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)

August 2002

[revised February 2003]

STATEMENT OF ORIGINALITY

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Conrad Sanderson, August 2002

ABSTRACT

Identity verification systems are an important part of our every day life. A typical example is the Automatic Teller Machine (ATM) which employs a simple identity verification scheme: the user is asked to enter their secret password after inserting their ATM 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 major 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. Thus new verification methods have emerged, where the password has either been replaced by, or used in addition to, biometrics such as the person's speech, face image or fingerprints.

Apart from the ATM example described above, biometrics can be applied to other areas, such as telephone & internet based banking, airline reservations & check-in, as well as forensic work and law enforcement applications.

Biometric systems based on face images and/or speech signals have been shown to be quite effective. However, their performance easily degrades in the presence of a mismatch between training and testing 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.

A system which uses more than one biometric at the same time is known as a multi-modal verification system; it is often comprised of several modality experts and a decision stage. Since a multi-modal system uses complimentary discriminative information, lower error rates can be achieved; moreover, such a system can also be more robust, since the contribution of the modality affected by environmental conditions can be decreased.

This thesis makes several contributions aimed at increasing the robustness of single- and multi-modal verification systems. Some of the major contributions are listed below.

The robustness of a speech based system to ambient noise is increased by using Maximum Auto-Correlation Value (MACV) features, which utilize information from the source part of the speech signal.

A new facial feature extraction technique is proposed (termed *DCT-mod2*), which utilizes polynomial coefficients derived from 2D Discrete Cosine Transform (DCT) coefficients of spatially neighbouring blocks. The *DCT-mod2* features are shown to be robust to an illumination direction change as well as being over 80 times quicker to compute than 2D Gabor wavelet derived features.

The fragility of Principal Component Analysis (PCA) derived features to an illumination direction change is solved by introducing a pre-processing step utilizing the *DCT-mod2* feature extraction. We show that the *enhanced PCA* technique retains all the positive aspects of traditional PCA (that is, robustness to compression artefacts and white Gaussian noise) while also being robust to the illumination direction change.

Several new methods, for use in fusion of speech and face information under noisy conditions, are proposed; these include a weight adjustment procedure, which explicitly measures the quality of the speech signal, and a decision stage comprised of a structurally noise resistant piece-wise linear classifier, which attempts to minimize the effects of noisy conditions via structural constraints on the decision boundary.

ACKNOWLEDGMENTS

I am profoundly grateful to Professor Kuldip K. Paliwal for accepting me as a doctoral student. Professor Paliwal's depth of knowledge, ideas and work discipline have been very inspirational. I am also grateful for his continuous patience and support during the course of my studies.

I am indebted to my family (especially Alina) for their support, understanding and help. Without them, this work would not have been possible.

I would like to thank the School of Microelectronic Engineering and Griffith University for providing a superb academic environment and financial support. My thanks also go to my friends (especially Sean Baxendell), colleagues at the Signal Processing Laboratory and visiting researchers for their invaluable suggestions and many interesting discussions, as well as the signal processing community as a whole, for helping me shape this thesis through reviewers' comments and many suggestions obtained at conferences.

Lastly, but not in the least, my thanks go to Dr Robert Davies, for his C++ matrix library[1], and to the numerous maintainers of the Linux kernel & GNU tools for providing a great operating system.

[1]The library can be obtained via www.robertnz.net

Table of Contents

List of Tables

List of Figures

NOTATION

$\vec{x}$	a column vector
$\vec{x}^T$	vector transpose of $\vec{x}$
x_i	i-th element of vector $\vec{x}$, eg. $\vec{x}^T = [\ x_1\ x_2\ ...\ x_D\ ]$, or, $\vec{x}^T = [\ x_i\ ]_{i=1}^{D}$
$\vec{x}_i$	i-th vector in a set
$\{\ \vec{x}_i\ \}_{i=1}^{N_V}$	set of N_V vectors
A^T	matrix transpose of A
A^{-1}	inverse of matrix A
$\|A\|$	determinant of matrix A
$\mathbf{\Sigma}$	covariance matrix
λ	parameter set (e.g., parameters of a GMM)

ACRONYMS

ATM	Automatic Teller Machine
BMS	Background Model Set
CMS	Cepstral Mean Subtraction
DCT	Discrete Cosine Transform
EER	Equal Error Rate
EGM	Elastic Graph Matching
EM	Expectation Maximization
ERM	Empricial Risk Minimization
FA	False Acceptance
FA%	False Acceptance rate
fps	frames per second
FR	False Rejection
FR%	False Rejection rate
GMM	Gaussian Mixture Model
MACVs	Maximum Auto-Correlation Values
MAP	Maximum *a posteriori*
MFCCs	Mel-Frequency Cepstral Coefficients
ML	Maximum Likelihood
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 Minimization
SVM	Support Vector Machine
TE	Total Error (defined as TE = FA%+FR%)
UBM	Universal Background Model
VAD	Voice Activity Detector

Automatic Person Verification Using Speech and Face Information

Chapter 1

Introduction

Identity verification (or authentication) systems pervade our every day life. For example, Automatic Teller Machines (ATMs) employ simple identity verification where the user is asked to enter their password after inserting their ATM card. If the password matches the one prescribed to the card, the user is allowed access to their bank account. Similar verification systems can be used to restrict access to rooms and buildings.

While the above verification technique is quite effective, it suffers from a major drawback: only the validity of the combination of a certain possession (in this case, the ATM card) and certain knowledge (the password) is verified. The ATM card can be lost or stolen, and the password can be compromised (e.g., somebody looks over your shoulder while you're entering it). Hence new verification methods have emerged, where the password has either been replaced by, or used in addition to, biometrics such as the person's speech, face image or fingerprints. Such physical attributes cannot be lost and vary significantly from person to person.

Apart from the applications listed above, biometrics can be applied to other areas, such as telephone & internet based banking, airline reservations & check-in and access to computer networks [22, 82, 84], as well as forensic work, where the task is to determine whether a given biometric sample belongs to a given suspect [25, 33], and law enforcement applications [12, 163].

It must be stressed that a verification system is different from an identification system: an identification system attempts to find the identity of a given person out of a pool of N people, while verification is inherently a two class task (from a security point of view this translates to: either the claimant is who he/she claims to be or he/she is an impostor).

It must also be noted that while the identification task has received considerable scientific interest, the verification task has the greatest application potential [33, 37]. Both verification and identification systems fall under the general umbrella of recognition systems.

As mentioned above, one biometric is the speech signal. 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] [33]. For this reason, this thesis concentrates on the latter category.

Speech based systems have been shown to be quite effective [112]. However, their performance easily degrades in the presence of a mismatch between training and testing conditions; usually this is in the form of channel distortion and/or ambient noise.

Another biometric is the face image; a face based system requires significantly less interaction than a speech based system, as the client does not have to do anything other than look at the camera for a few moments. While face based systems have shown to be effective for verification purposes [34], their performance can also suffer due to a mismatch between training and testing conditions; for example, a change in the illumination direction [3].

Apart from speech signals and face images, it is also possible to use biometrics such as the iris, fingerprints and hand geometry [30, 55, 120]. Yet another approach is to use more than one biometric at the same time. Such a system is known as a *multi-modal* verification system [17]; it is often comprised of several *modality experts* and a decision stage. Since a multi-modal verification system uses complementary discriminative information, lower error rates can be achieved; moreover, such a system can also be more robust, since the contribution of the modality affected by environmental conditions can be decreased.

This thesis makes several contributions aimed at increasing the robustness of single-

[1] However, one of the examiners of this thesis has pointed out that "text-dependent systems provide lower error rates and require less enrollment 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."

and multi-modal verification systems. Particularly, we increase the robustness of a speech based system subject to ambient noise; a face based system subject to illumination direction changes, compression artefacts and white Gaussian noise; and a multi-modal (speech and face) based system subject to ambient audio noise.

The rest of this chapter is organized as follows. Section 1.1 describes the organization of the thesis, provides a summary of the chapters and lists the sections which comprise the literature review. The contributions of the thesis are described in more detail in Section 1.2. Publications that resulted from the research for this thesis are listed in Section 1.3.

1.1 Thesis Organization

1.1.1 Chapter Summary

This thesis is comprised of three major parts: Speech Based Verification (Chapter 3), Face Based Verification (Chapter 5) and Fusion of Speech and Face Information (Chapter 6). It is supported by Chapter 2, which describes the Gaussian Mixture Model based classifier (which was used in experiments reported in this thesis) and Chapter 4, which describes the VidTIMIT database (which was used in experiments for face verification and fusion of speech and face information). The chapters are described in more detail below:

- **Chapter 2** begins by utilizing Bayesian Decision Theory to derive a two-class decision machine (classifier) used in the verification system. The machine is then implemented using the Gaussian Mixture Model (GMM) approach. The k-means, Expectation Maximization (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 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 verified in the following chapter.

- **Chapter 3** first reviews 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) are covered. A recently proposed feature set, termed Maximum Auto-Correlation Values (MACVs), which utilizes 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 correct implementation of the Gaussian Mixture Model classifier (described in Chapter 2) is verified. The use of MACVs is evaluated for reducing the performance degradation of a verification system used in noisy conditions.

- **Chapter 4** briefly describes two previous multi-modal databases (M2VTS and XM2VTS) and discusses their limitations. The VidTIMIT database, created by the author, is then described.

- **Chapter 5** first reviews important publications 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 are covered. Important 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, compression artefacts and white Gaussian noise.

  The chapter also evaluates the effects of likelihood normalization (which effectively modifies the decision threshold) in a face verification system subject to the above-mentioned image corruption types.

