here we use the common Principal Component Analysis (PCA) technique [174] (also known as eigenfaces), which is described in detail in Chapter 4. For convenience, we recap PCA based feature extraction as follows.

After the face is located, a geometrical normalisation is applied to account for varying distances to the camera. To find the face, we use template matching with several prototype faces of varying dimensions[7]. Using the distance between the eyes as a size measure, an affine transformation is used [59] to adjust the size of the image, resulting in the distance between the eyes to be the same for each person. Finally a 64×56 pixel (columns $\times$ rows) face window, containing the eyes and the nose (the most invariant face area to changes in the expression and hair style) is extracted from the image. The size normalised face image is represented by a matrix containing grey level pixel values. The matrix is then converted to a face vector, $\boldsymbol{v}$, by concatenating all the columns. A D-dimensional feature vector, $\boldsymbol{x}$, is then obtained by:

$$\boldsymbol{x} = \boldsymbol{U}^T(\boldsymbol{v} - \boldsymbol{v}_\mu) \tag{6.24}$$

[7] A "mother" prototype face was constructed by averaging manually extracted and size normalised faces from clients (non-impostors) in the VidTIMIT dataset. Prototype faces of various sizes were constructed by applying an affine transform to the "mother" prototype face.

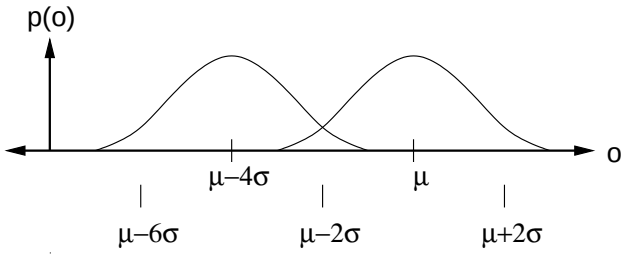

Figure 6.2: Graphical interpretation of the assumptions used in Section 6.4.4.

where $\boldsymbol{U}$ contains D eigenvectors (corresponding to the D largest eigenvalues) of the training data covariance matrix, and $\boldsymbol{v}_\mu$ is the mean of training face vectors. In our experiments we use training images from all clients (i.e. excluding impostors) find $\boldsymbol{U}$ and $\boldsymbol{v}_\mu$; moreover, $D = 20$. Preliminary experiments showed that while $D = 30$ obtained optimal face verification, the performance was not improved further with the use of fusion. Since we wish to evaluate how noisy audio conditions degrade fusion performance, we deliberately detuned the face expert so that fusion had a positive effect on performance in clean conditions.

We note that PCA is holistic in nature (that is, one face image yields one feature vector). However, a local feature based system (such as the DCT-based approach described in Chapter 4) can also be effectively used with the GMM opinion generator.

6.4.4 Mapping Opinions to the [0,1] Interval

The experiments reported throughout this chapter use the following method (inspired by [81]) of mapping the output of each expert to the $[0, 1]$ interval.

The original opinion of expert i, $o_{i,\text{orig}}$, is mapped to the $[0, 1]$ interval using a sigmoid:

$$o_i = \frac{1}{1 + \exp[-\tau_i(o_{i,\text{orig}})]} \tag{6.25}$$

where

$$\tau_i(o_{i,\text{orig}}) = \frac{o_{i,\text{orig}} - (\mu_i - 2\sigma_i)}{2\sigma_i} \tag{6.26}$$

where, for expert i, μ_i and σ_i are the mean and the standard deviation of original opinions for true claims, respectively. Assuming that the original opinions for true and impostor claims follow Gaussian distributions $\mathcal{N}(o_{i,\text{orig}}|\mu_i, {\sigma_i}^2)$ and $\mathcal{N}(o_{i,\text{orig}}|\mu_i - 4\sigma_i, {\sigma_i}^2)$ respectively, approximately 95% of the values lie in the $[\mu_i - 2\sigma_i \ , \ \mu_i + 2\sigma_i]$ and $[\mu_i - 6\sigma_i \ , \ \mu_i - 2\sigma_i]$ intervals, respectively [46] (see also Figure 6.2). Eqn. (6.26) maps the opinions to the $[-2, 2]$ interval, which corresponds to the approximately linear portion of the sigmoid in Eqn. (6.25). The sigmoid is necessary to take care of situations where the assumptions do not hold entirely.

6.4.5 Support Vector Machine Post-Classifier

The Support Vector Machine (SVM) [176] has been previously used by Ben-Yacoub et al. [22] as a post-classifier. While an in-depth description of SVM is beyond the scope of this section, important points are summarised. For more detail, the reader is referred to [31, 163].

The SVM is based on the principle of Structural Risk Minimisation (SRM) as opposed to Empirical Risk Minimisation (ERM) used in classical learning approaches. Under ERM, without testing on a separate data set, it is unknown which decision surface would have a good generalisation capability. For the case of the SVM, the decision surface has to satisfy a requirement which is thought to obtain the best generalisation capability. For example, let us assume we have a set of training vectors belonging to two completely separable classes and we seek a linear decision surface that separates the classes. Let us define the term margin as the sum of distances from the decision surface (in the space implied by the employed kernel, see below) to the two closest points from the two classes (one point from each class). We interpret the meaning of the margin as a measure of generalisation capability. Thus using the SRM principle, the optimal decision surface has the maximum margin.

The SVM (in its original form) is a binary classifier. Let us define a set S containing N_V opinion vectors (N_E-dimensional) belonging to two classes labelled as -1 and $+1$, indicating impostor and true claimant classes respectively:

$$S = \left\{ \ (\boldsymbol{o}_i, y_i) \mid \boldsymbol{o}_i \in \mathbb{R}^{N_E}, y_i \in \{-1, +1\} \ \right\}_{i=1}^{N_V} \tag{6.27}$$

The SVM uses the following function to map a given vector to its label space (i.e. -1 or $+1$):

$$f(\boldsymbol{o}) = \text{sign}\left(\sum_{i=1}^{N_V} \alpha_i y_i K(\boldsymbol{o}_i, \boldsymbol{o}) + b \right) \tag{6.28}$$

where vectors $\boldsymbol{o}_i$ with corresponding $\alpha_i > 0$ are known as support vectors (hence the name of the classifier). $K(\boldsymbol{d}, \boldsymbol{e})$ is a symmetric kernel function, subject to Mercer's condition [31, 176]. $\boldsymbol{\alpha}^T = [\alpha_i]_{i=1}^{N_V}$ is found by minimising (via quadratic programming):

$$-\sum_{i=1}^{N_V} \alpha_i + \frac{1}{2} \sum_{i=1}^{N_V} \sum_{j=1}^{N_V} \alpha_i \alpha_j y_i y_j K(\boldsymbol{o}_i, \boldsymbol{o}_j) \tag{6.29}$$

subject to constraints:

$$\boldsymbol{\alpha}^T \boldsymbol{y} = 0 \tag{6.30}$$

$$\alpha_i \in [0, C] \ \ \forall \ i \tag{6.31}$$

where, $\boldsymbol{y}^T = [\ y_i\ ]_{i=1}^{N_V}$ and C is a large positive value (e.g. 1000). C is used to allow training with non-separable data. The parameter b is found after $\boldsymbol{\alpha}$ has been found [31]. The kernel function $K(\boldsymbol{d}, \boldsymbol{e})$ can implement a dot product in a high (or possibly infinite) dimensional space, $\mathbb{R}^h$ (where $h \geq N_E$), which can improve separability of the data [157]. Note that the data is not explicitly

projected into the high dimensional space. Popular kernels used for pattern recognition problems are [31, 163]:

$$K(\boldsymbol{d}, \boldsymbol{e}) = \boldsymbol{d}^T \boldsymbol{e} \quad (6.32)$$

$$K(\boldsymbol{d}, \boldsymbol{e}) = (\boldsymbol{d}^T \boldsymbol{e} + 1)^p \quad (6.33)$$

$$K(\boldsymbol{d}, \boldsymbol{e}) = \exp(-\frac{1}{\sigma^2}||\boldsymbol{d} - \boldsymbol{e}||^2) \quad (6.34)$$

Eqn. (6.32) is a dot product, which is referred to as the linear kernel, Eqn. (6.33) is a p-th degree polynomial, while Eqn. (6.34) is a Gaussian kernel (where σ represents the standard deviation of the kernel).

The experiments reported here use the SVM engine developed by Joachims [78]. In a verification system there is generally more training data for the impostor class than the true claimant class. Thus a mis-classification on the impostor class (i.e. a FA error) has less contribution toward the EER than a misclassification on the true claimant class (i.e. a FR error). Hence standard SVM training, which in the non-separable case minimises the total misclassification rate, is not compatible with the EER criterion. Fortunately, Joachims' SVM engine allows setting of an appropriate cost of making an error on either class. While this does not explicitly guarantee training for EER, the cost can be tuned manually until performance close to EER is obtained.

6.4.6 Experiments

The experiments were done on the VidTIMIT dataset (see Section 6.4.1). The speech and frontal face experts are described in Sections 6.4.2 and 6.4.3, respectively. For the speech expert, best results on clean test data[8] were obtained with 32-Gaussian client models. For the face expert, best results were obtained with one-Gaussian client models.

Session 1 was used as the training data. To find the performance, Sessions 2 and 3 were used for obtaining expert opinions of known impostor and true claims. Four utterances, each from eight fixed persons (four male and four female), were used for simulating impostor accesses against the remaining 35 persons. For each of the remaining 35 persons, their four utterances were used separately as true claims. In total, there were 1120 impostor and 140 true claims.

In the first set of experiments, speech signals were corrupted by additive white Gaussian noise, with the resulting SNR varying from 12 to -8 dB. SNR of -8 dB was chosen as the end point as preliminary experiments showed that at this SNR the EER of the speech expert was close to chance level. In the second set of experiments, speech signals were corrupted speech signals were corrupted by adding "operations-room" noise from the NOISEX-92 corpus [177]. The "operations-room" noise contains background speech as well as machinery sounds. Again, the resulting SNR varied from 12 to -8 dB.

Performance of the following configurations was found: speech expert alone, face expert alone, feature vector concatenation, weighted summation fusion (equivalent to a post-classifier with a

[8] By clean data we mean original data which has not been artificially corrupted with noise.

linear decision boundary), the Bayesian post-classifier and the SVM post-classifier. For the latter three approaches, the face expert provided the first opinion (o_1) while the speech expert provided the second opinion (o_2) when forming the opinion vector $\boldsymbol{o} = [\ o_1\ o_2\ ]^T$.

The parameters for weighted summation fusion were found via an exhaustive search procedure. For the Bayesian post-classifier, two Gaussians were used to model the distribution of opinion vectors (one Gaussian each for true claimant and impostor distributions). Multiple Gaussians for each distribution, i.e. GMMs, were also evaluated but did not provide performance advantages. For the SVM post-classifier, the linear kernel [see Eqn. (6.32)] was used. Other kernels were also evaluated but did not provide performance advantages.

As described in Section 6.2.2, the basic idea of the feature vector concatenation is to concatenate the speech and face feature vectors to form a new feature vector. However, before concatenation can be done, the frame rates from the speech and face feature extractors must match. Recall that the frame rate for speech features is 100 fps while the standard frame rate for video is 25 fps (using off the shelf commercial PAL video cameras). A straightforward approach to match the frame rates is to artificially increase the video frame rate and generate the missing frames by copying original frames. It is also possible to decrease the frame rate of the speech features, but this would result in less speech information being available, decreasing performance [100]. Thus in the experiments reported in this section, the information loss is avoided by utilising the former approach of artificially increasing the video frame rate. As done by the speech expert, the feature vectors resulting from feature vector concatenation were processed by the VAD (Section 6.4.2). Best results on clean data were obtained with one-Gaussian client models.

The equivalency described in Section 6.2.5 has several implications on the measurement of performance of multi-expert systems. In speech based verification systems, the Equal Error Rate (EER) is often used as a measure of expected performance [44, 51]. In a single expert configuration this amounts to selecting the appropriate posterior threshold so that the False Acceptance Rate (FAR) is equal to the False Rejection Rate (FRR). In a multi-expert scenario this translates to selecting appropriate posterior parameters for opinion mapping (Section 6.4.4) and for the post-classifier (in the weighted summation case the parameters are $\boldsymbol{w}$ and t). In a multi-expert adaptive system, the weights are automatically tuned in an attempt to account the current reliability of one or more experts (as in the system proposed by Wark [184]). Tuning the threshold to obtain EER performance is equivalent to modifying one of the parameters of the post-classifier, which is in effect further adaptation of the post-classifier after observing the effect that the weights have on the distribution of f [Eqn. (6.1)] for true and impostor claims. Since this cannot be accomplished in real life, it is a fallacy to report the performance in noisy conditions in terms of EER for an adaptive multi-expert system.

Taking into account the above argumentation and to keep the presentation of results consistent between non-adaptive and adaptive systems, the results in this chapter are reported in the following manner. The post-classifier is tuned for EER performance on clean test data (analogous to the popular practice of using the posterior threshold in single-expert systems [44, 51]). Performance in

clean and noisy conditions is then reported in terms of Total Error (TE), defined as:

$$\mathrm{TE} = \mathrm{FAR} + \mathrm{FRR} \tag{6.35}$$

where the post-classifier parameters are fixed (in non-adaptive systems), or automatically varied (in adaptive systems). We note that posterior selection of parameters (for clean data) puts an optimistic bias on the results. However, since we wish to evaluate how noisy audio conditions degrade fusion performance, we would like to have an optimal starting point.

Performance of the face and speech experts is shown in Figure 6.3. Performance of the four multi-modal systems is shown in Figure 6.4 for white noise, and in Figure 6.5 for "operations-room" noise. Figures 6.6 and 6.7 show the distribution of opinion vectors in clean and noisy (SNR $= -8$ dB) conditions (white noise), respectively, with the decision boundaries used by the three post-classifier approaches.

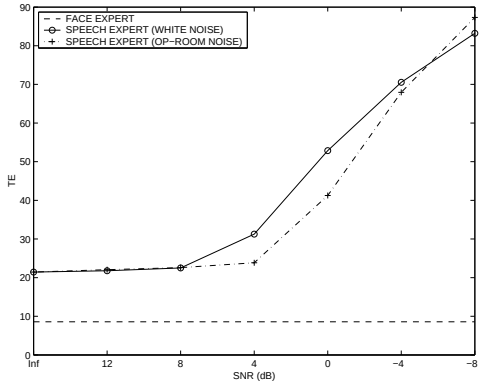

Figure 6.3: Performance of the speech and face experts.