- **Chapter 6** first reviews important concepts in the field of information fusion, followed by a review of previous work on audio-visual person recognition. It is shown that the

weighted summation fusion approach is equivalent to a post-classifier[2]which utilizes a linear decision surface; the implication of this fact on the performance measurement of a multi-modal system used in noisy conditions is discussed. The performance of several standard non-adaptive fusion approaches is evaluated in noisy conditions. Several new methods for combining speech and face information in noisy conditions are proposed and evaluated.

- **Chapter 7** summarizes the work presented in this thesis and presents the main conclusions that have been drawn from the work; the chapter also suggests future research.

1.1.2 Composite Literature Review

Since this thesis covers several distinct yet related topics, each chapter (apart from the Introduction and Conclusion chapters) has its own literature review; thus the overall literature review can be considered to be comprised of:

- The whole of Chapter 2, which covers the relevant Bayesian Decision Theory necessary to build a decision machine (classifier) for a verification system, as well as the Gaussian Mixture Model (GMM) implementation and surrounding issues.
- Sections 3.2 & 3.3, which cover the speech production process and feature extraction methods, respectively.
- Section 4.2, which describes two previous multi-modal databases.
- Section 5.2, which covers important face recognition approaches (and surrounding issues) and Sections 5.3.1.1 to 5.3.1.3 which cover PCA, 2D Gabor wavelet and 2D DCT based feature extraction techniques.

[2]a *post-classifier* makes the final verification decision based on the opinions of several modality experts; it is also known as a *decision stage*.

- Section 6.2, which provides an introduction to the relatively new field of information fusion (and how it applies to person recognition) and Section 6.3, which provides an overview of important contributions in the field of audio-visual person recognition.

1.2 Contributions

The work presented in this thesis makes original contributions in several different areas; the contributions are summarized as follows:

1. Section 5.3: a new feature set (termed *DCT-mod2*), designed for facial feature extraction robust to an illumination direction change, is proposed. The feature set utilizes polynomial coefficients derived from 2D DCT coefficients of spatially neighbouring blocks.

2. Section 5.4: a study of the effects of likelihood normalization in face verification. The study also shows the effects of an illumination direction change, compression artefacts and white Gaussian noise on PCA derived features, 2D DCT features, 2D Gabor wavelet features and *DCT-mod2* features.

3. Section 5.5: an enhanced version of PCA based feature extraction, where a pseudo-image comprised of *DCT-mod2* feature vectors is used instead of the raw face image.

4. Section 5.6: modification of the *DCT-mod2* approach by increasing the number of 2D DCT blocks used in deriving each feature vector; moreover, windowing is introduced, allowing variation of the contribution of each block.

5. Section 3.4.2: a study showing that performance degradation of a verification system used in noisy conditions can be reduced through the use of a recently proposed feature set, Maximum Auto-Correlation Values (MACVs), which utilizes information from the source part of the speech signal.

6. Sections 6.4 & 6.5. Section 6.4 demonstrates the equivalence of the weighted summation fusion approach to a post-classifier which utilizes a linear decision

surface, while Section 6.5 discusses the implication of the above equivalence on the measurement of performance of a multi-modal verification system used in noisy conditions.

7. Section 6.7.1: a weight adjustment procedure for use in weighted summation fusion of the opinions of speech and face experts; the procedure explicitly measures the quality of the speech signal.

8. Section 6.7.2: a modification to the Bayesian post-classifier, which allows adjustment of the degree of contribution of each expert to the final verification decision.

9. Section 6.8.1: a structurally noise resistant piece-wise linear post-classifier, which attempts to minimize the effects of noisy conditions via structural constraints on the decision boundary.

10. Section 6.8.2: a modification to the Bayesian post-classifier, which also attempts to impose structural constraints on the decision boundary.

11. Chapter 4: an audio-visual, multi-session database (known as VidTIMIT) for use in person verification experiments.

1.3 Publications Resulting from Research for this Thesis

This thesis has in many parts been shaped by colleagues' and reviewers' comments regarding many of the publications listed below; it has also been shaped by the comments and suggestions resulting from conference presentations.

1.3.1 Journal Articles

1. C. Sanderson and 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.

2. C. Sanderson and K. K. Paliwal, "Features for Robust Face-based Identity Verification", *Signal Processing*, Vol. 83, No. 5, 2003, pp. 931-940.

3. C. Sanderson and K. K. Paliwal, "Fast Feature Extraction Method for Robust Face Verification", *IEE Electronics Letters*, Vol. 38, No. 25, 2002, pp. 1648-1650.

4. C. Sanderson and K. K. Paliwal, "Automatic Person Verification Using Speech and Face Information", submitted to *Digital Signal Processing* on 4-Aug-2002.

5. C. Sanderson and K. K. Paliwal, "Structurally Noise Resistant Classifier for Multi-Modal Person Verification", submitted to *Pattern Recognition Letters* on 21-Jun-2002.

6. C. Sanderson and K. K. Paliwal, "Likelihood Normalization for Face Verification in Variable Image Conditions", submitted to *Image and Vision Computing* on 19-Mar-2002.

7. C. Sanderson and K. K. Paliwal, "Fast Features for Face Authentication Under Illumination Direction Changes", submitted to *Pattern Recognition Letters* on 7-Feb-2002.

1.3.2 Conference Papers

1. C. Sanderson and K. K. Paliwal, "Likelihood Normalization for Face Authentication in Variable Recording Conditions", *Proc. International Conf. on Image Processing*, Rochester, 2002, pp. 301-304 (Vol. 1).

2. C. Sanderson and K. K. Paliwal, "Polynomial Features for Robust Face Authentication", *Proc. International Conf. on Image Processing*, Rochester, 2002. pp. 997-1000 (Vol. 3).

3. C. Sanderson and K. K. Paliwal, "Information Fusion for Robust Speaker Verification", *Proc. 7th European Conf. Speech Communication and Technology*, Aalborg, 2001, pp. 755-758.

4. C. Sanderson and K. K. Paliwal, "Noise Compensation in a Multi-Modal Verification System", *Proc. International Conf. on Acoustics, Speech and Signal Processing*, Salt Lake City, 2001, pp. 157-160 (Vol. 1).

5. C. Sanderson and K. K. Paliwal, "Robust Face-Based Identity Verification", *Proc. Microelectronic Engineering Research Conf.* Brisbane, Australia, 2001.

6. C. Sanderson and K. K. Paliwal, "Training Method of a Piecewise Linear Classifier for a Multi-Modal Person Verification System", *Proc. Eighth Australian International Conf. on Speech Science and Technology*, Canberra, 2000, pp. 312-317.

7. C. Sanderson and K. K. Paliwal, "Adaptive Multi-Modal Person Verification System", *Proc. First IEEE Pacific-Rim Conf. on Multimedia*, Sydney, 2000, pp. 210-213.

8. C. Sanderson and K. K. Paliwal, "Multi-Modal Person Verification System Based on Face Profiles and Speech", *Proc. Fifth International Symposium on Signal Processing and its Applications*, Brisbane, 1999, Vol. 2, pp. 947-950.

Chapter 2

Gaussian Mixture Model Based Classifier

2.1 Abstract

In this chapter the Bayesian Decision Theory is used to derive a two-class decision machine (classifier) used in a verification system. The machine is then implemented using the Gaussian Mixture Model (GMM) approach. The k-means, Expectation Maximization (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). Next, error measures for finding the performance of a verification system 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 verified in the following chapter (Chapter 3).

2.2 Bayesian Decision Theory

A verification system, on the fundamental level, is 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, 35, 116] to implement the decision machine.

Let us denote *client specific* true claimant and impostor classes as C_1 and C_2,

respectively, and let $\vec{x} = [x_1\ x_2\ ...\ x_D]^T$ be the observation vector. Moreover, let $P(C_j)$ be the *a priori* probability of class C_j, and $p(\vec{x}|C_j)$ be the conditional probability density function (pdf) of $\vec{x}$, given class C_j. We seek to find the class that $\vec{x}$ belongs to. Using the *Bayes formula* [15, 85], we obtain:

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

where

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

Thus using the *Bayes formula* we obtain the *a posteriori* probability of C_j. It follows that the *Bayes decision rule* is then:

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

Or, more generally,

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

which is known as the *maximum a posteriori* decision rule. It must be noted that $p(\vec{x})$ is not required for making the decision - thus the decision rule becomes:

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

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

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

In practice, the true form of the pdf $p(\vec{x}|C_j)$ is unknown - hence a parametric representation, $\tilde{p}(\vec{x}|C_j)$, estimated from training data, is used instead. Since $\tilde{p}(\vec{x}|C_j)$ is only an

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

approximation[2], a "correction" function, $\dot{p}(\vec{x}|C_j)$, is required:

$$p(X|C_j) = \prod_{i=1}^{N_V} \tilde{p}(\vec{x}|C_j)\dot{p}(\vec{x}|C_j)P(C_j) \tag{2.7}$$

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

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

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

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

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:

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

which translates to:

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

where, for clarity,

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

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

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

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

Let us define

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

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

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

Let us define:

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

Since the true form of the pdf $p(\vec{x}|C_j)$ is unknown, the "correction" function, $\dot{p}(\vec{x}|C_j)$, is also unknown; moreover, in real life situations the *a priori* probabilities $P(C_1)$ and $P(C_2)$ are often not known. Thus in practice, $\frac{1}{N_V}\log\left[\frac{\dot{p}(X|C_2)P(C_2)}{\dot{p}(X|C_1)P(C_1)}\right]$ is replaced with an experimentally 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 normalization 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.

2.3 Gaussian Mixture Model

$\tilde{p}(\vec{x}|C_j)$ is represented by a Gaussian Mixture Model (GMM), capable of modeling arbitrarily complex densities [110]. For client K, $\mathcal{L}(X|C_2)$ is replaced by $\mathcal{L}(X|\lambda_{\overline{K}})$ (defined in Section 2.3.2) 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

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

$$p(\vec{x}|\lambda) = \sum_{k=1}^{N_G} m_k \, \mathcal{N}(\vec{x}; \vec{\mu}_k, \mathbf{\Sigma}_k) \tag{2.20}$$

$$\tag{2.21}$$

and

$$\lambda = \{m_k, \vec{\mu}_k, \mathbf{\Sigma}_k\}_{k=1}^{N_G} \tag{2.22}$$

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

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

2.3.1 Maximum Likelihood Parameter Estimation

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

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

The estimation problem can be solved using an iterative version of the Expectation Maximization (EM) algorithm [31, 109, 87]. The full derivation of the EM algorithm for GMM parameter estimation is beyond the scope of this chapter - the reader is encouraged to refer to [18, 109, 110] or see Appendix C.

The EM algorithm is comprised of iterating two steps: the *expectation* step, followed by the *maximization* 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 [35] (described in Section 2.3.1.1). The EM algorithm is implemented as follows:

Expectation step:

$$\text{for } k = 1, ..., N_G: \quad \text{for } i = 1, ..., N_V: \quad l_{k,i} = \frac{m_k \mathcal{N}(\vec{x}_i; \vec{\mu}_k, \mathbf{\Sigma}_k)}{\sum_{n=1}^{N_G} m_n \mathcal{N}(\vec{x}_i; \vec{\mu}_n, \mathbf{\Sigma}_n)} \tag{2.26}$$

$$\text{for } \quad k = 1, ..., N_G:$$

$$L_k = \sum_{i=1}^{N_V} l_{k,i} \tag{2.27}$$

$$\hat{m}_k = \frac{L_k}{N_V} \tag{2.28}$$

$$\hat{\vec{\mu}}_k = \frac{1}{L_k} \sum_{i=1}^{N_V} \vec{x}_i \; l_{k,i} \tag{2.29}$$

$$\hat{\mathbf{\Sigma}}_k = \frac{1}{L_k} \sum_{i=1}^{N_V} (\vec{x}_i - \hat{\vec{\mu}}_k)(\vec{x}_i - \hat{\vec{\mu}}_k)^T l_{k,i} \tag{2.30}$$

$$= \frac{1}{L_k} \left[\sum_{i=1}^{N_V} \vec{x}_i \vec{x}_i^T l_{k,i} \right] - \hat{\vec{\mu}}_k \hat{\vec{\mu}}_k^T \tag{2.31}$$

Maximisation step:

$$\{m_k, \vec{\mu}_k, \mathbf{\Sigma}_k\}_{k=1}^{N_G} = \{\hat{m}_k, \hat{\vec{\mu}}_k, \hat{\mathbf{\Sigma}}_k\}_{k=1}^{N_G} \tag{2.32}$$

In the E-Step, $l_{k,i} \in [0, 1]$ is the *a posteriori* probability of Gaussian k given $\vec{x}_i$ and current parameters. Thus the estimates $\hat{\vec{\mu}}_k$ and $\hat{\mathbf{\Sigma}}_k$ are merely weighted versions of the sample mean and sample covariance, respectively. For a full derivation of the EM algorithm for GMM parameters, the reader is directed to [18, 110] or Appendix C.

Overall, the algorithm is a hill climbing procedure for maximizing $p(X|\lambda)$. While it may not reach a global maximum, it is guaranteed to monotonically converge to a saddle point

or a local maximum [31, 35, 91]. It must also be noted that 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 initialized to sensible quasi-random values[3], faster convergence can be achieved when the initial estimate is provided via the k-means clustering algorithm [35], described in the following section.

2.3.1.1 k-means

We utilize the Kronecker delta function, $\delta(\cdot,\cdot)$, which has a value of 1 if its two arguments match, and the $rand(min, max)$ function, which generates a uniformly distributed random value in the $[min, max]$ interval. The k-means algorithm is described using the following pseudo-code:

for $k = 1, ..., N_G$: $\vec{\mu}_k = \vec{x}_{rand(1,N_V)}$ // randomly select initial means
loop = 0
endloop = $10 \times N_G$ // empirically chosen termination condition (see Sec.2.5.2)
finished = FALSE
do
  for $i = 1, ..., N_V$: $y_i = \arg \min_{k=1,...,N_G} ||\vec{\mu}_k - \vec{x}_i||$ // label each vector as belonging to its closest mean
  for $k = 1, ..., N_G$
    $N_k = \sum_{i=1}^{N_V} \delta(y_i, k)$ // count the number of vectors assigned to each mean
    $\hat{\vec{\mu}}_k = \frac{1}{N_k} \sum_{i=1}^{N_V} \vec{x}_i \delta(y_i, k)$ // find the new mean using vectors assigned to the old mean
  same = TRUE
  for $k = 1, ..., N_G$: if $\hat{\vec{\mu}}_k \neq \vec{\mu}_k$ then *same* = FALSE
  if *same* = TRUE then *finished* = TRUE // if the means haven't changed since last iteration, finish
  loop = *loop* + 1
  if *loop* $\geq$ *endloop* then *finished* = TRUE
  for $k = 1, ..., N_G$: $\vec{\mu}_k = \hat{\vec{\mu}}_k$ // overwrite old means with new means
until *finished* = TRUE

[3]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.

Once the estimated means, $\{ \vec{\mu}_k \}_{k=1}^{N_G}$, have been found, the estimated weights, $\{ m_k \}_{k=1}^{N_G}$, and covariance matrices, $\{ \mathbf{\Sigma}_k \}_{k=1}^{N_G}$, are found as follows:

$$m_k = \frac{N_k}{N_V} \tag{2.33}$$

$$\mathbf{\Sigma}_k = \frac{1}{N_k} \sum_{i=1}^{N_V} (\vec{x}_i - \vec{\mu}_k)(\vec{x}_i - \vec{\mu}_k)^T \delta(y_i, k) \tag{2.34}$$

It must be noted that the k-means algorithm can also be implemented in a different manner; for example, the "splitting" LBG algorithm [76].

2.3.2 Impostor Likelihood

2.3.2.1 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 [112, 118] (also known as cohort models [39, 117]). 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, $B = \{\lambda_b\}_{b=1}^{N_B}$:

$$\mathcal{L}(X|\lambda_{\overline{K}}) = \log \left[\frac{1}{N_B} \sum_{b=1}^{N_B} \exp \mathcal{L}(X|\lambda_b) \right] \tag{2.35}$$

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

In this thesis we have utilized the method described by Reynolds [112] to select the BMS for each client; the method is summarized 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:

$$d(\lambda_D, \lambda_E) = [\mathcal{L}(X_D|\lambda_D) - \mathcal{L}(X_D|\lambda_E)] + [\mathcal{L}(X_E|\lambda_E) - \mathcal{L}(X_E|\lambda_D)] \tag{2.36}$$

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 set Φ. Similarly, N_Ψ farthest models ($N_\Psi \geq N_B$) are placed in 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:

$$\lambda_i = \arg\max_{\lambda_j \in \Phi} \left[\sum_{\lambda_b \in B_{close}} \frac{d(\lambda_b, \lambda_j)}{d(\lambda_K, \lambda_j)} \right] \tag{2.37}$$

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:

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

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 [112].

2.3.2.2 Universal Background Model

An alternative approach to approximate the impostor model is via the use of the Universal Background Model (UBM). In this approach, pooled training data from *all* clients is utilized to construct a large mixture model as per Section 2.3.1. The average log likelihood that the claim for person K's identity is from an impostor is calculated using:

$$\mathcal{L}(X|\lambda_{\overline{K}}) = \mathcal{L}(X|\lambda_{UBM}) \tag{2.39}$$

The advantage is that the impostor likelihood is now client independent (as opposed to the BMS approach). Moreover, it has been found [114] that instead of constructing the client models directly from training data (using the EM algorithm), lower error rates can be obtained (on a large database) when the models are generated by adapting the UBM using a form of *maximum a posteriori* (MAP) adaptation [43, 115].

A full description of MAP adaptation is out of the scope of this chapter (the reader is encouraged to refer to [42, 43, 57, 115]). The update equations are summarized as follows. Given UBM parameters $\lambda_{UBM} = \{\dot{m}_k, \dot{\vec{\mu}}_k, \dot{\mathbf{\Sigma}}_k\}_{k=1}^{N_G}$ and a set of training feature vectors for a specific client, $X = \{\vec{x}_i\}_{i=1}^{N_V}$, the estimated weights ($\hat{m}_k$), means ($\hat{\vec{\mu}}_k$), and covariances ($\hat{\mathbf{\Sigma}}_k$) are found as per Eqns. (2.28)-(2.31). The final parameters, $\lambda = \{m_k, \vec{\mu}_k, \mathbf{\Sigma}_k\}_{k=1}^{N_G}$, are found using:

$$m_k = [\alpha\hat{m}_k + (1-\alpha)\dot{m}_k]\,\gamma \tag{2.40}$$

$$\vec{\mu}_k = \alpha\hat{\vec{\mu}}_k + (1-\alpha)\dot{\vec{\mu}}_k \tag{2.41}$$

$$\mathbf{\Sigma}_k = \left[\alpha\left(\hat{\mathbf{\Sigma}}_k + \hat{\vec{\mu}}_k\hat{\vec{\mu}}_k^T\right) + (1-\alpha)\left(\dot{\mathbf{\Sigma}}_k + \dot{\vec{\mu}}_k\dot{\vec{\mu}}_k^T)\right)\right] - \vec{\mu}_k\vec{\mu}_k^T \tag{2.42}$$

where γ is a scale factor to make sure all weights sum to 1. $\alpha = \frac{L_k}{L_k+r}$ is a data-dependent adaptation coefficient (L_k is found using Eqn. (2.27)) where r is a fixed relevance factor (typically $r \in [8, 20]$, see [115]). It must be noted that UBM mixture components will only be 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.