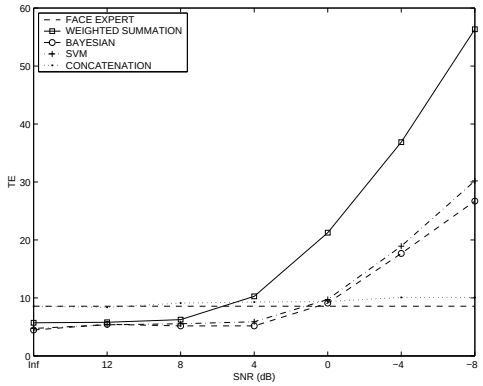

Figure 6.4: Performance of non-adaptive fusion techniques in the presence of white noise.

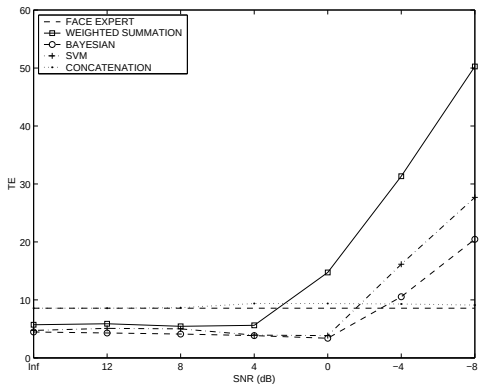

Figure 6.5: Performance of non-adaptive fusion techniques in the presence of operations-room noise.

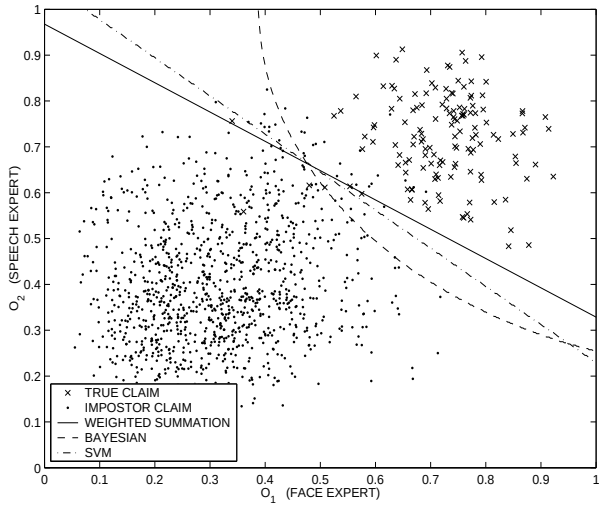

Figure 6.6: Decision boundaries used by fixed post-classifier fusion approaches and the distribution of opinion vectors for true and impostor claims (clean speech).

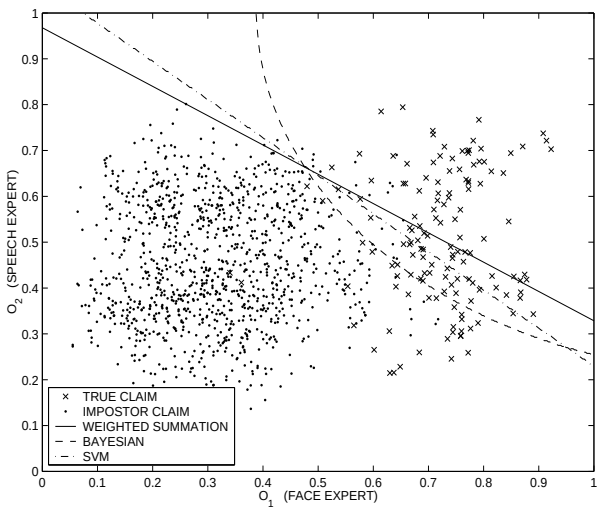

Figure 6.7: As per Figure 6.6, but using noisy speech (corrupted with white noise, SNR = -8 dB).

6.4.7 Discussion

Effect of Noisy Conditions on the Distribution of Opinion Vectors

For convenience, let us refer to the distribution of opinion vectors for true claims and impostor claims as the true claimant and impostor opinion distributions, respectively.

As can be observed in Figures 6.6 and 6.7, the main effect of noisy conditions is the movement of the mean of the true claim opinion distribution towards the o_1 axis. This movement can be partly explained by analysing Eqn. (6.22). Let us suppose a true claim has been made. In clean conditions $\mathcal{L}(X|\lambda_C)$ will be high while $\mathcal{L}(X|\lambda_{\overline{C}})$ will be low, causing o_2 (the opinion of the speech expert) to be high. When the speech expert is processing noisy speech signals, there is a mismatch between training and testing conditions, causing the feature vectors to drift away from the feature space described by the true claimant model (λ_C). This in turn causes $\mathcal{L}(X|\lambda_C)$ to decrease. If $\mathcal{L}(X|\lambda_{\overline{C}})$ decreases by the same amount as $\mathcal{L}(X|\lambda_C)$, then o_2 is relatively unchanged. However, as $\lambda_{\overline{C}}$ is a good representation of the general population, it usually covers a wide area of the feature space (see Section 6.4.2). Thus while the feature vectors may have drifted away from the space described by the true claimant model, they may still be inside the space described by the anti-client model, causing $\mathcal{L}(X|\lambda_{\overline{C}})$ to decrease by a smaller amount, which in turn causes o_2 to decrease.

Let us now suppose that several impostor claims have been made. In clean conditions $\mathcal{L}(X|\lambda_C)$ will be low while $\mathcal{L}(X|\lambda_{\overline{C}})$ will be high, causing o_2 to be low. The true claimant model does not represent the impostor feature space, indicating that $\mathcal{L}(X|\lambda_C)$ should be consistently low for impostor claims in noisy conditions. As mentioned above, $\lambda_{\overline{C}}$ usually covers a wide area of the feature space, thus even though the features have drifted due to mismatched conditions, the may still

be inside the space described by the anti-client model. This indicates that $\mathcal{L}(X|\lambda_{\overline{C}})$ should remain relatively high in noisy conditions, which in turn indicates that the impostor opinion distribution should change relatively little due to noisy conditions.

While Figures 6.6 and 6.7 show the effects of corrupting speech signals with additive white Gaussian noise, we have observed similar effects with the "operations-room" noise.

Effect of Noisy Conditions on Performance

In clean conditions, the weighted summation approach, SVM and Bayesian post-classifiers obtain performance better than either the face or speech expert. However, in high noise levels (SNR = -8 dB), all have performance worse than the face expert. This is expected since in all cases the decision mechanism uses fixed parameters.

All three approaches exhibit similar performance upto a SNR of 8 dB. As the SNR decreases further, the weighted summation approach is significantly more affected than the SVM and Bayesian post-classifiers. The differences in performance in noisy conditions can be attributed to the decision boundaries used by each approach, shown in Figures 6.6 and 6.7. It can be seen that the weighted summation approach has a decision boundary which results in the most mis-classifications of true claimant opinion vectors in noisy conditions.

The performance of the feature concatenation fusion approach is relatively more robust than the three post-classifier approaches. However, for most SNRs the performance is worse than the face expert, suggesting that while in this case feature concatenation fusion is relatively robust to the effects of noise, it is not optimal. The relatively poor performance in clean conditions can be attributed to the VAD. The entire speech signal was classified as containing speech instead of only the speech segments, thus providing a significant amount of irrelevant (non-discriminatory) information when modelling and calculating opinions.

Unlike the feature vectors obtained from the speech signal (which could contain either background noise or speech) each facial feature vector contained valid face information. Since the speech and facial vectors were concatenated to form one feature vector, the VAD could not distinguish between feature vectors containing background noise and speech. As stated previously, best results were obtained with one-Gaussian client models (compared to 32-Gaussian client models for the speech-only expert), suggesting that when more Gaussians were used, they were used for modelling the non-discriminatory information. Moreover, since one-Gaussian models are inherently less precise than 32-Gaussian models, we would expect them to be more robust to changes in distribution of feature vectors. Indeed the results suggest that this is occurring.

To address the relatively poor performance of the feature vector concatenation approach in clean conditions, it would hence be useful to explore the idea of concatenating only the audio vectors classified as speech (by the VAD) with the corresponding face vectors. This should aid in significantly reducing the amount of irrelevant (non-discriminative) information that is currently being used.

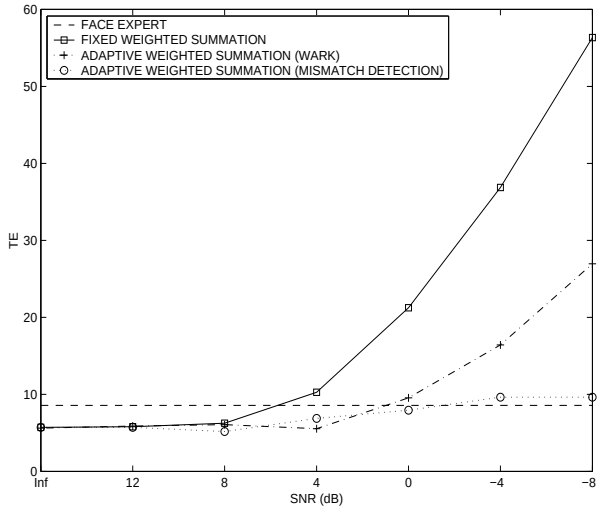

Figure 6.8: Performance of adaptive fusion techniques in the presence of white noise.

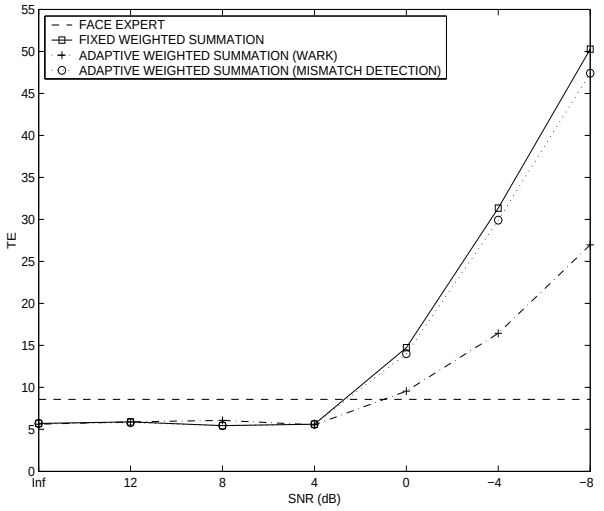

Figure 6.9: Performance of adaptive fusion techniques in the presence of operations-room noise.

6.5 Performance of Adaptive Approaches in Noisy Audio Conditions

In this section we evaluate the performance of several adaptive opinion fusion methods described in Section 6.3.2, namely weighted summation fusion with Wark's weight selection and the mismatch detection weight adjustment method.

The experiment setup is similar to the one described in Section 6.4.6. Based on manual observation of plots of speech signals from the VidTIMIT dataset, N_{noise} was set to 30 for the mismatch detection method [see Eqn. (6.15)]. One Gaussian for λ_{noise} was sufficient in preliminary experiments. The sigmoid parameters a and b [in Eqn. (6.16)] were obtained by observing how q in Eqn. (6.15) decreased as the SNR was lowered (using white Gaussian noise) on utterances in Session 1 (i.e. training utterances). The resulting value of q_{map} in Eqn. (6.16) was close to one for clean utterances and close to zero for utterances with an SNR of -8 dB.

Performance of the adaptive systems is shown in Figure 6.8 for white noise, and in Figure 6.9 for "operations-room" noise.

6.5.1 Discussion

Wark's weight selection approach assumes that under noisy conditions, the distance between a given opinion for an impostor claim and the corresponding model of opinions for impostor claims will decrease [see Eqn. (6.14)]. However, in experiments on the VidTIMIT dataset, the impostor distribution changed relatively little due to noisy conditions (as discussed in Section 6.4.7), thus Wark's posterior confidences (κ) for impostor claims changed relatively little as the SNR was lowered. However, Wark's approach appears to be more robust than the fixed weighted summation

approach. This is not due to the posterior confidences (κ), but due to the decision boundary being steeper from the start (thus being able to partially take into account the movement of opinion vectors due to noisy conditions). The nature of decision boundary was largely determined by the prior confidences (ζ) found with Eqn. (6.12).

For the case of white noise, when the mismatch detection weight adjustment method is used in the weighted summation approach, the performance gently deteriorates as the SNR is lowered, becoming slightly worse than the performance of the face expert at an SNR of -4 dB. For the case of "operations-room" noise, the mismatch detection method shows its limitation of being dependent on the noise type. The algorithm was configured to operate with white noise and was unable to handle the "operations-room" noise, resulting in performance very similar to the fixed (non-adaptive) approach.

6.6 Structurally Noise Resistant Post-Classifiers

Partly inspired by the SVM implementation of the SRM principle (see Section 6.4.5) and by the movement of opinion vectors due to presence of noise (see Section 6.4.7) a structurally noise resistant piece-wise linear (PL) post-classifier is developed (Section 6.6.1). As the name suggests, the decision boundary used by the post-classifier is designed so that the contribution of errors from the movement of opinion vectors is minimised. This is in comparison to standard post-classifier approaches, where the decision boundary is selected to optimise performance on clean data, with little or no regard to how the distributions of opinions may change due to noisy conditions. The Bayesian classifier presented in Section 6.3.1 is modified to introduce a similar structural constraint (Section 6.6.2). The performance of the two proposed post-classifiers is evaluated in Section 6.6.3.

6.6.1 Piece-Wise Linear Post-Classifier Definition

Let us describe the PL post-classifier as a discriminant function composed of two linear discriminant functions:

$$g(\boldsymbol{o}) = \begin{cases} a(\boldsymbol{o}) & \text{if } o_2 \geq o_{2,int} \\ \\ b(\boldsymbol{o}) & \text{otherwise} \end{cases} \tag{6.36}$$

where $\boldsymbol{o} = [\ o_1\ o_2\ ]^T$ is a two-dimensional opinion vector,

$$a(\boldsymbol{o}) = m_1 o_1 - o_2 + c_1 \tag{6.37}$$

$$b(\boldsymbol{o}) = m_2 o_1 - o_2 + c_2 \tag{6.38}$$

and $o_{2,int}$ is the threshold for selecting whether to use $a(\boldsymbol{o})$ or $b(\boldsymbol{o})$. Figure 6.10 shows an example of the decision boundary. The verification decision is reached as follows: the claim is accepted when $g(\boldsymbol{o}) \leq 0$ (i.e. true claimant) and rejected when $g(\boldsymbol{o}) > 0$ (i.e. impostor).

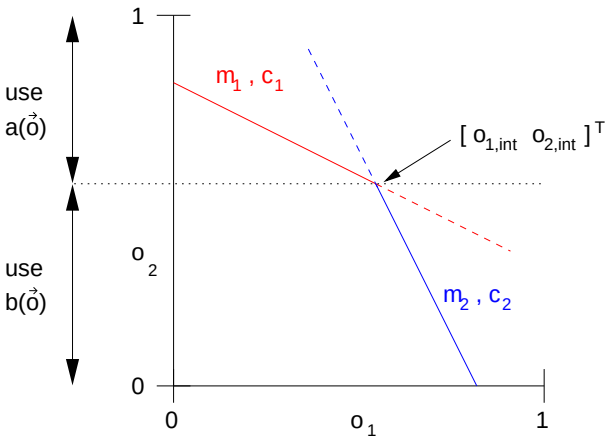

Figure 6.10: Example decision boundary of the PL classifier.

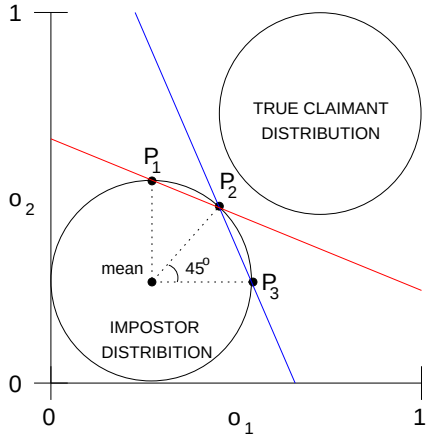

Figure 6.11: Points used in the initial solution of PL classifier parameters.

The first segment of the decision boundary can be described by $a(\boldsymbol{o}) = 0$, which reduces Eqn. (6.37) to:

$$o_2 = m_1 o_1 + c_1 \tag{6.39}$$

If we assume o_2 is a function of o_1, Eqn. (6.39) is simply the description of a line [168], where m_1 is the gradient and c_1 is the value at which the line intercepts the o_2 axis. Similar argument can be applied to the description of the second segment of the decision boundary. Given m_1, c_1, m_2 and c_2, we can find $o_{2,int}$ as follows. The two lines intersect at a single point $\boldsymbol{o}_{int} = [\ o_{1,int}\ o_{2,int}\ ]^T$. Moreover, when the two lines intersect, $a(\boldsymbol{o}_{int}) = b(\boldsymbol{o}_{int}) = 0$. Hence

$$o_{2,int} = m_1 o_{1,int} + c_1 = m_2 o_{1,int} + c_2 \tag{6.40}$$

which leads to:

$$o_{1,int} = \frac{c_1 - c_2}{m_2 - m_1} \tag{6.41}$$

$$o_{2,int} = m_2 \left(\frac{c_1 - c_2}{m_2 - m_1} \right) + c_2 \tag{6.42}$$

Structural Constraints and Training

As described in Section 6.4.7, the main effect of noisy conditions is the movement of opinion vectors for true claims toward the o_1 axis. We would like to obtain a decision boundary which minimises the increase of errors due to this movement. Structurally, this requirement translates to a decision boundary that is as steep as possible. Furthermore, to keep consistency with the experiments done in Sections 6.4 and 6.5, the classifier should be trained for EER performance. This in turn translates to the following constraints on the parameters of the PL classifier:

1. Both lines must exist in valid 2D opinion space (where the opinion from each expert is in the [0,1] interval) indicating that their intersect is constrained to exist in valid 2D opinion space.

2. Gradients for both lines need to be as large as possible (so the decision boundary that is as steep as possible).

3. The EER criterion must be satisfied.

Let $\lambda_{\text{PL}} = \{m_1, c_1, m_2, c_2\}$ be the set of PL classifier parameters. Given an initial solution, described in Section 6.6.1, the downhill simplex optimisation method [124, 139] can be used to find the final parameters. The following function is minimised:

$$\varepsilon(\lambda_{\text{PL}}) = \epsilon_1(\lambda_{\text{PL}}) + \epsilon_2(\lambda_{\text{PL}}) + \epsilon_3(\lambda_{\text{PL}}) \tag{6.43}$$

where $\epsilon_1(\lambda_{\text{PL}})$ through $\epsilon_3(\lambda)_{\text{PL}}$ (defined below) represent constraints 1-3 described above, respectively.

$$\epsilon_1(\lambda_{\text{PL}}) = \gamma_1 + \gamma_2 \tag{6.44}$$

$$\text{where } \gamma_j = \begin{cases} |o_{j,int}| & \text{if } o_{j,int} < 0 \text{ or } o_{j,int} > 1 \\ \\ 0 & \text{otherwise} \end{cases} \tag{6.45}$$

where $o_{1,int}$ and $o_{2,int}$ are found using Eqns. (6.41) and (6.42), respectively,

$$\epsilon_2(\lambda_{\text{PL}}) = \left|\frac{1}{m_1}\right| + \left|\frac{1}{m_2}\right| \tag{6.46}$$

and finally

$$\epsilon_3(\lambda_{\text{PL}}) = \left|\frac{\text{FAR}}{100\%} - \frac{\text{FRR}}{100\%}\right| \tag{6.47}$$

Initial Solution of PL Parameters

The initial solution for λ_{PL} (required by the downhill simplex optimisation) is based on the impostor opinion distribution. Let us assume that the distribution can be described by a 2D Gaussian function with a diagonal covariance matrix [see Eqn.(6.21)], indicating that it can be characterised by $\{\mu_1, \mu_2, \sigma_1, \sigma_2\}$ where μ_j and σ_j is the mean and standard deviation in the j-th dimension, respectively. Under the Gaussian assumption, 95% of the values for the j-th dimension lie in the $[\mu_j - 2\sigma_j, \mu_j + 2\sigma_j]$ interval. Let us use this property to define three points in 2D opinion space (shown graphically in Figure 6.11):

$$P_1 = (\ x_1\ ,\ y_1\ ) = (\ \mu_1\ ,\ \mu_2 + 2\sigma_2\ ) \tag{6.48}$$

$$P_2 = (\ x_2\ ,\ y_2\ ) = \left(\ \mu_1 + 2\sigma_1 \cos\left[\frac{\pi}{4}\right]\ ,\ \mu_2 + 2\sigma_2 \sin\left[\frac{\pi}{4}\right]\ \right) \tag{6.49}$$

$$P_3 = (\ x_3\ ,\ y_3\ ) = (\ \mu_1 + 2\sigma_1\ ,\ \mu_2\ ) \tag{6.50}$$

Thus the gradient (m_1) and the intercept (c_1) for the first line can be found using:

$$m_1 = \frac{y_2 - y_1}{x_2 - x_1} \tag{6.51}$$

$$c_1 = y_1 - m_1 x_1 \tag{6.52}$$

Similarly, the gradient (m_2) and the intercept (c_2) for the second line can be found using:

$$m_2 = \frac{y_3 - y_2}{x_3 - x_2} \tag{6.53}$$

$$c_2 = y_2 - m_2 x_2 \tag{6.54}$$

The initial solution for real data is shown in Figure 6.14.

6.6.2 Modified Bayesian Post-Classifier

In Figure 6.6 it can be seen that the decision boundary made by the Bayesian post-classifier (described in Section 6.3.1) envelops the true claimant opinion distribution. The downward movement of the vectors due to noisy conditions (discussed in Section 6.4.7) crosses the boundary and is the main cause of the error increases. If the decision boundary was forced to envelop the distribution of opinion vectors for impostor claims, the error increase would be reduced. This can be accomplished by modifying the decision rule described in (6.10) to use only the impostor likelihood (i.e. $\log p(o_i|\lambda_{i,\text{true}}) = 0 \ \ \forall\ i$):

$$\text{chosen class} = \begin{cases} C_1 & \text{if} \quad -\sum_{i=1}^{N_E} p(o_i|\lambda_{i,\text{imp}}) > t \\ \\ C_2 & \text{otherwise} \end{cases} \tag{6.55}$$

where C_1 and C_2 are the true claimant and impostor classes, respectively.

6.6.3 Experiments and Discussion

The performance of the proposed PL and modified Bayesian post-classifiers is evaluated. The experiment setup is the same as described in Section 6.4.6, with the results for white noise shown in Figure 6.12 and for "operations-room" noise in Figure 6.13. Figures 6.14 and 6.15 show the distribution of opinion vectors in clean and noisy (SNR = −8 dB) conditions (white noise), respectively, with the decision boundaries used by the proposed approaches.

As can be observed, the decision boundary used by the PL post-classifier effectively takes into account the movement of opinion vectors due to noisy conditions. Comparing Figures 6.8 and 6.12 it can be seen that the proposed PL post-classifier has similar performance to the adaptive weighted summation approach, with the advantage of having a fixed (non-adaptive) structure. Moreover, unlike the mismatch detection weight update algorithm used in the adaptive approach, the PL post-classifier does not make a direct assumption about the type of noise that caused the mismatch between training and testing conditions.

The performance of the modified Bayesian post-classifier is comparable to that of the PL post-classifier due to the nature of the decision boundary. However, unlike the PL post-classifier, the modified Bayesian post-classifier avoids heuristics, is easy to train and is easily extendable to three or more experts.

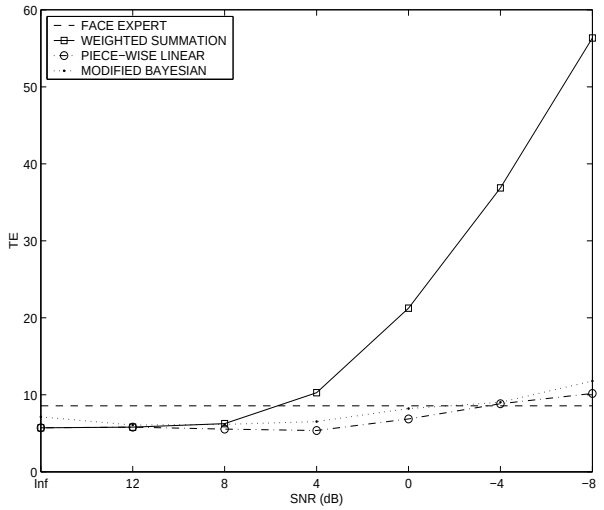

Figure 6.12: Performance of structurally noise resistant fusion techniques in the presence of white noise.

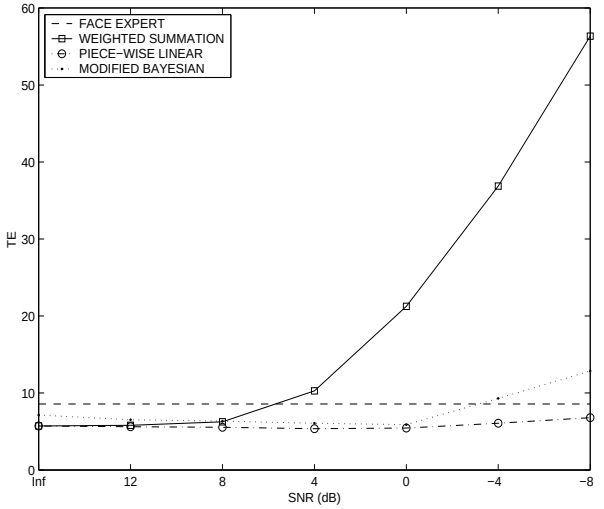

Figure 6.13: Performance of structurally noise resistant fusion techniques in the presence of operations-room noise.

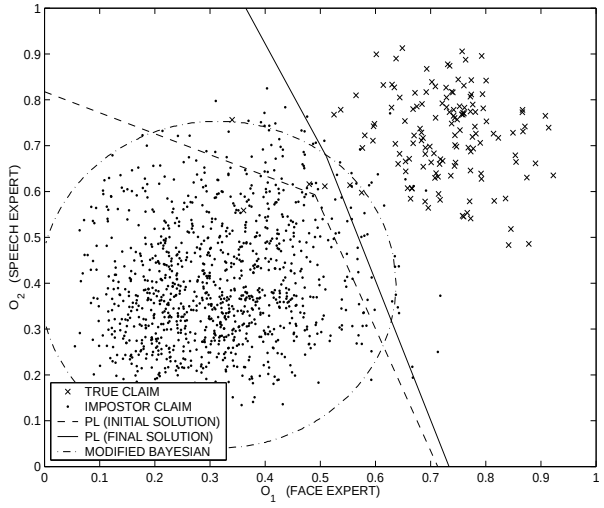

Figure 6.14: Decision boundaries used by structurally noise resistant fusion approaches and the distribution of opinion vectors for true and impostor claims (clean speech).

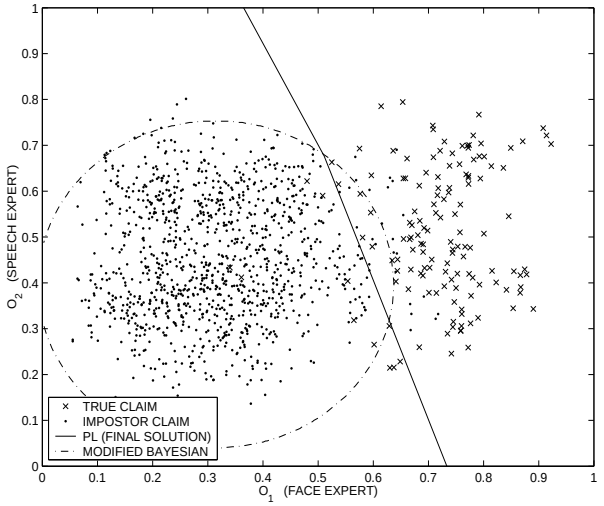

Figure 6.15: As per Figure 6.14, but using noisy speech (corrupted with white noise, SNR = −8 dB).

Reliably finding good parameters for the PL classifier through the downhill simplex optimisation algorithm can be an issue. We have found it to be sensitive to initial conditions – if the initial solution is not close to the optimum according to Eqn. (6.43), for example due to opinion vectors that are outliers, the optimisation can diverge. Eqn. (6.43) is itself a contributor to this instability, as it has no explicit smoothness constraints. The surface made by Eqn. (6.43) is likely to have many local minima.