2.4 Error Measures

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. Thus the performance is measured in terms of False Acceptance rate (FA%) and False Rejection rate (FR%), defined as:

$$\text{FA\%} = \frac{I_A}{I_T} \times 100\% \tag{2.43}$$

$$\text{FR\%} = \frac{C_R}{C_T} \times 100\% \tag{2.44}$$

where I_A is the number of impostors classified as true claimants, I_T is the total number of impostor classification tests, C_R is the number of true claimants classified as impostors, and C_T is the total number of true claimant classification tests.

Since the errors are related, minimizing the FA% increases the FR% (and vice versa). The trade-off between FA% and FR% is adjusted 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 FA% as low as possible, even at the expense of a high FR%.

The trade-off between FA% and FR% can be graphically represented by a Receiver Operating Characteristics (ROC) plot or a Detection Error Trade-off (DET) plot [33]. The ROC plot is on a linear scale, while the DET plot is on a log scale (which can improve the visual appearance of the curves). In both cases the FR% is plotted as a function of FA%.

To quantify the performance into a single number, Equal Error Rate (EER), is often used [39]. Here the system is configured to operate with FA% = FR%.

It must be noted that the threshold is adjusted to obtain desired performance on *test* data (data unseen by the system up to this point). Such a threshold is known as the *a posteriori* threshold. However, if the threshold is fixed before finding the performance, the threshold is known as the *a priori* threshold [38]. The *a priori* threshold can be found via experimental means using training data or *evaluation* data (data which has also been unseen by the system up to this point, but is separate from *test* data).

Logically, the *a priori* threshold is more realistic. However, it is often difficult to find a reliable *a priori* threshold [38, 33]. The *test* section of a database is often divided into two sets: *evaluation* data and true *test* data. If the *evaluation* data is not representative of the *test* data, then the *a priori* threshold will achieve significantly different results on *evaluation* and *test data*. Moreover, such a database division reduces the number of verification tests, thus decreasing the statistical significance of the results. For these reasons, many researchers prefer to use the *a posteriori* threshold and interpret the performance obtained as the *expected* performance.

In keeping with tradition, the *a posteriori* threshold (set to obtain EER performance) is used in all *single biometric* verification experiments throughout this thesis.

2.5 Implementation Issues

2.5.1 EM Algorithm

Reynolds [110] showed 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 is computationally expensive, the maximum number of iterations has been limited to 10 in all experiments reported in this thesis.

2.5.2 k-means

The k-means algorithm is used to provide an initial estimate of the GMM parameters $\lambda = \{m_k, \vec{\mu}_k, \mathbf{\Sigma}_k\}_{k=1}^{N_G}$ 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 $10 \times N_G$. 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 $||\vec{a} - \vec{b}||$ operation which is the Euclidean distance between $\vec{a}$ and $\vec{b}$. To prevent one of the dimensions dominating the result (due to a relatively large variance), it is necessary to first transform the training data ($X = \{\vec{x}_i\}_{i=1}^{N_V}$) so each dimension has zero mean and unit variance:

$$\dot{X} = \left\{ \; T(\vec{x}_i)^T \; \right\}_{i=1}^{N_V} \tag{2.45}$$

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

$$T(\vec{x}) \quad = \quad [\ (1/\sigma_d)(x_d - \mu_d)\ ]_{d=1}^{D} \tag{2.46}$$

and the corresponding inverse function is:

$$T^{-1}(\dot{\vec{x}}) \quad = \quad [\ \sigma_d \dot{x}_d + \mu_d\ ]_{d=1}^{D} \tag{2.47}$$

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, the inverse transformation $T^{-1}(\cdot)$ is applied to them before the estimated covariances are calculated [Eqn. (2.34)] using the original data (X).

2.5.3 Impostor Likelihood

When using the BMS approach to calculate the impostor likelihood, one would like to use as many background speakers 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 experimental conditions, the size of the BMS is limited to 10 models.

If there is little statistical correspondence between the UBM and client training data, the final model is largely similar to the UBM. This will result in poor verification performance and strongly suggests that a large data set is required for training. Moreover, the need to represent the impostor population reliably implies a large training data set. In practice

this is not easily achieved, due to the size of experimental databases (number of clients and corresponding training data), as well as computational limitations of processing a large data set (amount of memory, hard drive space and processing speed). Thus for most of the experiments reported in this thesis, the BMS approach is used for calculating the impostor likelihood.

2.5.4 Type of Covariance Matrix

The general definition of a GMM (see Section 2.3) supports full covariance matrices, i.e., a covariance matrix with all its elements. However, like many researchers, in this thesis 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 modeling using an N_G-Gaussian full covariance GMM can be equally well achieved using a larger mixture diagonal covariance GMM; moreover, diagonal covariance GMMs with $N_G > 1$ can model distribution of feature vectors with correlated elements [115].

- Using diagonal covariance matrices reduces the number of unknown parameters; thus less training data is required than for full covariance matrices [35].

- It has been empirically observed that diagonal covariance GMMs outperform full covariance GMMs [112, 113, 115].

Chapter 3

Speech Based Verification

3.1 Abstract

In this chapter we first review 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) are covered. A recently proposed feature set, termed Maximum Auto-Correlation Values (MACVs), which utilizes 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 database 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 [112]. 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.

Publications resulting from this research: [130, 135].

3.2 Introduction

3.2.1 Speech Production Process

Speech can be categorized 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 modeled 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., [Λ] vs [ε][1]). 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) [124].

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 forces the vocal folds apart. Shortly afterward the air pressure is temporarily equalized, and the vocal folds close again, completing the cycle. The cycle occurs at a typical frequency of 60-160 Hz for males and 160-400 Hz for females [103, 56] (average values are 132 Hz and 223 Hz for males and females, respectively [143]). 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.

[1] Here we use the International Phonetic Alphabet [51]; the sound [Λ] 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*.

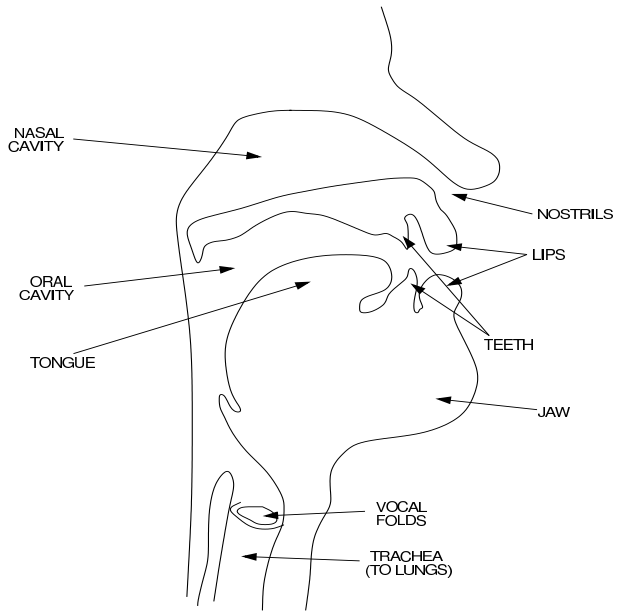

Figure 3.1: Major vocal tract components (after [124])

During the production of unvoiced sounds, the vocal folds do not vibrate. 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*, where 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 emphasized, 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 [124].

3.2.2 Automatic Speaker Verification

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) [8, 33, 112, 115]. 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 [38, 125] and Cepstral Mean Subtraction (CMS) [38].

Recently Wildermoth and Paliwal [161] proposed a new feature set, termed Maximum Auto-Correlation Values (MACVs), which utilizes 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.

3.2.3 Chapter Organization

The rest of this chapter is organized as follows. In Section 3.3 we describe the MFCC, CMS, delta and MACV speech feature extraction techniques, as well as a parametric Voice Activity Detector (VAD) used for disregarding silence and noise segments of the speech signal. Section 3.4 is devoted to experiments confirming the correct implementation of the GMM classifier and the MFCC feature extractor, as well as evaluating the use of MACV features to reduce the effects of mismatched conditions.

3.3 Feature Extraction Methods

3.3.1 MFCC Features

In MFCC feature extraction, the speech signal is analyzed 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 [97].

Let us denote the speech frame as $\vec{s}^T = [\ s_i\ ]_{i=1}^{N_S}$, 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 [105]:

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

where

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

The complex spectrum of $\vec{\hat{s}}^T = [\ \hat{s}_i\ ]_{i=1}^{N_S}$ is then obtained using the Fast Fourier Transform (FFT) algorithm [104, 105]. The square of the magnitude of the complex spectrum is represented as $\vec{S}$ (in our experiments we use a 2048 point representation).

A set of triangular-shaped filters is spaced according to the Mel-scale [100], simulating the processing done by the human ear [56, 92, 93]. For filters chosen to cover the telephone bandwidth, the center 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 [100], the upper and lower passband frequencies of each filter are the center frequencies of adjacent filters. For the filter centered at 300 Hz, the lower passband frequency is 200 Hz, while the upper passband frequency for the filter centered at 3482 Hz is 4000 Hz. The responses of $N_F = 17$ filters are shown in Figure 3.2.