One alternative approach would be simply to use the initial solution as the final solution. A more elaborate alternative would be to use an SVM with a specifically designed kernel [163] that takes into account the movement of opinion vectors, or the relative robustness of the underlying experts. For example, a compound kernel can be constructed from two underlying kernels [163], with each kernel processing opinions from a specific modality expert. The contribution of each modality can then be accomplished through appropriate weighting of the underlying kernels, in a manner similar to the weighted summation of opinions (Section 6.2.5).

6.7 Summary

This chapter provided an overview of important concepts in the field of information fusion, followed by a review of milestones in audio-visual person identification and verification. Several adaptive and non-adaptive techniques for reaching the verification decision (i.e. whether to accept or reject the claimant), based on combined speech and face information, were evaluated in clean and noisy audio conditions on a common dataset. It was shown that in clean conditions most of the non-adaptive approaches provide similar performance and in noisy conditions most exhibit deterioration in performance. Moreover, it was shown that current adaptive approaches are either inadequate or use restrictive assumptions. A new category of classifiers was then introduced, where the decision boundary is fixed but constructed to take into account how the distributions of opinions are likely to change due to noisy conditions. Compared to a previously proposed adaptive approach, the proposed classifiers do not make a direct assumption about the type of noise that causes the mismatch between training and testing conditions.

Appendix A

The VidTIMIT Dataset

Overview

In this appendix two previous multi-modal datasets, M2VTS and XM2VTS, are briefly described. The VidTIMIT dataset is then described, which was created by the author while taking into account issues with the abovementioned datasets.

M2VTS and XM2VTS datasets

At the start of research for this work, only one widely distributed multi-modal database existed, namely the M2VTS database [136]. The database is comprised of video sequences and corresponding audio recordings of 37 people counting '0' to '9' in their native language (mostly in French). There are five sessions per person (with one '0' to '9' utterance per session), spaced apart by at least one week. A head rotation sequence was also recorded during each session, where each person moved their head to the left and then to the right. The head rotation is meant to facilitate extraction of profile or 3D information.

The major drawbacks of the M2VTS database are its small size and the limited vocabulary (one phrase consisting of the '0' to '9' count). The small size results in several problems. The data set needs to be divided into at least 2 sections, representing the training and testing sections (typically, M2VTS sessions 1 to 3 are labelled as training data and session 4 as test data, with session 5 left out due to particular recording conditions). A small amount of training data can easily result in unreliable statistical models (as used in Chapter 2). A small test set results in a small number of verification tests, thus any relative improvement of one verification approach over another is dubious. Lastly, a verification method developed on the M2VTS database cannot be guaranteed to work in the more general text-independent mode, since the training phrase is the same as the testing phrase.

The Extended M2TVS (XM2VTS) dataset [110], released several years later, addresses some of these problems. The main differences are: 295 subjects, three fixed phrases (with two utterances of each phrase) and four sessions. The phrases are:

- "0 1 2 3 4 5 6 7 8 9"
- "5 0 6 9 2 8 1 3 7 4"
- "Joe took fathers green shoe bench out"

While the number of subjects results in a much larger number of verification tests, the database is more suited for testing of text-dependent verification systems. While it is possible to obtain a pseudo text-independent setup by training a system using only phrases 1 and 2 and testing it on phrase 3, the training data is not a good representation of the test data – this can lead to poor performance.

The VidTIMIT Dataset

The VidTIMIT dataset was created to address some of the issues present in the M2VTS and XM2VTS datasets. The dataset is comprised of video and corresponding audio recordings of 43 volunteers (19 female and 24 male), reciting short sentences. It was recorded in 3 sessions, with a mean delay of 7 days between Session 1 and 2, and 6 days between Session 2 and 3.

The delay between sessions allows for changes in the voice, hair style, make-up, clothing and mood (which can affect the pronunciation), thus incorporating attributes which would be present during the deployment of a verification system. Additionally, the zoom factor of the camera was randomly perturbed after each recording.

The sentences were chosen from the test section of the NTIMIT corpus [80]. There are ten sentences per person. The first six sentences (sorted alpha-numerically by filename) are assigned to Session 1. The next two sentences are assigned to Session 2 with the remaining two to Session 3. The first two sentences for all persons are the same, with the remaining eight generally different for each person. The mean duration of each sentence is 4.25 seconds, or approximately 106 video frames (using 25 fps).

A typical example[1] of the sentences used is in Table A.1. There is complete correspondence of the subject IDs between VidTIMIT and NTIMIT (and hence the recited sentences).

In addition to the sentences, each person performed an extended head rotation sequence in each session, which allows for extraction of profile and 3D information. The sequence consists of the person moving their head to the left, right, back to the centre, up, then down and finally return to centre. Examples images are shown in Figures A.1 and A.2.

The recording was done in a noisy office environment (mostly computer fan noise) using a broadcast quality digital video camera. The video of each person is stored as a numbered sequence of JPEG images with a resolution of 384 × 512 pixels (rows × columns). A quality setting of 90% was used during the creation of the JPEG images. The corresponding audio[2][3] is stored as a mono,

[1] Copyright restrictions on the NTIMIT corpus prevent the list of all sentences used in VidTIMIT.
[2] The audio was recorded using the camera's microphone.
[3] There is no audio for the head rotation sequences.

16 bit, 32 kHz WAV file. The entire dataset occupies approximately 3 Gb.

For almost all the recordings the lighting setup[4] was as follows:

1. Standard overhead fluorescent tubes, like in most ofce environments. The lights were covered with A4 size white office paper in order to diffuse the light – this reduced the glare on the face and top of the head.

2. An incandescent lamp in front of the person (just below the camera). The lamp was covered with a sheet of A4 size white office paper.

Session 1 is intended to be used as the training section, while Sessions 2 & 3 are intended to be the test section. It must be noted that unlike the M2VTS and XM2VTS databases, all sessions contain various phonetically balanced sentences. For each person, no sentences are repeated across the testing and training sections. The database is thus suited for the development of a text-independent verification system.

The number of subjects in the VidTIMIT database is somewhat larger than in the M2VTS database. However, while in the M2VTS database there is only one test utterance per person, there are four in the VidTIMIT database. Thus the number of verification tests possible on the VidTIMIT database is over 4 times larger than on the M2VTS database.

Section ID	Sentence ID	Sentence text
Session 1	sa1	She had your dark suit in greasy wash water all year
	sa2	Don't ask me to carry an oily rag like that
	si1398	Do they make class-biased decisions?
	si2028	He took his mask from his forehead and threw it, unexpectedly, across the deck
	si768	Make lid for sugar bowl the same as jar lids, omitting design disk
	sx138	The clumsy customer spilled some expensive perfume
Session 2	sx228	The viewpoint overlooked the ocean
	sx318	Please dig my potatoes up before frost
Session 3	sx408	I'd ride the subway, but I haven't enough change
	sx48	Grandmother outgrew her upbringing in petticoats

Table A.1: Typical example of sentences used in the VidTIMIT database

[4] An exception to the lighting setup occurs for Session 3 of mbdg0 and Session 2 of mjar0. For these cases the incandescent lamp was switched off.

Figure A.1: Example subjects from the VidTIMIT dataset. The first, second and third columns represent images taken in Session 1, 2 and 3, respectively.

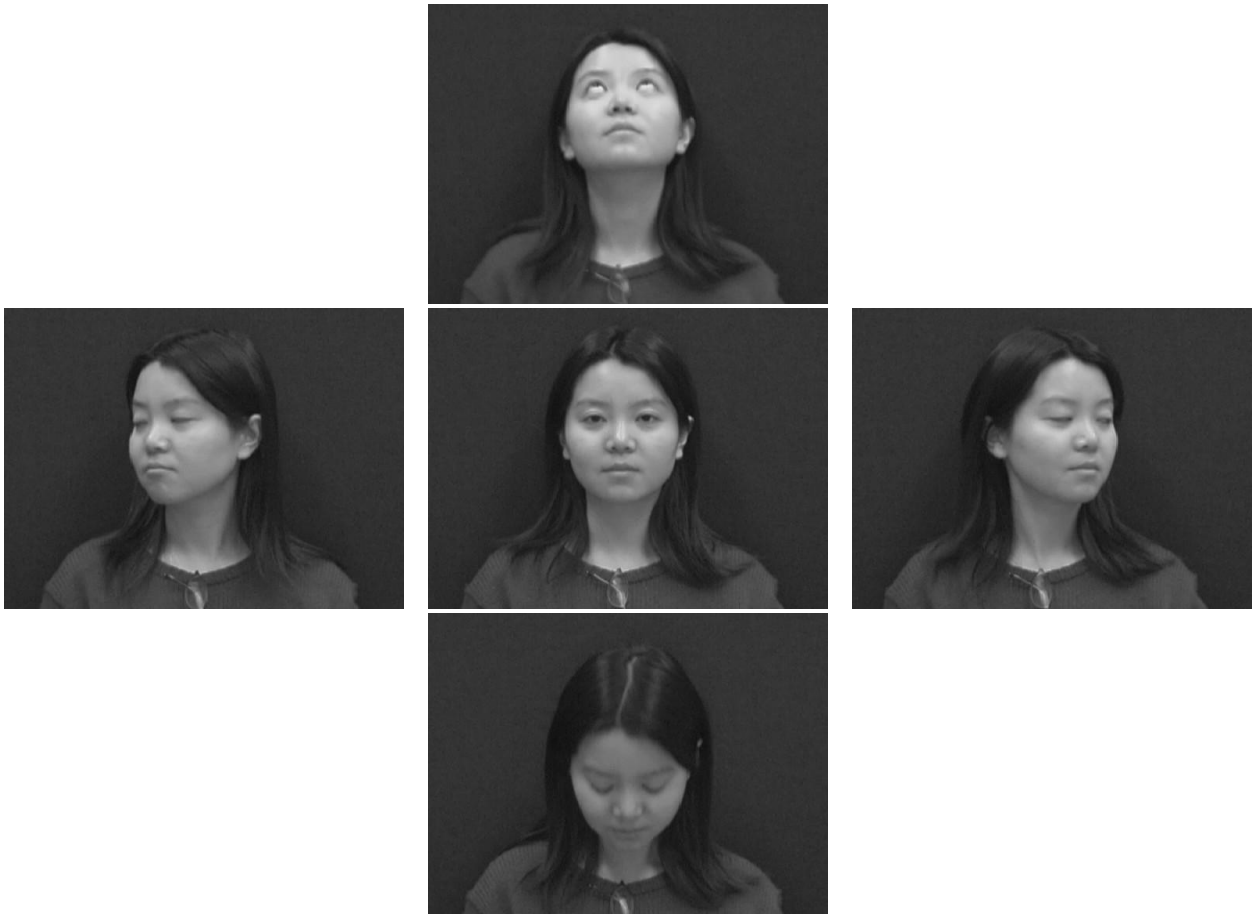

Figure A.2: Extracts from a head rotation sequence.

Appendix B

EM Algorithm for Gaussian Mixture Models

In the Gaussian Mixture Model (GMM) approach, a D-dimensional observation vector $\boldsymbol{x}$ is modelled by:

$$p(\boldsymbol{x}|\Theta) = \sum_{m=1}^{M} w_m p(\boldsymbol{x}|\theta_m) \tag{B.1}$$

where $\sum_{m=1}^{M} w_m = 1$, $w_m \geq 0$ and $p(\boldsymbol{x}|\theta_m)$ is a multivariate Gaussian probability density function with parameter set $\theta_m = \{\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}$:

$$p(\boldsymbol{x}|\theta_m) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = \frac{1}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}_m|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_m)^T\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_m)\right] \tag{B.2}$$

where $\boldsymbol{\mu}_m$ is the mean vector and Σ_m is the covariance matrix. Thus the complete parameter set for Eqn. (B.1) is expressed as $\Theta = \{w_m, \theta_m\}_{m=1}^{M}$. Our aim is to find Θ so the likelihood function

$$p(X|\Theta) = \prod_{i=1}^{N} p(\boldsymbol{x}_i|\Theta) \tag{B.3}$$

is maximised. Here, $X = \{\boldsymbol{x}_i\}_{i=1}^{N}$ is the set of training data.

The Expectation-Maximisation (EM) algorithm [42, 109, 143] is a likelihood function optimisation technique, often used in the pattern recognition literature [46]. It is a general method for finding the maximum-likelihood estimate of the parameters of an assumed distribution, when either the training data is incomplete or has missing values, or when the likelihood function can be made analytically tractable by assuming the existence of (and values for) missing data.

To apply the EM algorithm to our GMM problem, we must first assume that our training data X is incomplete and assume the existence of missing data $Y = \{y_i\}_{i=1}^{N}$, where the values of y_i indicate the mixture component that "generated" $\boldsymbol{x}_i$. Thus $y_i \in [1, M]\ \forall\ i$ and $y_i = m$ if the i-th feature vector ($\boldsymbol{x}_i$) was "generated" by the m-th mixture component. If we know the values for Y, then Eqn. (B.3) can be modified to:

$$p(X, Y|\Theta) = \prod_{i=1}^{N} w_{y_i} p(\boldsymbol{x}_i|\theta_{y_i}) \tag{B.4}$$

As its name suggests, the EM algorithm is comprised of two steps: expectation, followed by maximisation. In the expectation step, the expected value of the complete data log-likelihood, $\log p(X, Y|\Theta)$, is found with respect to the unknown data $Y = \{y_i\}_{i=1}^N$ given training data $X = \{\boldsymbol{x}_i\}_{i=1}^N$ and current parameter estimates, $\Theta^{[k]}$ (where k indicates the iteration number):

$$Q(\Theta, \Theta^{[k]}) = E\left[\log p(X, Y|\Theta) \mid X, \Theta^{[k]}\right] \tag{B.5}$$

Since Y is a random variable with distribution $p(\boldsymbol{y}|X, \Theta^{[k]})$, Eqn. (B.5) can be written as:

$$Q(\Theta, \Theta^{[k]}) = \int_{\boldsymbol{y}\in\Upsilon} \log p(X, \boldsymbol{y}|\Theta)\ p(\boldsymbol{y}|X, \Theta^{[k]})\ \ d\boldsymbol{y} \tag{B.6}$$

where $\boldsymbol{y}$ is an instance of the missing data and Υ is the space of values $\boldsymbol{y}$ can take on. The maximisation step then maximises the expectation:

$$\Theta^{[k+1]} = \arg\max_{\Theta} Q(\Theta, \Theta^{[k]}) \tag{B.7}$$

The expectation and maximisation steps are iterated until convergence, or when the increase in likelihood falls below a pre-defined threshold. As can be seen in Eqn. (B.6), we require $p(\boldsymbol{y}|X, \Theta^{[k]})$. We can define it as follows:

$$p(\boldsymbol{y}|X, \Theta^{[k]}) = \prod_{i=1}^{N} p(y_i|\boldsymbol{x}_i, \Theta^{[k]}) \tag{B.8}$$

Given initial parameters[1] $\Theta^{[k]}$, we can compute $p(\boldsymbol{x}_i|\theta_m^{[k]})$. Moreover, we can interpret the mixing weights (w_m) as a-priori probabilities of each mixture component [i.e., $w_m = p(m|\Theta^{[k]})$]. Hence we can apply Bayes' rule [46] to obtain:

$$p(y_i|\boldsymbol{x}_i, \Theta^{[k]}) = \frac{p(\boldsymbol{x}_i|\theta_{y_i}^{[k]}) p(y_i|\Theta^{[k]})}{p(\boldsymbol{x}_i|\Theta^{[k]})} \tag{B.9}$$

$$= \frac{p(\boldsymbol{x}_i|\theta_{y_i}^{[k]}) p(y_i|\Theta^{[k]})}{\sum_{n=1}^{M} p(\boldsymbol{x}_i|\theta_n^{[k]}) p(n|\Theta^{[k]})} \tag{B.10}$$

Expanding Eqn. (B.6) yields:

[1] Parameters for $k = 0$ can be found via the k-means algorithm [46, 97] (see also Section 2.4.1).

$$Q(\Theta, \Theta^{[k]}) = \int_{\boldsymbol{y}\in\Upsilon} \log p(X, \boldsymbol{y}|\Theta)\ p(\boldsymbol{y}|X, \Theta^{[k]})\ \ d\boldsymbol{y} \tag{B.11}$$

$$= \sum_{\boldsymbol{y}\in\Upsilon} \log \prod_{i=1}^{N} w_{y_i} p(\boldsymbol{x}_i|\theta_{y_i}) \prod_{j=1}^{N} p(y_j|\boldsymbol{x}_j, \Theta^{[k]}) \tag{B.12}$$

$$= \sum_{y_1=1}^{M} \sum_{y_2=1}^{M} \cdots \sum_{y_N=1}^{M} \sum_{i=1}^{N} \log\left[w_{y_i} p(\boldsymbol{x}_i|\theta_{y_i})\right] \prod_{j=1}^{N} p(y_j|\boldsymbol{x}_j, \Theta^{[k]}) \tag{B.13}$$

It can be shown [24] that Eqn. (B.13) can be simplified to:

$$Q(\Theta, \Theta^{[k]}) = \sum_{m=1}^{M} \sum_{i=1}^{N} \log[w_m p(\boldsymbol{x}_i|\theta_m)]\ p(m|\boldsymbol{x}_i, \Theta^{[k]}) \tag{B.14}$$

$$= \sum_{m=1}^{M} \sum_{i=1}^{N} \log[w_m]\ p(m|\boldsymbol{x}_i, \Theta^{[k]}) + \sum_{m=1}^{M} \sum_{i=1}^{N} \log[p(\boldsymbol{x}_i|\theta_m)]\ p(m|\boldsymbol{x}_i, \Theta^{[k]}) \tag{B.15}$$

$$= Q_1 \quad + \quad Q_2 \tag{B.16}$$

Hence Q_1 and Q_2 can be maximised separately, to obtain w_m and $\theta_m = \{\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}$, respectively. To find the expression which maximises w_m, we need to introduce the Lagrange multiplier [46] ψ, with the constraint $\sum_m w_m = 1$, take the derivative of Q_1 with respect to w_m and set the result to zero:

$$\frac{\partial Q_1}{\partial w_m} = 0 \tag{B.17}$$

$$\therefore\ 0 = \frac{\partial}{\partial w_m} \left\{ \sum_{m=1}^{M} \sum_{i=1}^{N} \log[w_m]\ p(m|\boldsymbol{x}_i, \Theta^{[k]}) + \psi \left[(\sum_m w_m) - 1 \right] \right\} \tag{B.18}$$

$$= \sum_{i=1}^{N} \frac{1}{w_m}\ p(m|\boldsymbol{x}_i, \Theta^{[k]}) + \psi \tag{B.19}$$

Let us rearrange Eqn. (B.19) so we can obtain a value for ψ:

$$-\psi w_m = \sum_{i=1}^{N} p(m|\boldsymbol{x}_i, \Theta^{[k]}) \tag{B.20}$$

Summing both sides over m yields:

$$-\psi \sum_m w_m = \sum_{i=1}^{N} \sum_m p(m|\boldsymbol{x}_i, \Theta^{[k]}) \tag{B.21}$$

$$-\psi 1 = \sum_{i=1}^{N} 1 \tag{B.22}$$

$$\psi = -N \tag{B.23}$$

By substituting Eqn. (B.23) into Eqn. (B.19) we obtain:

$$N = \sum_{i=1}^{N} \frac{1}{w_m}\, p(m|\boldsymbol{x}_i, \Theta^{[k]}) \tag{B.24}$$

$$\therefore\ w_m = \frac{1}{N} \sum_{i=1}^{N} p(m|\boldsymbol{x}_i, \Theta^{[k]}) \tag{B.25}$$

To find expressions which maximise $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}_m$, let us now expand Q_2:

$$Q_2 = \sum_{m=1}^{M} \sum_{i=1}^{N} \log[p(\boldsymbol{x}_i|\theta_m)]\ p(m|\boldsymbol{x}_i, \Theta^{[k]}) \tag{B.26}$$

$$= \sum_{m=1}^{M} \sum_{i=1}^{N} \left[-\frac{1}{2} \log(|\Sigma_m|) - \frac{1}{2} (\boldsymbol{x}_i - \boldsymbol{\mu}_m)^T \boldsymbol{\Sigma}_m^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_m) \right] p(m|\boldsymbol{x}_i, \Theta^{[k]}) \tag{B.27}$$

where $-\frac{D}{2}\log(2\pi)$ was omitted since it vanishes when taking a derivative with respect to $\boldsymbol{\mu}_m$ or $\boldsymbol{\Sigma}_m^{-1}$. To find the expression which maximises $\boldsymbol{\mu}_m$, we need to take the derivative of Q_2 with respect to $\boldsymbol{\mu}_m$, and set the result to zero:

$$\frac{\partial Q_2}{\partial \boldsymbol{\mu}_m} = 0 \tag{B.28}$$

$$0 = \frac{\partial}{\partial \boldsymbol{\mu}_m} \left\{ \sum_{m=1}^{M} \sum_{i=1}^{N} \left[-\frac{1}{2} \log(|\Sigma_m|) - \frac{1}{2} (\boldsymbol{x}_i - \boldsymbol{\mu}_m)^T \boldsymbol{\Sigma}_m^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_m) \right] p(m|\boldsymbol{x}_i, \Theta^{[k]}) \right\} \tag{B.29}$$

At this point let us recall some results from matrix theory. Lütkepohl [102] states that $\frac{\partial \boldsymbol{z}^T \boldsymbol{A} \boldsymbol{z}}{\partial \boldsymbol{z}} = (\boldsymbol{A} + \boldsymbol{A}^T)\boldsymbol{z}$, $(\boldsymbol{A}^{-1})^T = (\boldsymbol{A}^T)^{-1}$ and if $\boldsymbol{A}$ is symmetric, then $\boldsymbol{A} = \boldsymbol{A}^T$.

Since $\boldsymbol{\Sigma}_m$ is symmetric, Eqn. (B.29) reduces to:

$$0 = \sum_{i=1}^{N} -\frac{1}{2} 2\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_m) p(m|\boldsymbol{x}_i, \Theta^{[k]}) \tag{B.30}$$

$$= \sum_{i=1}^{N} \left[-\boldsymbol{\Sigma}_m^{-1}\boldsymbol{x}_i p(m|\boldsymbol{x}_i, \Theta^{[k]}) + \boldsymbol{\Sigma}_m^{-1}\boldsymbol{\mu}_m p(m|\boldsymbol{x}_i, \Theta^{[k]}) \right] \tag{B.31}$$

$$\therefore \sum_{i=1}^{N} \boldsymbol{\Sigma}_m^{-1}\boldsymbol{\mu}_m p(m|\boldsymbol{x}_i, \Theta^{[k]}) = \sum_{i=1}^{N} \boldsymbol{\Sigma}_m^{-1}\boldsymbol{x}_i p(m|\boldsymbol{x}_i, \Theta^{[k]}) \tag{B.32}$$

multiply both sides by $\boldsymbol{\Sigma}_m$:

$$\sum_{i=1}^{N} \boldsymbol{\mu}_m p(m|\boldsymbol{x}_i, \Theta^{[k]}) = \sum_{i=1}^{N} \boldsymbol{x}_i p(m|\boldsymbol{x}_i, \Theta^{[k]}) \tag{B.33}$$

$$\therefore \boldsymbol{\mu}_m = \frac{\sum_{i=1}^{N} \boldsymbol{x}_i p(m|\boldsymbol{x}_i, \Theta^{[k]})}{\sum_{i=1}^{N} p(m|\boldsymbol{x}_i, \Theta^{[k]})} \tag{B.34}$$

Lütkepohl [102] states that: $|\boldsymbol{A}^{-1}| = |\boldsymbol{A}|^{-1}$ and $\mathrm{tr}(\boldsymbol{AB}) = \mathrm{tr}(\boldsymbol{BA})$.
Since $\mathrm{tr}(\boldsymbol{z}A\boldsymbol{z}^T) = \mathrm{tr}(\text{scalar})$, we can rewrite Eqn. (B.27) as:

$$Q_2 = \sum_{m=1}^{M}\sum_{i=1}^{N} \left[\frac{1}{2}\log(|\boldsymbol{\Sigma}_m^{-1}|) - \frac{1}{2}\mathrm{tr}(\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_m)(\boldsymbol{x}_i - \boldsymbol{\mu}_m)^T) \right] p(m|\boldsymbol{x}_i, \Theta^{[k]}) \tag{B.35}$$

Let us quote some more results from Lütkepohl [102]: $\frac{\partial \log(|\boldsymbol{A}|)}{\partial \boldsymbol{A}} = (\boldsymbol{A}^T)^{-1}$ and $\frac{\partial \mathrm{tr}(\boldsymbol{BA})}{\partial \boldsymbol{B}} = \boldsymbol{A}^T$. Moreover, we note that $\boldsymbol{z}\boldsymbol{z}^T$ is a symmetric matrix. To find an expression which maximises $\boldsymbol{\Sigma}_m$, we can take the derivative of Eqn. (B.35) with respect to $\boldsymbol{\Sigma}_m^{-1}$ and set the result to zero:

$$0 = \frac{\partial Q_2}{\partial \boldsymbol{\Sigma}_m^{-1}} \tag{B.36}$$

$$= \frac{\partial}{\partial \boldsymbol{\Sigma}_m^{-1}} \left\{ \sum_{m=1}^{M}\sum_{i=1}^{N} \left[\frac{1}{2}\log(|\boldsymbol{\Sigma}_m^{-1}|) - \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_m)(\boldsymbol{x}_i - \boldsymbol{\mu}_m)^T\right) \right] p(m|\boldsymbol{x}_i, \Theta^{[k]}) \right\} \tag{B.37}$$

$$= \sum_{i=1}^{N} \left[\frac{1}{2}\boldsymbol{\Sigma}_m - \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_m)(\boldsymbol{x}_i - \boldsymbol{\mu}_m)^T \right] p(m|\boldsymbol{x}_i, \Theta^{[k]}) \tag{B.38}$$

$$\tag{B.39}$$

thus

$$\frac{1}{2}\boldsymbol{\Sigma}_m \sum_{i=1}^{N} p(m|\boldsymbol{x}_i, \Theta^{[k]}) = \frac{1}{2}\sum_{i=1}^{N}(\boldsymbol{x}_i - \boldsymbol{\mu}_m)(\boldsymbol{x}_i - \boldsymbol{\mu}_m)^T p(m|\boldsymbol{x}_i, \Theta^{[k]}) \tag{B.40}$$

$$\therefore \ \boldsymbol{\Sigma}_m = \frac{\sum_{i=1}^{N}(\boldsymbol{x}_i - \boldsymbol{\mu}_m)(\boldsymbol{x}_i - \boldsymbol{\mu}_m)^T p(m|\boldsymbol{x}_i, \Theta^{[k]})}{\sum_{i=1}^{N} p(m|\boldsymbol{x}_i, \Theta^{[k]})} \tag{B.41}$$

In summary,

$$w_m^{[k+1]} = \frac{1}{N}\sum_{i=1}^{N} p(m|\boldsymbol{x}_i, \Theta^{[k]}) \tag{B.42}$$

$$\boldsymbol{\mu}_m^{[k+1]} = \frac{\sum_{i=1}^{N} \boldsymbol{x}_i \ p(m|\boldsymbol{x}_i, \Theta^{[k]})}{\sum_{i=1}^{N} p(m|\boldsymbol{x}_i, \Theta^{[k]})} \tag{B.43}$$

$$\boldsymbol{\Sigma}_m^{[k+1]} = \frac{\sum_{i=1}^{N}(\boldsymbol{x}_i - \boldsymbol{\mu}_m^{[k+1]})(\boldsymbol{x}_i - \boldsymbol{\mu}_m^{[k+1]})^T p(m|\boldsymbol{x}_i, \Theta^{[k]})}{\sum_{i=1}^{N} p(m|\boldsymbol{x}_i, \Theta^{[k]})} \tag{B.44}$$

where

$$p(m|\boldsymbol{x}_i, \Theta^{[k]}) = \frac{p(\boldsymbol{x}_i|\theta_m^{[k]})p(m|\Theta^{[k]})}{\sum_{n=1}^{M} p(\boldsymbol{x}_i|\theta_n^{[k]})p(n|\Theta^{[k]})} \tag{B.45}$$

which can be explicitly stated as:

$$p(m|\boldsymbol{x}_i, \Theta^{[k]}) = \frac{\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_m^{[k]}, \boldsymbol{\Sigma}_m^{[k]})w_m^{[k]}}{\sum_{n=1}^{M} \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_n^{[k]}, \boldsymbol{\Sigma}_n^{[k]})w_n^{[k]}} \tag{B.46}$$