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

$$e_i = \vec{f_i}^T \vec{S}, \qquad i = 1, 2, ..., N_F \tag{3.3}$$

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

$$\vec{e} = F^T \vec{S} \tag{3.4}$$

where $F = [\ \vec{f_1}\ \vec{f_2}\ ...\ \vec{f}_{N_F}\ ]$. It must be noted that Eqn. (3.4) can be interpreted as a form of dimensionality reduction. In effect, the energy vector $\vec{e}$ represents the smoothed (Mel-warped) spectrum of $\vec{s}$, which is a good representation of the filter part of speech [124].

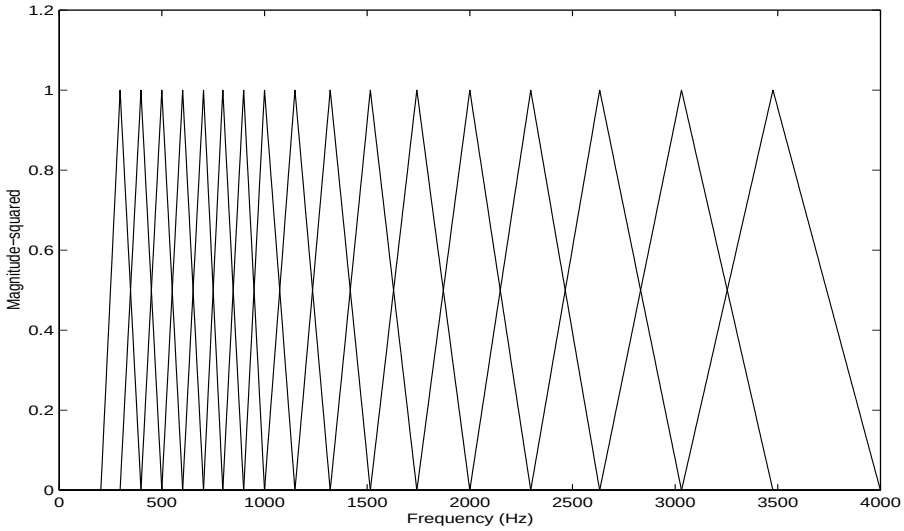

Figure 3.2: Mel-scale filter bank

In order to obtain amplitude normalization, as well as to take into account the diagonal covariance matrix constraint in the GMM classifier (see Section 2.5.4), a form of 1D Discrete Cosine Transform (1D DCT) [41] is applied to the log version of $\vec{e}$:

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

One reason for using the log version of $\vec{e}$ is explained in Section 3.3.2. Eqn. (3.5) can be rewritten in matrix notation:

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

where

$$\vec{e}^T_{\log} = [\ \log(e_i)\ ]_{i=1}^{N_F} \tag{3.7}$$

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

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

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:

$$\vec{x} \;=\; [\; g_2 \; g_3 \; ... \; g_{N_F} \;]^T \tag{3.9}$$

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

Another popular speech feature extraction method is based on Linear Prediction Cepstral Coefficients (LPCC) [97], which originated from speech compression applications [10, 56, 81]. However, MFCC features are used for experiments in this thesis since it has been shown that they are generally more robust than LPCC features for speaker recognition applications [111].

3.3.2 CMS Features

Let us assume that a signal z is comprised of an original speech signal a that is being filtered by a channel[2] 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. Because the energy vector $\vec{e}$ from Eqn. (3.4) represents the smoothed (Mel-warped) spectrum, Eqn. (3.11) is analogous to:

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

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

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

[2]For example, a telephone channel.

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

$$\vec{g} = C^T\left(\vec{e}^{\,a}_{\log} + \vec{e}^{\,b}_{\log}\right) \quad (3.15)$$

$$= C^T\vec{e}^{\,a}_{\log} + C^T\vec{e}^{\,b}_{\log} \quad (3.16)$$

$$= \vec{g}^{\,a} + \vec{g}^{\,b} \quad (3.17)$$

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

$$\vec{x} = \vec{x}^{\,a} + \vec{x}^{\,b} \quad (3.18)$$

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

$$\vec{x}^{\,\mu} = \frac{1}{N_V}\sum_{i=1}^{N_V}\vec{x}_i \quad (3.19)$$

$$= \frac{1}{N_V}\sum_{i=1}^{N_V}\left(\vec{x}_i^{\,a} + \vec{x}_i^{\,b}\right) \quad (3.20)$$

$$= \frac{1}{N_V}\sum_{i=1}^{N_V}\vec{x}_i^{\,a} + \frac{1}{N_V}\sum_{i=1}^{N_V}\vec{x}_i^{\,b} \quad (3.21)$$

Assuming that channel characteristics are time invariant leads to:

$$\vec{x}^{\,\mu} = \frac{1}{N_V}\sum_{i=1}^{N_V}\vec{x}^{\,a} + \vec{x}^{\,b} \quad (3.22)$$

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 $\frac{1}{N_V}\sum_{i=1}^{N_V}\vec{x}^{\,a}$ tends toward zero [13]. Thus $\vec{x}^{\,b}$ can be found using Eqn. (3.19) and we can obtain channel compensated vectors using:

$$\{\vec{x}_i^{\,\text{comp}}\}_{i=1}^{N_V} = \{\vec{x}_i - \vec{x}^{\,\mu}\}_{i=1}^{N_V} \quad (3.23)$$

The above procedure is known as Cepstral Mean Subtraction (CMS) and Cepstral Mean Normalization (CMN) [11, 13, 38, 111, 113].

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, 45], 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 [52, 53], can be used.

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

3.3.3 Delta 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 [125]. Given a sequence of instantaneous spectrum feature vectors, $\{\vec{x}_i\}_{i=1}^{N_V}$, the corresponding transitional spectrum feature vectors are calculated using a modified 1st order orthogonal polynomial fit [38, 60, 125]:

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

where $\vec{h}$ is a $2K+1$ dimensional symmetric window vector. Typically, $K = 2$ and a rectangular window is used [8, 113, 115] (thus $\Delta\vec{x}_i$ 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 [113].

While being more robust to channel effects, delta features do not perform as well as static features in matched conditions [125]. Thus it is general practice to combine the two feature sets by concatenating the delta feature vector with the static feature vector:

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

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:

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

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

Delta-delta (or acceleration) feature vectors ($\Delta\Delta\vec{x}$) can be obtained by applying Eqn. (3.24) to delta feature vectors. However, use of delta-delta features has shown no measurable improvement in speaker verification performance [33].

3.3.4 MACV Features

In MFCC features (and hence CMS and delta features) only the system part of the speech signal is effectively utilized. There can be two ways of utilizing pitch (or pitch-related) information:

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 autocorrelation function, which for a speech frame $\vec{s}^T = [\ s_i\ ]_{i=1}^{N_S}$ is defined as [105]:

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

If $\vec{s}$ 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 [56, 103], indicating that valid pitch lags are approximately between 2.5ms and 16ms. Thus the period of $\vec{s}$ 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 Normalized Cross-Correlation Method [9] and the Average Magnitude Difference Method [94, 119]), suffer from pitch doubling and halving as well as other errors [56].

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 may 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 [56], indicating that their output cannot be incorporated into the feature vector for each frame.

The recently proposed Maximum Auto-Correlation Value (MACV) feature set [161] overcomes 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. Normalize $\{R(k)\}_{k=0}^{N_S-1}$ by its maximum, i.e., $\left\{\hat{R}(k)\right\}_{k=0}^{N_S-1} = \left\{\frac{R(k)}{R(0)}\right\}_{k=0}^{N_S-1}$.
3. Divide the higher portion (from 2.5 ms to 16 ms) of $\{\hat{R}(k)\}_{k=0}^{N_S-1}$ into N_M equal parts (typically, $N_M = 5$ [161]).

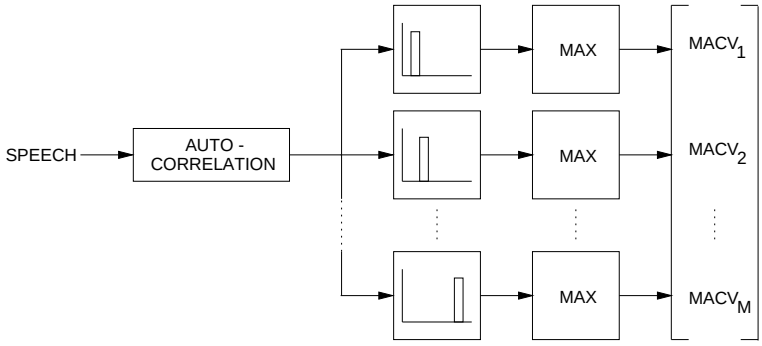

Figure 3.3: MACV feature extractor (after [161])

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 autocorrelation 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 3.4.2.

3.3.5 Voice Activity Detector

In addition to pauses between words, the start and the end of speech signals in many databases often contains only background noise. Since these segments do not contain speaker dependent information, it would be advantageous to disregard them during modeling 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 [110] (seemingly used in his following work, i.e., [112, 113, 114, 115]) we have developed a parametric VAD based on the work by Haigh [47, 48].