If we let $l_{m,i} = p(m|\boldsymbol{x}_i, \Theta^{[k]})$ and $L_m = \sum_{i=1}^{N} l_{m,i}$, we can restate Eqns. (B.42) to (B.44) as:

$$w_m^{[k+1]} = \frac{L_m}{N} \tag{B.47}$$

$$\boldsymbol{\mu}_m^{[k+1]} = \frac{1}{L_m}\sum_{i=1}^{N} \boldsymbol{x}_i \ l_{m,i} \tag{B.48}$$

$$\boldsymbol{\Sigma}_m^{[k+1]} = \frac{1}{L_m}\sum_{i=1}^{N}(\boldsymbol{x}_i - \boldsymbol{\mu}_m^{[k+1]})(\boldsymbol{x}_i - \boldsymbol{\mu}_m^{[k+1]})^T l_{m,i} \tag{B.49}$$

Appendix C

Derivation of Offset-MLLR

In the offset-MLLR approach, each mean is redefined as [c.f. Eqn. (5.7)]:

$$\widehat{\boldsymbol{\mu}}_g = \boldsymbol{\mu}_g + \boldsymbol{\Delta}_g \tag{C.1}$$

where $\boldsymbol{\Delta}_g$ maximises the likelihood of given training data. Substituting (C.1) into (5.4) results in:

$$p(\boldsymbol{x}|\widehat{\boldsymbol{\mu}}_g, \boldsymbol{\Sigma}_g) = \frac{\exp\left[-\frac{1}{2}(\boldsymbol{x} - \{\boldsymbol{\mu}_g + \boldsymbol{\Delta}_g\})^T \boldsymbol{\Sigma}_g^{-1}(\boldsymbol{x} - \{\boldsymbol{\mu}_g + \boldsymbol{\Delta}_g\})\right]}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}_g|^{\frac{1}{2}}} \tag{C.2}$$

In the framework of the Expectation Maximisation (EM) algorithm, we assume that our training data X is incomplete and assume the existence of missing data $Y = \{y_i\}_{i=1}^{N_V}$, where the values of y_i indicate the mixture component (i.e. the Gaussian) that "generated" $\boldsymbol{x}_i$. Thus $y_i \in [1, N_G]\ \forall\ i$ and $y_i = m$ if the i-th feature vector ($\boldsymbol{x}_i$) was "generated" by the m-th Gaussian. An auxiliary function is defined as follows:

$$Q(\lambda, \lambda^{\mathtt{old}}) = \mathrm{E}_Y\left[\log p(X, Y|\lambda)\ |\ X, \lambda^{\mathtt{old}}\right] \tag{C.3}$$

It can be shown [42], that maximising $Q(\lambda, \lambda^{\mathtt{old}})$, i.e.:

$$\lambda^{\mathtt{new}} = \arg\max_{\lambda} Q(\lambda, \lambda^{\mathtt{old}}) \tag{C.4}$$

results in $p(X|\lambda^{\mathtt{new}}) \geq p(X|\lambda^{\mathtt{old}})$ (i.e. the likelihood of the training data X increases). Evaluating the expectation in Eqn. (C.3) results in [24]:

$$Q(\lambda, \lambda^{\mathtt{old}}) = \sum_{g=1}^{N_G}\sum_{i=1}^{N_V} \log[w_g]\ p(g|\boldsymbol{x}_i, \lambda^{\mathtt{old}}) + \sum_{g=1}^{N_G}\sum_{i=1}^{N_V} \log[p(\boldsymbol{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)]\ p(g|\boldsymbol{x}_i, \lambda^{\mathtt{old}}) \tag{C.5}$$

$$= Q_1 \quad + \quad Q_2 \tag{C.6}$$

where

$$p(g|\boldsymbol{x}_i, \lambda^{\texttt{old}}) = \frac{w_g^{\texttt{old}} \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_g^{\texttt{old}}, \Sigma_g^{\texttt{old}})}{\sum_{n=1}^{N_G} w_n^{\texttt{old}} \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_n^{\texttt{old}}, \Sigma_n^{\texttt{old}})} \tag{C.7}$$

A common maximisation technique is to take the derivative of $Q(\lambda, \lambda^{\texttt{old}})$ with respect to the parameter to be maximised and set the result to zero. Since we are interested in finding $\boldsymbol{\Delta}_g$, we only need to take the derivative of Q_2:

$$\begin{aligned}
0 &= \frac{\partial}{\partial \boldsymbol{\Delta}_g} \sum_{g=1}^{N_G} \sum_{i=1}^{N_V} \log[p(\boldsymbol{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)]\ p(g|\boldsymbol{x}_i, \lambda^{\texttt{old}}) && \text{(C.8)} \\
&= \frac{\partial}{\partial \boldsymbol{\Delta}_g} \sum_{g=1}^{N_G} \sum_{i=1}^{N_V} \left[-\frac{1}{2} (\boldsymbol{x}_i - \{\boldsymbol{\mu}_g + \boldsymbol{\Delta}_g\})^T \boldsymbol{\Sigma}_g^{-1} (\boldsymbol{x}_i - \{\boldsymbol{\mu}_g + \boldsymbol{\Delta}_g\}) \right] p(g|\boldsymbol{x}_i, \lambda^{\texttt{old}}) && \text{(C.9)} \\
&= \sum_{i=1}^{N_V} p(g|\boldsymbol{x}_i, \lambda^{\texttt{old}}) \boldsymbol{\Sigma}_g^{-1} (\boldsymbol{x}_i - \{\boldsymbol{\mu}_g + \boldsymbol{\Delta}_g\}) && \text{(C.10)}
\end{aligned}$$

where $-\frac{D}{2}\log(2\pi)$ and $-\frac{1}{2}\log(|\boldsymbol{\Sigma}_g|)$ were omitted in Eqn. (C.9) since they vanish when taking the derivative. Re-arranging Eqn. (C.10) yields:

$$\boldsymbol{\Delta}_g = \frac{\sum_{i=1}^{N_V} p(g|\boldsymbol{x}_i, \lambda^{\texttt{old}}) \boldsymbol{x}_i}{\sum_{i=1}^{N_V} p(g|\boldsymbol{x}_i, \lambda^{\texttt{old}})} - \boldsymbol{\mu}_g \tag{C.11}$$

Substituting Eqn. (C.11) into Eqn. (C.1) yields:

$$\widehat{\boldsymbol{\mu}}_g = \frac{\sum_{i=1}^{N_V} p(g|\boldsymbol{x}_i, \lambda^{\texttt{old}}) \boldsymbol{x}_i}{\sum_{i=1}^{N_V} p(g|\boldsymbol{x}_i, \lambda^{\texttt{old}})} \tag{C.12}$$

which is the standard maximum likelihood re-estimation formula for the mean.

Following [96], we modify the re-estimation formula for tied transformation parameters (e.g. a single $\boldsymbol{\Delta}$ shared by all means). If $\boldsymbol{\Delta}_S$ is shared by N_S Gaussians $\{g_r\}_{r=1}^{N_S}$, Eqn. (C.9) is modified to:

$$\begin{aligned}
0 &= \frac{\partial}{\partial \boldsymbol{\Delta}_S} \sum_{r=1}^{N_S} \sum_{i=1}^{N_V} \left[-\frac{1}{2} (\boldsymbol{x}_i - \{\boldsymbol{\mu}_{g_r} + \boldsymbol{\Delta}_S\})^T \boldsymbol{\Sigma}_{g_r}^{-1} (\boldsymbol{x}_i - \{\boldsymbol{\mu}_{g_r} + \boldsymbol{\Delta}_S\}) \right] p(g_r|\boldsymbol{x}_i, \lambda^{\texttt{old}}) && \text{(C.13)} \\
&= \sum_{r=1}^{N_S} \sum_{i=1}^{N_V} p(g_r|\boldsymbol{x}_i, \lambda^{\texttt{old}}) \boldsymbol{\Sigma}_{g_r}^{-1} (\boldsymbol{x}_i - \{\boldsymbol{\mu}_{g_r} + \boldsymbol{\Delta}_S\}) && \text{(C.14)}
\end{aligned}$$

which leads to:

$$\boldsymbol{\Delta}_S = \left[\sum_{r=1}^{N_S} \sum_{i=1}^{N_V} p(g_r|\boldsymbol{x}_i, \lambda^{\texttt{old}}) \boldsymbol{\Sigma}_{g_r}^{-1} \right]^{-1} \left[\sum_{r=1}^{N_S} \sum_{i=1}^{N_V} p(g_r|\boldsymbol{x}_i, \lambda^{\texttt{old}}) \boldsymbol{\Sigma}_{g_r}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_{g_r}) \right] \tag{C.15}$$

References

[1] Y. Abdeljaoued. Fusion of person authentication probabilities by Bayesian statistics. Second International Conference on Audio- and Video-based Biometric Person Authentication, Washington D.C., 1999, pp. 172-175.

[2] B. Achermann, H. Bunke. Combination of classifiers on the decision level for face recognition. Technical Report IAM-96-002, Institut für Informatik und angewandte Mathematik, Universität Bern, 1996.

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

[4] A. Adjoudani, C. Benoît. Audio-visual speech recognition compared across two architectures. European Conference on Speech Communication and Technology, Madrid, Spain, 1995, Vol. 2, pp. 1563-1567.

[5] L.A. Alexandre, A.C. Campilho, M. Kamel. On combining classifiers using sum and product rules. Pattern Recognition Letters, Vol. 22, 2001, pp. 1283-1289.

[6] H. Altiçay, M. Demirekler. Comparison of different objective functions for optimal linear combination of classifiers for speaker identification. IEEE International Conference on Acoustics, Speech and Signal Processing, Salt Lake City, 2001, Vol. 1, pp. 401-404.

[7] B.S. Atal. Automatic speaker recognition based on pitch contours. PhD Thesis, Polytechnic Institute of Brooklyn, 1968.

[8] B.S. Atal, M.R. Schroeder. Predictive coding of speech signals. Report of the 6th International Congress on Acoustics, Tokyo, 1968.

[9] B.S. Atal. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. Journal of the Acoustical Society of America, Vol. 55, No. 6, 1974, pp. 1304-1312.

[10] J.J. Atick, P.A. Griffin, A.N. Redlich. Statistical approach to shape from shading: reconstruction of three-dimensional face surfaces from single two-dimensional images. Neural Computation, Vol. 8, 1996, pp. 1321–1340.

[11] W. Atkins. A testing time for face recognition technology. Biometric Technology Today, Vol. 9, No. 3, 2001, pp. 8-11.

[12] R. Auckenthaler, M. Carey, H. Lloyd-Thomas. Score normalization for text-independent speaker verification systems. Digital Signal Processing, Vol. 10, 2000, pp. 42-54.

[13] R. Balchandran, V. Ramanujam, R. Mammone. Channel estimation and normalization by coherent spectral averaging for robust speaker verification. European Conference on Speech Communication and Technology, Budapest, 1999, pp. 755-758.

[14] Y. Barniv, D. Casasent. Multisensor image registration: Experimental verification. Proceedings of the SPIE, Vol. 292, 1981, pp. 160-171.

[15] T. Bayes. An essay towards solving a problem in the doctrine of chances. Philosophical Transactions of the Royal Society, Vol. 53, 1763, pp. 370-418.

[16] J. Bedo, C. Sanderson, A. Kowalczyk. An efficient alternative to SVM based recursive feature elimination with applications in natural language processing and bioinformatics. AI 2006: Advances in Artificial Intelligence, Lecture Notes in Computer Science, Vol. 4304, 2006, pp. 170-180.

[17] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, 1997, pp. 711-720.

[18] S. Bengio, J. Mariethoz, S. Marcel. Evaluation of biometric technology on XM2VTS. IDIAP Research Report 01-21, Martigny, Switzerland, 2001.

[19] S. Bengio. Multimodal authentication using asynchronous HMMs. Audio- and Video-Based Biometric Person Authentication, Lecture Notes in Computer Science, Vol. 2688, 2003, pp. 770-777.

[20] S. Bengio. Multimodal speech processing using asynchronous hidden Markov models. Information Fusion, Vol. 5, 2004, pp. 81–89.

[21] S. Bengio, J. Mariéthoz. The expected performance curve: a new assessment measure for person authentication. Odyssey 2004: The Speaker and Language Recognition Workshop, Toledo, 2004, pp. 279-284.

[22] S. Ben-Yacoub, Y. Abdeljaoued, E. Mayoraz. Fusion of face and speech data for person identity verification. IEEE Transactions on Neural Networks, Vol. 10, No. 5, 1999, pp. 1065-1074.

[23] D. Beymer, T. Poggio. Face recognition from one example view. International Conference on Computer Vision (ICCV), Cambridge, 1995, pp. 500-507.

[24] J.A. Bilmes. A gentle tutorial of the EM algorithm and its applications to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report TR-97-021, International Computer Science Institute, Berkeley, California, 1998.

[25] R.M. Bolle, J.H. Connell, N.K. Ratha. Biometric perils and patches. Pattern Recognition, Vol. 35, 2002, pp. 2727–2738.

[26] K. Bowyer, K. Chang, P. Flynn. A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition. Computer Vision and Image Understanding, Vol. 101, No. 1, 2006, pp. 1-15.

[27] V. Blanz, S. Romdhani, T. Vetter. Face identification across different poses and illuminations with a 3D morphable model. IEEE International Conference on Automatic Face and Gesture Recognition, Washington, D.C., 2002, pp. 192-197.

[28] R. Brunelli, T. Poggio. Face recognition: features versus templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 10, 1993, pp. 1042-1052.

[29] R. Brunelli, D. Falavigna, T. Poggio, L. Stringa. Automatic person recognition using acoustic and geometric features. Machine Vision and Applications, Vol. 8, No. 5, 1995, pp. 317-325.

[30] R. Brunelli, D. Falavigna. Person identification using multiple cues. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10, No. 17, 1995, pp. 955-965.

[31] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, Vol. 2, No 2, 1998, pp. 121-167.

[32] E. Caulcott. Significance tests, Routledge & Kegan Paul, London, 1973.

[33] F. Cardinaux, C. Sanderson, S. Bengio. User authentication via adapted statistical models of face images. IEEE Transactions on Signal Processing, Vol. 54, No. 1, 2006, pp. 361–373.

[34] K.R. Castleman. Digital Image Processing, Prentice-Hall, USA, 1996.

[35] C. Champod, D. Meuwly. The inference of identity in forensic speaker recognition. Speech Communication, Vol. 31, No. 2-3, 2000, pp. 193-203.

[36] R. Chellappa, C.L. Wilson, S. Sirohey. Human and machine recognition of faces: a survey. Proceedings of the IEEE, Vol. 83, No. 5, 1995, pp. 705-740.