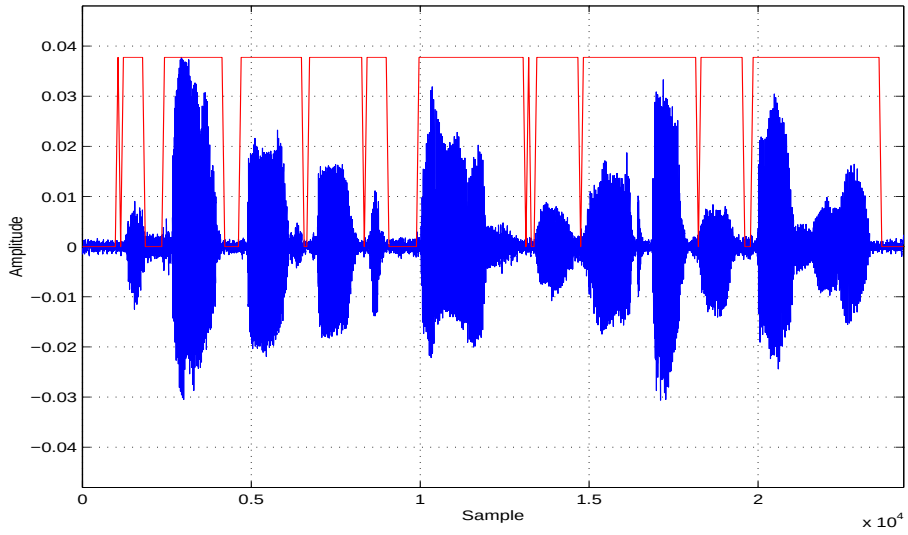

Figure 3.4: Typical result of speech selection using the parametric VAD; high level of the red line indicates the segments that have been selected as speech. The above utterance is: *before thursday's exam, review every formula.*

The parametric VAD is implemented as follows. Each utterance is completely parameterized using a given feature extraction technique, resulting in a set of feature vectors, $X = \{\vec{x}_i\}_{i=1}^{N_V}$. A single Gaussian GMM (representing the background noise) is constructed using the first N_{noise} vectors[3]. 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 experimentally found to provide good discrimination ability across various parameterization methods:

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

where

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

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]For the NTIMIT database [61], $N_{\text{noise}} = 10$.

3.4 Experiments

3.4.1 Verification of Correct GMM and MFCC Implementation

In this section we verify the implementation of the Gaussian Mixture Model classifier (described in Chapter 2) and the MFCC feature extractor by comparing the results obtained with the results published by Reynolds in [112].

Reynolds' experimental setup is as follows. Speech signals are taken from the *test* section of the telephone speech NTIMIT database [61], 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.

32-Gaussian GMMs were used as client models. For each client, his/her 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 2.3.2.1). 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, indicating that the GMM classifier and the MFCC feature extractor were implemented correctly. The negligible difference could be attributed to the tuning of the VAD.

Source	*EER (%)*
Reynolds [112]	7.19
This thesis	7.22

Table 3.1: Comparison of EER

3.4.2 Evaluation of MACVs in Noisy Conditions

Speech signals were taken from the *test* section of the telephone speech NTIMIT database [61], 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).

20 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 [112], the BMS for each client was comprised of 10 models (N_Φ and N_Ψ were set to 20; see Section 2.3.2.1); the BMS was constructed by considering the other 147 client models. 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 EER.

In the first experiment we studied the effect of the number of Gaussians on verification performance while utilizing 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 [35, 91]. Overfitting is said to occur when the classifier is "too tuned" to the training data, resulting in poor generalization on test data. Taking into account Occam's Razor principle [35, 91], which in effect pleads for 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

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 for varying number of Gaussians (MFCC parameterization)

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 ($\vec{x}$) has been concatenated with the feature vector containing delta versions of the MFCC features ($\Delta\vec{x}$). 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[4] 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 through 3.7.

In Figure 3.5 it can be seen that the CMS features are the least affected by changes in the SNR, at the expense of slightly worse performance than MFCC features on clean speech (as expected; see Section 3.3.2). 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

[4]SNR (dB) $= 10\log_{10}\frac{\sum_i s_i^2}{\sum_i (s_i - n_i)^2}$ where s_i and n_i are the i-th samples from the original and noisy speech signals, respectively.

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 3.3.4, and suggest that use of the MACV feature set has beneficial effects on the performance of a verification system in noisy conditions.

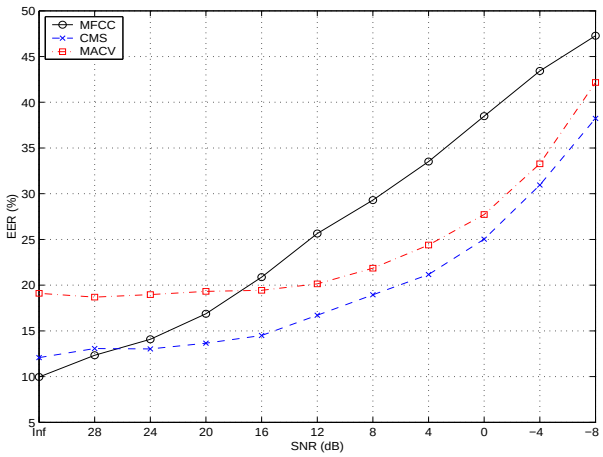

Figure 3.5: Performance of baseline features

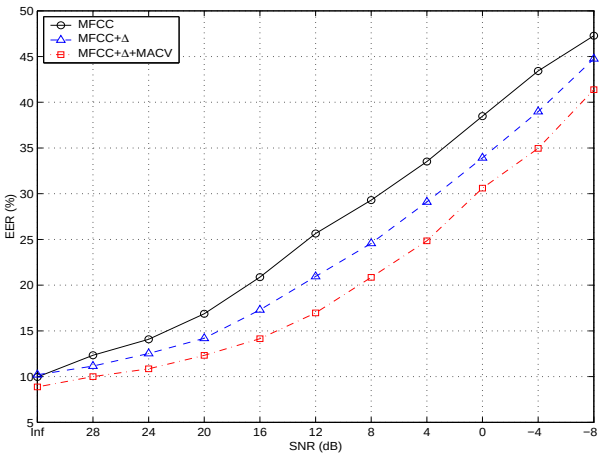

Figure 3.6: Performance of MFCC based features

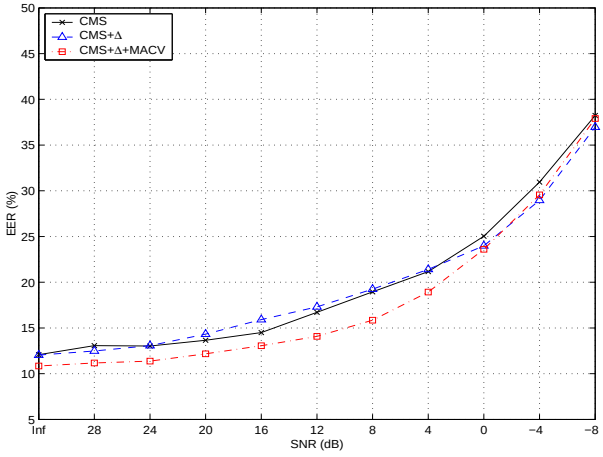

Figure 3.7: Performance of CMS based features

3.5 Summary

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. A recently proposed feature set, termed Maximum Auto-Correlation Values (MACVs), which utilizes 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 database 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 [112]. 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.

Chapter 4

VidTIMIT database

4.1 Abstract

In this chapter two previous multi-modal databases, M2VTS and XM2VTS, are briefly described. Their limitations are discussed, such as the size and cost. The VidTIMIT database, created by the author while taking into account the problems with M2VTS and XM2VTS databases, is then described.

4.2 M2VTS and XM2VTS databases

At the start of research for this thesis, only one widely distributed multi-modal database existed, namely the M2VTS database [101]. 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 very 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 labeled 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) database [90], 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:

1. "0 1 2 3 4 5 6 7 8 9"
2. "5 0 6 9 2 8 1 3 7 4"
3. "Joe took fathers green shoe bench out"

While the number of subjects results in a much larger number of verification tests, the database is inherently suited for development 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 hardly representative of the test data - easily leading to poor performance.

At the time of release, the XM2VTS database was quite expensive to obtain. Moreover, it was distributed on DVD-RAM media at a time when the DVD-RAM drives were quite expensive and not widely available. Due to financial limitations, we were not able to obtain the XM2VTS database.

Taking into account the problems with the M2VTS and XM2VTS databases, the author has created the VidTIMIT database, described in the following section.

4.3 VidTIMIT database

The VidTIMIT database, created by the author, 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 [61]. There are 10 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.

A typical example of the sentences used is in Table 4.1. There is complete correspondence of the subject IDs between VidTIMIT and NTIMIT (and hence the recited sentences). Copyright restrictions on the NTIMIT corpus prevent the list of all sentences used in the VidTIMIT database.

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 center, up, then down and finally return to center.

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 $\times$ 512 pixels (rows $\times$ columns). 90% quality setting was used during the creation of the JPEG images. The corresponding audio[1] is stored as a mono, 16 bit, 32 kHz WAV file. The entire database occupies approximately 3.5 Gb and is distributed on six CD-ROMs.

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 test and train sections. The database is thus suited for the development

[1]The audio was recorded using the camera's microphone.

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 4.1: Typical example of sentences used in the VidTIMIT database

Examples images of all subjects are shown in Figures 4.1 through 4.9. The first, second and third columns represent images taken in Session 1, 2 and 3, respectively.

Figure 4.1: Subjects in the VidTIMIT database (Part A). The first, second and third columns represent images taken in Session 1, 2 and 3, respectively.