[37] L-F. Chen, H-Y. Liao, J-C. Lin, C-C. Han. Why recognition in a statistics-based face recognition system should be based on the pure face portion: a probabilistic decision-based proof. Pattern Recognition, Vol. 34, No. 7, 2001, pp. 1393-1403.

[38] C.C. Chibelushi, F. Deravi, J. S. Mason. Voice and facial image integration for speaker recognition. IEEE International Symposium and Multimedia Technologies and Future Applications, Southampton, UK, 1993.

[39] J. Czyz, J. Kittler, L. Vandendorpe. Multiple classifer combination for face-based identity verifcation. Pattern Recognition, Vol. 37, No. 7, 2004, pp. 1459–1469.

[40] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, C. Bray. Visual cetegorization with bags of keypoints. Workshop on Statistical Learning in Computer Vision (in conjunction with ECCV'04), Prague, 2004.

[41] J. Daugman. The importance of being random: statistical principles of iris recognition. Pattern Recognition, Vol. 36, No. 2, 2003, pp. 279–291.

[42] A.P. Dempster, N.M. Laird, D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B (Methodological), Vol. 39, No. 1, 1977, pp. 1–38.

[43] U. Dieckmann, P. Plankensteiner, T. Wagner. SESAM: a biometric person identification system using sensor fusion. Pattern Recognition Letters, Vol. 18, No. 9, 1997, pp. 827-833.

[44] G.R. Doddington, M.A. Przybycki, A.F. Martin, D.A. Reynolds. The NIST speaker recognition evaluation - Overview, methodology, systems, results, perspective. Speech Communication, Vol. 31, No. 2-3, 2000, pp. 225-254.

[45] B. Duc, S. Fischer, J. Bigün. Face authentication with gabor information on deformable graphs. IEEE Transactions on Image Processing, Vol. 8, No. 4, 1999, pp. 504-516.

[46] R.O. Duda, P.E. Hart, D.G. Stork. Pattern Classification, John Wiley & Sons, USA, 2001.

[47] J.-L. Dugelay, J.-C. Junqua, C. Kotropoulos, R. Kuhn, F. Perronnin, I. Pitas. Recent advances in biometric person authentication. International Conference on Acoustics, Speech and Signal Processing (ICASSP), Orlando, 2002, Vol. IV, pp. 4060-4063.

[48] S. Eickeler, S. Müller, G. Rigoll. Recognition of JPEG compressed face images based on statistical methods. Image and Vision Computing, Vol. 18, No. 4, 2000, pp. 279-287.

[49] K.R. Farrell. Text-dependent speaker verification using data fusion. IEEE International Conference on Acoustics, Speech and Signal Processing, Detroit, Michigan, 1995, Vol. 1, pp. 349-352.

[50] S. Furui. Cepstral analysis technique for automatic speaker verification. IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 29, No. 2, 1981, pp. 254-272.

[51] S. Furui. Recent advances in speaker recognition. Pattern Recognition Letters, Vol. 18, No. 9, 1997, pp. 859-872.

[52] B. Gajic, K.K. Paliwal. Robust feature extraction using subband spectral centroid histograms. International Conference on Acoustics, Speech and Signal Processing, Salt Lake City, 2001, pp. 85-88.

[53] M.J.F. Gales, P.C. Woodland. Variance compensation within the MLLR framework. Technical Report 242, Cambridge University Engineering Department, UK, 1996.

[54] J-L. Gauvain, C-H. Lee. Bayesian learning for hidden Markov model with Gaussian mixture state observation densities. Speech Communication, Vol. 11, 1992, pp. 205-213.

[55] J-L. Gauvain, C-H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. IEEE Transactions on Speech and Audio Processing, Vol. 2, No. 2, 1994. pp. 291-298.

[56] A. Gersho, R.M. Gray. Vector Quantization and Signal Compression, Springer, 1991.

[57] D. Genoud, F. Bimbot, G. Gravier, G. Chollet. Combining methods to improve speaker verification. International Conference on Spoken Language Processing, Philadelphia, 1996, Vol. 3, pp. 1756-1759.

[58] H. Gish, M. Schmidt, Text-independent speaker identification. IEEE Signal Processing Magazine, Vol. 11, No. 4, 1994, pp. 18-32.

[59] R.C. Gonzalez, R.E. Woods. Digital Image Processing. Addison-Wesley, 1992.

[60] D. Graham, N. Allinson. Face recognition from unfamiliar views: subspace methods and pose dependency. IEEE International Conference on Automatic Face and Gesture Recognition (AFGR), Nara, 1998, pp. 348-353.

[61] R. Gross, J. Yang, A. Waibel. Growing gaussian mixture models for pose invariant face recognition. International Conference on Pattern Recognition, Barcelona, 2000, Vol. 1, pp. 1088-1091.

[62] M.A. Grudin. On internal representations in face recognition systems. Pattern Recognition, Vol. 33, No. 7, 2000, pp. 1161-1177.

[63] I. Guyon, J. Weston, S. Barnhill, V. Vapnik. Gene selection for cancer classication using support vector machines. Machine Learning, Vol. 46, 2002, pp. 389-422.

[64] J.A. Haigh, J.S. Mason. A voice activity detector based on cepstral analysis. European Conference on Speech Communication and Technology, 1993, Vol. 2, pp. 1103-1106.

[65] J.A. Haigh. Voice activity detection for conversational analysis. Master's Thesis, University of Wales, 1994.

[66] D.L. Hall, J. Llinas. Multisensor data fusion. Handbook of Multisensor Data Fusion (editors: D. L. Hall and J. Llinas), CRC Press, USA, 2001, pp. 1-1 - 1-10.

[67] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, S. Pankanti, A. Senior, C.-F. Shu, Y.L. Tian. Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking. IEEE Signal Processing Magazine, Vol. 22, No. 2, 2005, pp. 38–51.

[68] H. Hermansky, N. Morgan, A. Bayya, P. Kohn. RASTA-PLP speech analysis technique. IEEE International Conference on Acoustics, Speech and Signal Processing, San Francisco, 1992, Vol. 1, pp. 121-124.

[69] H. Hermansky, N. Morgan. RASTA Processing of Speech. IEEE Transactions on Speech and Audio Processing, Vol. 2, No. 4, 1994, pp. 578-589.

[70] E. Hjelmås, B.K. Low. Face detection: a survey. Computer Vision and Image Understanding, Vol. 83, No. 3, 2001, pp. 236-274.

[71] T.K. Ho, J.J. Hull, S.N. Srihari. Decision combination in multiple classifier systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, No. 1, 1994, pp. 66-75.

[72] L. Hong, A. Jain. Integrating faces and fingerprints for personal identification. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 12, 1998, pp. 1295-1306.

[73] X. Huang, A. Acero, H-W. Hon. Spoken Language Processing: A Guide to Theory, Algorithm and System Development, Prentice Hall PTR, New Jersey, 2001.

[74] Q. Huo, C. Chan, C-H. Lee. Bayesian adaptive learning of the parameters of hidden Markov model for speech recognition. IEEE Transactions on Speech and Audio Processing, Vol. 3, No. 5, 1995, pp. 334-345.

[75] International Phonetic Association. Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet, Cambridge University Press, 1999.

[76] S.S. Iyengar, L. Prasad, H. Min. Advances in Distributed Sensor Technology, Prentice Hall PTR, New Jersey, 1995.

[77] A. Jain, U. Uludag. Hiding biometric data. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, 2003, pp. 1494–1498.

[78] T. Joachims. Making large-scale SVM learning practical. Advances in Kernel Methods - Support Vector Learning, MIT-Press, 1999.

[79] N.L. Johnson, F.C. Leone. Statistics and Experimental Design in Engineering and the Physical Sciences (Vol. 1), John Wiley & Sons, USA, 1977.

[80] C. Jankowski, A. Kalyanswamy, S. Basson, J. Spitz. NTIMIT: a phonetically balanced, continuous speech telephone bandwidth speech database. International Conference on Acoustics, Speech and Signal Processing, Albuquerque, 1990, Vol. 1, pp. 109-112.

[81] P. Jourlin, J. Luettin, D. Genoud, H. Wassner. Acoustic-labial speaker verification. Pattern Recognition Letters, Vol. 18, No. 9, 1997, pp. 853-858.

[82] P. Jourlin, J. Luettin, D. Genoud, H. Wassner. Integrating acoustic and labial information for speaker indentification and verification. European Conference on Speech Communication and Technology, Rhodes, Greece, 1997, Vol. 3, pp. 1603-1606.

[83] Behrooz Kamgar-Parsi, Behzad Kamgar-Parsi, A. Jain, J. Dayhoff. Aircraft detection: a case study in using human similarity measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 12, 2001, pp. 1404–1414.

[84] D-S. Kim, S-Y. Lee, R.M. Kil. Auditory processing of speech signals for robust speech recognition in real-world noisy environments. IEEE Transactions on Speech and Audio Processing, Vol. 7, No. 1, 1999, pp. 55-69.

[85] M. Kirby, L. Sirovich. Application of the Karhunen-Loève procedure for the characterization of human faces. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 1, 1990, pp. 103-108.

[86] J. Kittler, J. Matas, K. Johnsson, M. U. Ramos Sánchez. Combining evidence in personal identity verification systems. Pattern Recognition Letters, Vol. 18, No. 9, 1997, pp. 845-852.

[87] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas. On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 3, 1998, pp. 226-239.

[88] L.H. Koh, S. Ranganath, Y.V. Venkatesh. An integrated automatic face detection and recognition system. Pattern Recognition, Vol. 35, No. 6, 2002, pp. 1259-1273.

[89] T. Kohonen, The self-organizing map. Proceedings of the IEEE, Vol. 78, No. 9, 1990, pp. 1464-1480.

[90] S.G. Kong, J. Heo, B.R. Abidi, J. Paik, M.A. Abidi. Recent advances in visual and infrared face recognition - a review. Computer Vision and Image Understanding, Vol. 97, No. 1, 2005, pp. 103–135.

[91] C. Kotropoulos, A. Tefas, I. Pitas. Frontal face authentication using morphological elastic graph matching. IEEE Transactions on Image Processing, Vol. 9, No. 4, 2000, pp. 555-560.

[92] C. Kotropoulos, A. Tefas, I. Pitas. Morphological elastic graph matching applied to frontal face authentication under well-controlled and real conditions. Pattern Recognition, Vol. 33, No. 12, 2000, pp. 1935-1947.

[93] S. Lawrence, C.L. Giles, A.C. Tsoi, A.D. Back. Face recognition: a convolutional neural-network approach. IEEE Transactions on Neural Networks, Vol. 8, No. 1, 1997, pp. 98-113.

[94] M. Lades, J.C. Vorbrüggen, J. Buhmann, J. Lange, C. v.d. Malsburg, R.P. Würtz, W. Konen. Distortion invariant object recognition in the dynamic link architecture. IEEE Transactions on Computers, Vol. 42, No. 3, 1993, pp. 300-311.

[95] T.S. Lee. Image representation using 2D Gabor wavelets. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 10, 1996, pp. 959-971.

[96] C.J. Leggetter, P.C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. Computer Speech and Language, Vol. 9, No. 2, 1995, pp. 171–185.

[97] Y. Linde, A. Buzo, R.M. Gray. An algorithm for vector quantization. IEEE Transactions on Communications, Vol. 28, No. 1, 1980, pp. 84-95.

[98] M. Lockie (editor). Facial verification bureau launched by police IT group. Biometric Technology Today, Vol. 10, No. 3, 2002, pp. 3-4.

[99] S. Lucey, T. Chen. A GMM parts based face representation for improved verification through relevance adaptation. Computer Vision and Pattern Recognition, Washington, D.C., 2004, Vol. 2, pp. 855-861.

[100] J. Luettin. Visual speech and speaker recognition. PhD Thesis, Department of Computer Science, University of Sheffield, 1997.

[101] R.C. Luo, M.G. Kay. Introduction. Multisensor Integration and Fusion for Intelligent Machines and Systems (editors: R.C. Luo, M.G. Kay), Ablex Publishing Corporation, Norwood, NJ, 1995, pp. 1-26.

[102] H. Lütkepohl. Handbook of Matrices, John Wiley & Sons, UK, 1996.

[103] J. Makhoul. Spectral analysis of speech by linear prediction. IEEE Transactions on Acoustics Speech and Signal Processing, Vol. 21, No. 3, 1973, pp. 140-148.

[104] J. Mariéthoz, S. Bengio. A comparative study of adaptation methods for speaker verification. International Conference on Spoken Language Processing (ICSLP), Denver, 2002, pp. 581-584.

[105] J. Markowitz. Speech systems work together in harmony. Biometric Technology Today, Vol. 9, No. 4, 2001, pp. 7-8.

[106] M. Martelli, N.J. Majaj, D.G. Pelli. Are faces processed like words? A diagnostic test for recognition by parts. Journal of Vision, Vol. 5, No. 1, 2005, pp. 58-70.

[107] J. Matas, K. Jonsson, J. Kittler. Fast face localisation and verification. Image and Vision Computing, Vol. 17, No. 8, 1999, pp. 757-581.

[108] T. Maurer, C. v.d. Malsburg. Learning feature transformations to recognize faces rotated in depth. International Conference on Artificial Neural Networks (ICANN), Paris, 1995, pp. 353-358.

[109] G.J. McLachlan, T. Krishnan. The EM Algorithm and Extensions (2nd edition). John Wiley & Sons. 2008.

[110] K. Messer, J. Matas, J. Kittler, J. Luettin, G. Maitre. XM2VTSDB: the extended M2VTS database. Second International Conference on Audio- and Video-based Biometric Person Authentication, Washington D.C., 1999, pp. 72-77.

[111] K. Messer, J. Kittler, M. Sadeghi, S. Marcel, C. Marcel, S. Bengio, F. Cardinaux, C. Sanderson, J. Czyz, L. Vandendorpe, S. Srisuk, M. Petrou, W. Kurutach, A. Kadyrov, R. Paredes, B. Kepenekci, F.B. Tek, G.B. Akar, F. Deravi, N. Mavity. Face verification competition on the XM2VTS database. Audio- and Video-Based Biometric Person Authentication, Lecture Notes in Computer Science, Volume 2688, 2003, pp. 964–974.

[112] E. Messmer. Pentagon lab may give biometrics needed boost. CNN.com web site (http://www.cnn.com/2001/TECH/science/03/20/pentagon.biometrics.idg/index.html), 20 March 2001.