Figure 4.2: Subjects in the VidTIMIT database (Part B)

Figure 4.3: Subjects in the VidTIMIT database (Part C)

Figure 4.4: Subjects in the VidTIMIT database (Part D)

Figure 4.5: Subjects in the VidTIMIT database (Part E)

Figure 4.6: Subjects in the VidTIMIT database (Part F)

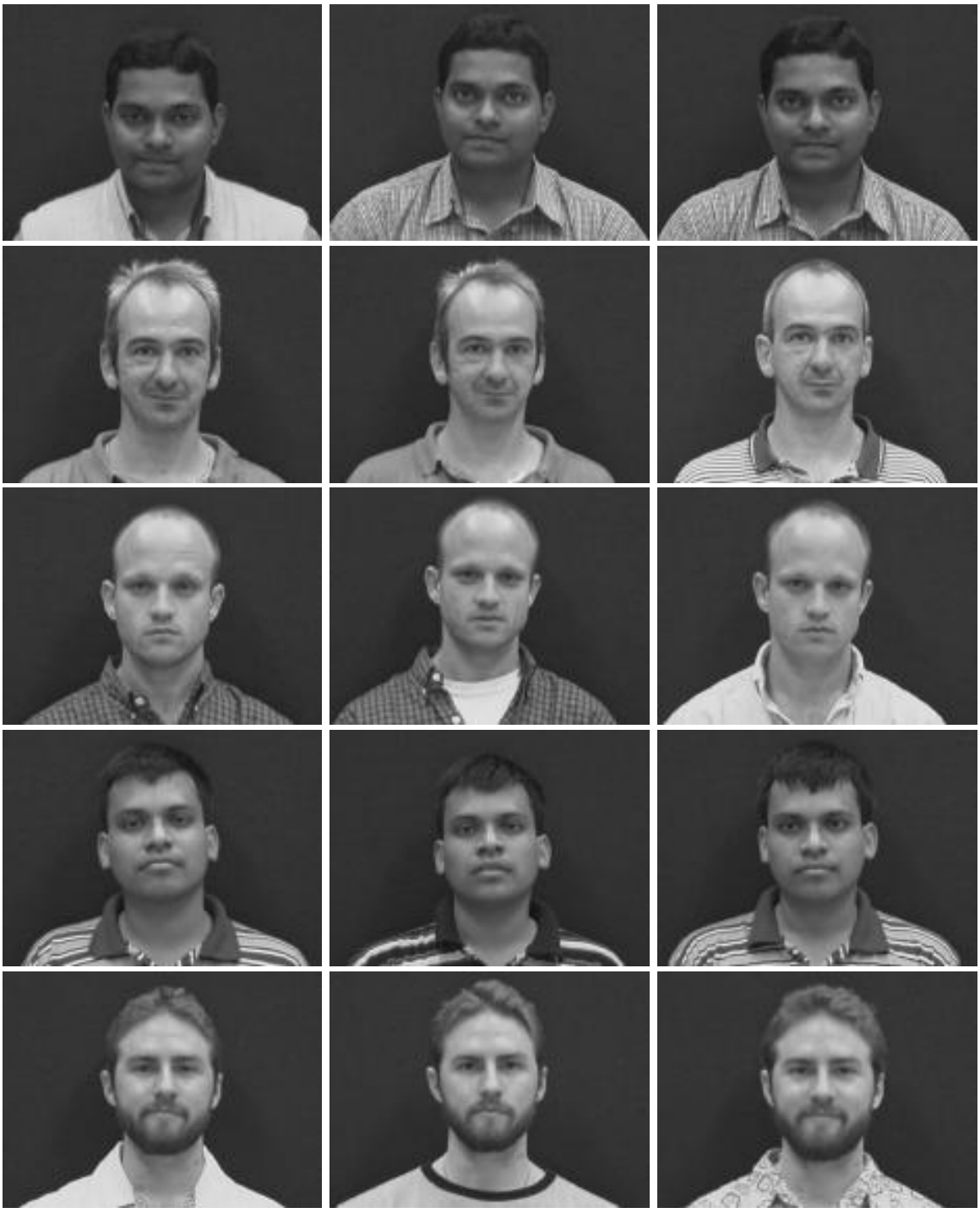

Figure 4.7: Subjects in the VidTIMIT database (Part G)

Figure 4.8: Subjects in the VidTIMIT database (Part H)

Figure 4.9: Subjects in the VidTIMIT database (Part I)

Chapter 5

Face Based Verification

5.1 Abstract

In this chapter we first review important publications in the field of face recognition (Section 5.2). Geometric features, templates, Principal Component Analysis (PCA), pseudo-2D Hidden Markov Models (HMM), Elastic Graph Matching (EGM), as well as other points are covered. Important issues, such as the effects of an illumination direction change and the use of different face areas, are also covered.

In Section 5.3 a new feature set (termed *DCT-mod2*) is proposed; the feature set utilizes 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 database suggest that the proposed feature set is the most robust, followed by (in order of robustness and performance): 2D Gabor wavelets, 2D DCT coefficients and PCA (eigenface) derived features. Moreover, compared to Gabor wavelets, the *DCT-mod2* feature set is over 80 times quicker to compute.

In Section 5.4 the effects of likelihood normalization in face verification are studied. Current face verification systems use a fixed threshold (or decision surface) to make the final accept or reject decision; this approach does not take into account a mismatch between training and testing conditions, where use of corrupted face images can lead to a false rejection of the claimant. To account for varying image conditions, the decision threshold can be automatically tuned through the use of likelihood normalization. The effectiveness of three likelihood normalization approaches, the Background Model Set (BMS), the Universal

Background Model (UBM) and an alternate version of UBM, denoted as UBM-alt[1], is evaluated. Experiments using face images corrupted by an illumination change, compression artefacts and white Gaussian noise, show that likelihood normalization has little effect when using PCA derived features, while all three normalization approaches provide significant performance improvements when using 2D DCT, 2D Gabor wavelet or *DCT-mod2* features. Out of the three, the UBM-alt approach is the most useful, as it provides performance which is close to the best approach (BMS) while having the advantage of being client-independent. The results also show that while PCA derived features are greatly affected by an illumination direction change, they are quite immune to compression artefacts and white Gaussian noise.

In Section 5.5 we propose to solve the fragility of PCA derived features to the illumination direction change 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 then performed. We show that the *enhanced PCA* technique retains all the positive aspects of traditional PCA, while also being robust to changes in the illumination direction.

In Section 5.6, the *DCT-mod2* approach is extended by increasing the number of blocks used in deriving each feature vector; moreover, windowing is introduced, allowing the variation of the contribution of each block. Results show that depending on the window used, the modified feature set is less robust compared to the original feature set when using face images corrupted with an illumination direction change; however, the modified set is more robust to compression artefacts and white Gaussian noise.

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

Publications resulting from this research: [131, 132, 133, 134, 136, 137, 138].

[1]In UBM-alt normalization, both the client and the impostor models are generated using the EM algorithm, which is in contrast to UBM normalization where the client models are generated by adapting the impostor model.

5.2 Summary of Past Face Recognition Approaches

This section presents a concise review of previous approaches to automatic face recognition. It goes into detail with the most important and/or popular approaches; the reader is also directed to recent survey articles [26, 46].

Generally speaking, a full face recognition system can be thought of as being comprised of three stages:

1. Face localization and segmentation
2. Normalization
3. The actual face identification/verification, which can be further subdivided into:
   (a) Feature extraction
   (b) Classification

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 last stage (3). Some recent approaches to face location and segmentation are presented in [50, 108, 122, 162]. The second stage (normalization) usually involves an affine transformation [41] (to correct for size and rotation), but it can also involve an illumination normalization (however, illumination normalization may not be necessary if the feature extraction method is robust against varying illumination).

There are many approaches to face recognition - ranging from the Principal Component Analysis (PCA) approach (also known as eigenfaces) [86, 150], Elastic Graph Matching (EGM) [34, 74], Artificial Neural Networks [73, 151], to pseudo-2D Hidden Markov Models (HMM) [36, 123]. All these systems differ in terms of the feature extraction procedure and/or the classification technique used. These systems, and many others, are described in the sections below.

It must be noted that while the verification task has the greatest application potential [33], past research on face recognition systems has concentrated on the identification aspect; moreover, while identification and verification systems share feature extraction techniques

and in many cases a large part of the classifier structure, there is no *a priori* guarantee that an approach used in the identification scenario would work equally well in the verification scenario.

5.2.1 Geometric Features vs Templates

Brunelli and Poggio [19] compared the performance of a system utilizing automatically extracted geometric features and a classifier based on the squared Mahalanobis distance [35] (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 center 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
- 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 face area (from eyebrows downward), were used by a classifier based on normalized cross correlation with a set of template images. In both systems, the size of the face image was first normalized. 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.

5.2.2 Principal Component Analysis (eigenfaces) and Related Techniques

Inspired by the work of Kirby and Sirovich [66], Turk and Pentland [150] proposed the use of Principal Component Analysis (PCA) [88] as a holistic feature extraction method for use in face recognition.

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

$$\vec{x} = U^T(\vec{f} - \vec{f}_\mu) \tag{5.1}$$

where matrix U contains D eigenvectors (with largest corresponding eigenvalues) of the training data covariance matrix, and $\vec{f}_\mu$ is the mean of training face vectors. The eigenvectors are referred to as "eigenfaces" (see Section 5.3.1.1 for full derivation).

As $\vec{x}$ is in effect a dimensionality reduced version of $\vec{f}$, the above PCA based feature extraction technique is sensitive to translation, rotation, scaling as well as changes in illumination. Thus prior to feature extraction, the face image must be normalized (e.g., the location of the eyes must be the same for each person and any illumination changes must be compensated).

On a database of 16 people and using a 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.

Moghaddam and Pentland [86] 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 [19]. 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 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.* [16] 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 databases (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 significantly less than PCA derived features.

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

Samaria [123] extended 1D HMMs (popular in speech recognition [56, 106]) 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 [see Eqn.(2.23)] 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 maximized). 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 translation, indicating that the face normalization stage need not be as accurate as for the PCA based approach.

Samaria showed that on a 40 person database 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 analyzed 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 leveled off when

40 eigenfaces were used.