[113] I. Miller, J.E. Freund, R.A. Johnson. Probability and Statistics for Engineers (4th edition), Prentice-Hall, USA, 1990.

[114] T.M. Mitchell. Machine Learning, WCB/McGraw-Hill, USA, 1997.

[115] B. Moghaddam, A. Pentland. Probabilistic visual learning for object representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, 1997, pp. 696-710.

[116] T.K. Moon. Expectation-maximization algorithm. IEEE Signal Processing Magazine, Vol. 13, No. 6, 1996, pp. 47-60.

[117] T.K. Moon, W.C. Stirling. Mathematical Methods and Algorithms for Signal Processing, Prentice Hall, Upper Saddle River, New Jersey, 2000.

[118] H. Moon, P.J. Phillips. Computational and performance aspects of PCA-based face-recognition algorithms. Perception, Vol. 30, 2001, pp. 303-321.

[119] B.C.J. Moore. Frequency analysis and masking. Hearing (editors: D. A. Eddins, D. M. Green), Academic Press, USA, 1995.

[120] B.C.J. Moore. Information extraction and perceptual grouping in the auditory system. Human and Machine Perception: Information Fusion (editors: V. Cantoni, V. D. Gesù, A. Setti, D. Tegolo), Plenum Press, New York, 1997.

[121] J.A. Moorer. The optimum comb method of pitch period analysis of continuous digitized speech. IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 22, No. 3, 1974, pp. 330-338.

[122] A.V. Nefian, M.H. Hayes. Hidden Markov models for face recognition. International Conference on Acoustics, Speech and Signal Processing, Seattle, 1998, Vol. 5, pp. 2721-2724.

[123] A.V. Nefian, L.H. Liang, T. Fu, X.X. Liu. A Bayesian approach to audio-visual speaker identification. Audio- and Video-Based Biometric Person Authentication, Lecture Notes in Computer Science, Volume 2688, 2003, pp. 761–769.

[124] J.A. Nelder, R. Mead. A simplex method for function minimization. The Computer Journal, Vol. 7, No. 4, 1965, pp. 308-313.

[125] A.P. Nilsen. Why keep searching when it's already "their"? Reconsidering "everybody's" pronoun problem. The English Journal, Vol. 90, No. 4, 2001, pp. 68–73.

[126] P. Niyogi, F. Girosi, T. Poggio. Incorporating prior information in machine learning by creating virtual examples. Proceedings of the IEEE, Vol. 86, No. 11, 1998, pp. 2196–2209.

[127] E. Nowak, F. Jurie, B. Triggs. Sampling strategies for bag-of-features image classification. European Conference on Computer Vision, Part IV, Lecture Notes in Computer Science, Vol. 3954, 2006, pp. 490-503.

[128] J. Ortega-Garcia, J. Bigun, D. Reynolds, J. Gonzales-Rodriguez. Authentication gets personal with biometrics. IEEE Signal Processing Magazine, Vol. 21, No. 2, 2004, pp. 50–62.

[129] K.K. Paliwal. Speech processing techniques. Advances in Speech, Hearing and Language Processing (editor: W.A. Ainsworth), Vol. 1, 1990, pp. 1-78.

[130] K.K. Paliwal. Spectral subband centroid features for speech recognition. International Conference on Acoustics, Speech and Signal Processing, Seattle, Washington, 1998, Vol. 2, pp. 617-620.

[131] T.W. Parsons. Voice and Speech Processing, McGraw-Hill, USA, 1987.

[132] L.F. Pau. Fusion of multisensor data in pattern recognition. Pattern recognition theory and applications (editors: J. Kittler, K.S. Fu, L.F. Pau), Springer, 1982.

[133] A. Pentland, B. Moghaddam, T. Starner. View-based and modular eigenspaces for face recognition. International Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, 1994, pp. 84-91.

[134] P.J. Phillips, H. Moon, S.A. Rizvi, P.J. Rauss. The FERET evaluation methodology for face-recognition algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 22, No. 10, 2000, pp. 1090–1104.

[135] J. Picone. Signal modeling techniques in speech recognition. Proceedings of the IEEE, Vol. 81, No. 9, 1993, pp. 1215-1247.

[136] S. Pigeon, L. Vandendorpe. The M2VTS multimodal face database (release 1.00). International Conference on Audio- and Video-based Biometric Person Authentication, Crans-Montana, Switzerland, 1997, pp. 403-409.

[137] N. Poh, S. Bengio. Non-linear variance reduction techniques in biometric authentication. Workshop on Multimodal User Authentication, Santa Barbara, 2003, pp. 123-130.

[138] G. Potamianos, H. Graf. Discriminative training of HMM stream exponents for audio-visual speech recognition. International Conference on Acoustics, Speech and Signal Processing, Seattle, 1998, pp. 3733-3736.

[139] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery. Numerical Recipes in C: the Art of Scientific Computing, Cambridge University Press, 1992.

[140] J.G. Proakis, D.G. Manolakis. Digital Signal Processing: Principles, Algorithms, and Applications (3rd edition), Prentice Hall, USA, 1996.

[141] L. Rabiner, B-H. Juang. Fundamentals of Speech Recognition, Prentice Hall PTR, 1993.

[142] V. Radová, J. Psutka. An approach to speaker identification using multiple classifiers. IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP), Munich, Germany, 1997, Vol. 2, pp. 1135-1138.

[143] R.A. Redner, H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. SIAM Review, Vol. 26, No. 2, 1984, pp. 195-239.

[144] D.A. Reynolds. A Gaussian mixture modeling approach to text-independent speaker identification. Technical Report 967, Lincoln Laboratory, Massachusetts Institute of Technology, 1993.

[145] D.A. Reynolds. Experimental evaluation of features for robust speaker identification. IEEE Transactions on Speech and Audio Processing, Vol. 2, No. 4, 1994, pp. 639-643.

[146] D.A. Reynolds. Speaker identification and verification using Gaussian mixture speaker models. Speech Communication, Vol. 17, No. 1-2, 1995, pp. 91-108.

[147] D.A. Reynolds. Robust text-independent speaker identification using Gaussian mixture speaker models. IEEE Transactions on Speech and Audio Processing, Vol. 3, No. 1, 1995, pp. 72-83.

[148] D.A. Reynolds. Comparison of background normalization methods for text-independent speaker verification. European Conference on Speech Communication and Technology, Rhodes, Greece, 1997, Vol. 2, pp. 963-966.

[149] D. Reynolds, T. Quatieri, R. Dunn. Speaker verification using adapted Gaussian mixture models. Digital Signal Processing, Vol. 10, No. 1-3, 2000, pp. 19-41.

[150] J.A. Rice. Mathematical Statistics and Data Analysis, 2nd edition, Duxbury Press, 1995.

[151] C.P. Robert. The Bayesian Choice: A Decision-Theoretic Motivation, Springer-Verlag, New York, 1994.

[152] Y. Rodriguez, F. Cardinaux, S. Bengio, J. Mariéthoz. Measuring the performance of face localization systems. Image and Vision Computing, Vol. 24, No. 8, 2006, pp. 882–893.

[153] A.E. Rosenberg, J. DeLong, C-H. Lee, B-H. Juang, F. K. Soong. The use of cohort normalized scores for speaker verification. International Conference on Spoken Language Processing, Alberta, 1992, Vol. 1, pp. 599-602.

[154] A.E. Rosenberg, S. Parthasarathy. Speaker background models for connected digit password speaker verification. International Conference on Acoustics, Speech and Signal Processing, Atlanta, 1996, Vol. 1, pp. 81-84.

[155] M. Ross, H. Shaffer, A. Cohen, R. Freudberg, H. Manley. Average magnitude diffrence function pitch extractor. IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 22, No. 5, 1974, pp. 353-362.

[156] A. Ross, A. Jain. Information fusion in biometrics. Pattern Recognition Letters, Vol. 24, 2003, pp. 2115–2125.

[157] V. Roth, V. Steinhage. Nonlinear discriminant analysis using kernel functions. Technical Report Nr. IAI-TR-99-7 (ISSN 0944-8535), University of Bonn, 1999.

[158] F. Samaria. Face recognition using hidden Markov models. PhD Thesis, University of Cambridge, 1994.

[159] C. Sanderson, Ting Shang, B.C. Lovell. Towards pose-invariant 2D face classification for surveillance. Analysis and Modeling of Faces and Gestures, Lecture Notes in Computer Science (LNCS), Vol. 4778, 2007, pp. 276-289.

[160] C. Sanderson, K.K. Paliwal. Noise compensation in a person verification system using face and multiple speech features. Pattern Recognition, Vol. 36, No. 2, 2003, pp. 293–302.

[161] T. Shan, B.C. Lovell. Face recognition robust to head pose from one sample image. International Conference on Pattern Recognition (ICPR), Vol. 1, 2006, pp. 515–518.

[162] D. O'Shaughnessy. Speech communications: human and machine (2nd edition), IEEE Press, New York, 2000.

[163] J. Shawe-Taylor, N. Cristianini. Kernel Methods for Pattern Analysis, Cambridge University Press, 2004.

[164] J. Sivic, A. Zisserman. Video Google: A text retrieval approach to object matching in videos. International Conference on Computer Vision (ICCV), Vol. 2, 2003, pp. 1470-1477.

[165] F.K. Soong, A.E. Rosenberg. On the use of instantaneous and transitional spectral information in speaker recognition. IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 36, No. 6, 1988, pp. 871-879.

[166] P. Silsbee, A. Bovik. Computer lipreading for improved accuracy in automatic speech recognition. IEEE Transactions on Speech and Audio Processing Vol. 4, No. 5, 1996, pp. 337-351.

[167] B. Sonesson. The functional anatomy of the speech organs. Manual of Phonetics (editor: B. Malmberg), North-Holland, Amsterdam, 1968, pp. 45-75.

[168] E.W. Swokowski. Calculus (5th edition), PWS-Kent, USA, 1991.

[169] A. Tefas, C. Kotropoulos, I. Pitas. Face authentication by using elastic graph matching and support vector machines. International Conference on Acoustics, Speech and Signal Processing, Istanbul, 2000, pp. 2409-2412 (Vol. 4).

[170] R.R. Tenney, N.R. Sandell Jr. Detection with distributed sensors. IEEE Transactions on Aerospace and Electronic Systems, Vol. 17, No. 4, 1981, pp. 501–510.

[171] R.R. Tenney, N.R. Sandell Jr. Strategies for distributed decisionmaking. IEEE Transactions on Systems, Man and Cybernetics, Vol. 11, No. 8, 1981, pp. 527–538.

[172] Norman Poh Hoon Thian, C. Sanderson, S. Bengio. Spectral subband centroids as complementary features for speaker authentication. Biometric Authentication, Lecture Notes in Computer Science, Volume 3072, 2004, pp. 631–639.

[173] T. Thong, Y.C. Jenq. Hardware and architecture. Handbook for Digital Signal Processing (editors: S. K. Mitra and J. F. Kaiser), John Wiley & Sons, USA, 1993, pp. 721-781.

[174] M. Turk, A. Pentland. Eigenfaces for recognition. Journal of Cognitive Neuroscience, Vol. 3, No. 1, 1991, pp. 71-86.

[175] D. Valentin, H. Abdi, A.J. O'Toole, G.W. Cottrell. Connectionist models of face processing: a survey. Pattern Recognition, Vol. 27, No. 9, 1994, pp. 1209-1230.

[176] V.N. Vapnik. The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.

[177] A. Varga, H.J.M. Steeneken, M. Tomlinson, D. Jones. The NOISEX-92 study on the effect of additive noise on automatic speech recognition. Technical Report, Defence Evaluation and Research Agency (DERA), Speech Research Unit, Malvern, United Kingdom, 1992.

[178] P.K. Varshney. Distributed Detection and Data Fusion, Springer-Verlag, New York, 1997.

[179] P. Verlinde. A Contribution to multi-modal identity verification using decision fusion. PhD Thesis, Department of Signal and Image Processing, Telecom Paris, France, 1999.

[180] T. Vetter, T. Poggio. Linear object classes and image synthesis from a single example image. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, 1997, pp. 733–742.

[181] T. Wark, S. Sridharan, V. Chandran. Robust speaker verification via fusion of speech and lip modalities. International Conference on Acoustics, Speech and Signal Processing, Phoenix, Arizona, 1999, Vol. 6, pp. 3061-3064.

[182] T. Wark, S. Sridharan, V. Chandran. Robust speaker verification via asynchronous fusion of speech and lip information. International Conference on Audio- and Video-based Biometric Person Authentication, Washington, D.C., 1999, pp. 37-42.

[183] T. Wark, S. Sridharan, V. Chandran. The use of temporal speech and lip information for multi-modal speaker identification via multi-stream HMM's. International Conference on Acoustics, Speech and Signal Processing, Istanbul, 2000, pp. 2389-2392.

[184] T.J. Wark. Multi-modal speech processing for automatic speaker recognition. PhD Thesis, School of Electrical & Electronic Systems Engineering, Queensland University of Technology, Brisbane, 2000.

[185] J.L. Wayman. Digital signal processing in biometric identification: a review. IEEE International Conference on Image Processing (ICIP), Rochester, 2002, Vol. 1, pp. 37-40.

[186] G.K. Wallace. The JPEG still picture compression standard. Communications of the Association for Computing Machinery, Vol. 34, No. 4, 1991, pp. 30-44.

[187] G.K. Wallace. The JPEG still picture compression standard. IEEE Transactions on Consumer Electronics, Vol. 38, No. 1, 1992, pp. xviii-xxxiv.

[188] A. Webb. Statistical Pattern Recognition, John Wiley & Sons, UK, 2002.

[189] B. Wildermoth, K. K. Paliwal. Use of voicing and pitch information for speaker recognition. Australian International Conference on Speech Science and Technology, Canberra, 2000, pp. 324-328.

[190] J.D. Woodward. Biometrics: privacy's foe or privacy's friend? Proceedings of the IEEE, Vol. 85, No. 9, 1997, pp. 1480-1492.

[191] M-H. Yang, D.J. Kriegman, N. Ahuja. Detecting faces in images: a survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 1, 2002, pp. 34–58.

[192] J. Zhang, Y. Yan, M. Lades. Face recognition: eigenface, elastic matching, and neural nets. Proceedings of the IEEE, Vol. 85, No. 9, 1997, pp. 1423-1435.

[193] W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld. Face recognition: a literature survey. ACM Computing Surveys Vol. 35, No. 4, 2003, pp. 399–458.

Index