In related work, Nefian and Hayes [95] proposed to use 2D Discrete Cosine Transform (2D DCT) coefficients [41] 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.* [36] extended the pseudo-2D HMM approach to use 2D DCT coefficients directly from JPEG compressed images [158, 159]; moreover, they have also shown that utilizing 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).

5.2.4 Elastic Graph Matching (EGM) Based Techniques

Lades *et al.* [74] 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 [75], differing in orientation and scale (see Section 5.3.1.2 for more information).

Comparing two faces is 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 change) and to a lesser extent, changes in the view point. The quality of a match is evaluated using a distance function:

$$d(T,R) = \sum_{i=1}^{N_N} d_f(T_i, R_i) + \xi \sum_{i=1}^{N_N} d_s(T_i, R_i) \tag{5.2}$$

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 penalizing 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 minimized via translation of the test grid and perturbation of the locations of its nodes. Lades *et al.* proposed an approximate solution to the minimization 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 $\xi \rightarrow \infty$ in Eqn. (5.2)]. 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.* [34] 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 database 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 minimization of $d(T,R)$ is omitted (i.e., the test grid is kept rigid).

Kotropoulos *et al.* [71] used the outputs of multiscale morphological dilation and erosion operations [41] to yield a feature vector for each node. Compared to feature vectors based on 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 [145] show that the morphological operation based approach has slightly lower error rates than the standard approach based on Gabor wavelets.

5.2.5 Other Approaches

Matas *et al.* [83] 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 transformation includes translation, rotation 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 utilized 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 database which had mainly expression changes show a minor improvement over Duc's extended EGM approach (described in Section 5.2.4).

Lawrence *et al.* [73] proposed the use of a hybrid neural-network approach to face recognition. The system combined local image sampling, a self-organizing map (SOM) [70] and a convolutional neural network. On a database 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 5.2.2) 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).

5.2.6 Important Issues

Zhang *et al.* [164] compared the performance of the EGM approach with a system comprised of the PCA based feature extractor and a nearest neighbour classifier. Results on a combined database 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.* [72] 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 normalization technique, which, on a small database containing face images collected in real life conditions, was shown to significantly improve the performance of a EGM based system which utilized 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 [75], edge maps, first & second derivatives and log transformations [41]. Several classifiers, based on pixel differences between two processed images, were also evaluated; all of the classifiers produced similar identification results. On a database 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 emphasized 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.* [16] 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.* [27] quantitatively proved 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 [86] (see Section 5.2.2), 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 needs to be disregarded as it is mostly affected by expression changes and beards.

5.3 Feature Extraction for Face Verification

From the review in Section 5.2 it is evident that PCA derived features, and to a lesser extent, 2D Gabor wavelet derived features, are affected by an illumination direction change. As will be shown, 2D DCT based features are also sensitive to changes in the illumination direction. In this section we introduce four new feature sets, which are significantly less affected by an illumination direction change: *DCT-delta*, *DCT-mod*, *DCT-mod-delta* and *DCT-mod2*. We will show that the *DCT-mod2* method, which utilizes 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.

The rest of this section is organized as follows. In Section 5.3.1, we review the PCA, 2D Gabor wavelet and 2D DCT feature extraction methods, and describe the proposed feature extraction methods. The performance of the described feature extraction techniques is compared in Section 5.3.2. The results are discussed and conclusions drawn in Section 5.3.3.

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

5.3.1 Feature Extraction Techniques

5.3.1.1 Eigenfaces (PCA)

Given a face image matrix[2] F of size $Y \times X$, we construct a vector representation by concatenating all the columns of F to form a column vector $\vec{f}$ of dimensionality YX. Given a set of training vectors $\{\vec{f}_i\}_{i=1}^{N_P}$ for all persons, we define the mean of the training set as $\vec{f}_\mu$. A new set of mean subtracted vectors is formed using:

$$\vec{g}_i = \vec{f}_i - \vec{f}_\mu, \quad i = 1, 2, ..., N_P \tag{5.3}$$

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

$$C = GG^T \tag{5.4}$$

Due to the size of C, calculation of the eigenvectors of 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 [150] exploit this fact to determine the eigenvectors using an alternative method, summarized as follows. Let us denote the eigenvectors of matrix G^TG as $\vec{v}_j$ with corresponding eigenvalues λ_j:

$$G^TG\vec{v}_j = \lambda_j\vec{v}_j \tag{5.5}$$

Pre-multiplying both sides by G gives us:

$$GG^TG\vec{v}_j = \lambda_jG\vec{v}_j \tag{5.6}$$

Letting $\vec{u}_j = G\vec{v}_j$ and substituting for C from Eqn. (5.4):

$$C\vec{u}_j = \lambda_j\vec{u}_j \tag{5.7}$$

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

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

$$\vec{x} = U^T(\vec{f} - \vec{f}_\mu) \tag{5.8}$$

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

5.3.1.2 2D Gabor Wavelets

The biologically inspired family of 2D Gabor wavelets is defined as follows [75]:

$$\Psi(y, x, \omega, \theta) = \frac{\omega}{\kappa\sqrt{2\pi}}\psi_A(y, x, \omega, \theta)\left[\psi_B(y, x, \omega, \theta) - \exp\left\{-\frac{\kappa^2}{2}\right\}\right] \tag{5.9}$$

where

$$\psi_A(y, x, \omega, \theta) = \exp\left\{-\frac{\omega^2}{8\kappa^2}\left[4(y\sin\theta + x\cos\theta)^2 + (y\cos\theta - x\sin\theta)^2\right]\right\} \tag{5.10}$$

and

$$\psi_B(y, x, \omega, \theta) = \exp\{i(\omega y\sin\theta + \omega x\cos\theta)\} \tag{5.11}$$

Here ω is the radial frequency in radians per unit length and θ is the wavelet orientation in radians. Each wavelet is centered 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 [75].

Feature extraction is done as follows. A coarse rectangular grid is placed over given face image F. At each node of the grid, the inner product of F with each member of the family is computed:

$$P_{j,k} = \int_y\int_x \Psi(y_0 - y, x_0 - x, \omega_j, \theta_k)F(y, x)\ dx\ dy \tag{5.12}$$

for $j = 1, 2, ..., N_\omega$ and $k = 1, 2, ..., N_\theta$. Here, the node is located at (y_0, x_0). An $N_\omega N_\theta$-dimensional feature vector[3] for location (y_0, x_0), is then constructed using the modulus of each inner product [74]:

$$\vec{x} = \left[\begin{array}{cccccccccccc} |P_{1,1}| & |P_{1,2}| & \cdots & |P_{1,N_\omega}| & \cdots & |P_{2,1}| & |P_{2,2}| & \cdots & |P_{2,N_\omega}| & \cdots & |P_{N_\theta,N_\omega}| \end{array} \right]^T \quad (5.13)$$

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

5.3.1.3 2D Discrete Cosine Transform

Here the given face image is analyzed on a block by block basis. 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 5.1). The result is an $N \times N$ matrix $C(v, u)$ containing 2D DCT coefficients:

$$C(v, u) = \alpha(v)\alpha(u) \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} f(y, x)\beta(y, x, v, u) \quad \text{for } v, u = 0, 1, 2, ..., N-1 \quad (5.14)$$

where

$$\alpha(v) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } v = 0 \\ \sqrt{\frac{2}{N}} & \text{for } v = 1, 2, ..., N-1 \end{cases} \quad (5.15)$$

and

$$\beta(y, x, v, u) = \cos\left[\frac{(2y+1)v\pi}{2N}\right] \cos\left[\frac{(2x+1)u\pi}{2N}\right] \quad (5.16)$$

The coefficients are ordered according to a zig-zag pattern, reflecting the amount of information stored [41] (see Figure 5.2). For a block located at (b, a), the baseline 2D DCT feature vector is composed of:

$$\vec{x} = \left[\begin{array}{cccc} c_0^{(b,a)} & c_1^{(b,a)} & ... & c_{M-1}^{(b,a)} \end{array} \right]^T \quad (5.17)$$

where $c_n^{(b,a)}$ denotes the n-th 2D DCT coefficient and M is the number of retained coefficients[4]. To ensure adequate representation of the image, each block overlaps its

[3] Typically, $N_\omega = 3$ and $N_\theta = 6$, resulting in an 18 dimensional vector.
[4] In our experiments, $M = 15$.

horizontally and vertically neighbouring blocks by 50% [36]. Thus 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[5].

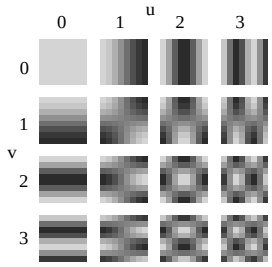

Figure 5.1: Several 2D DCT basis functions for N=8. Lighter colours represent larger values.

u

v	**0**	**1**	**2**	**3**
0	0	1	5	6
1	2	4	7	12
2	3	8	11	13
3	9	10	14	15

Figure 5.2: Ordering of 2D DCT coefficients $C(v, u)$ for N=4.

5.3.1.4 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 [125] (see also Section 3.3.3).

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

$$\Delta^h c_n^{(b,a)} = \frac{\sum_{k=-K}^{K} k h_k \; c_n^{(b,a+k)}}{\sum_{k=-K}^{K} h_k k^2} \tag{5.18}$$

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

$$\Delta^v c_n^{(b,a)} = \frac{\sum_{k=-K}^{K} k h_k \; c_n^{(b+k,a)}}{\sum_{k=-K}^{K} h_k k^2} \tag{5.19}$$

where h is a $2K + 1$ dimensional symmetric window vector. In this section we shall use $K = 1$ and a rectangular window.

Let us assume that we have three horizontally consecutive blocks X, Y and Z. Each block is composed of two components: facial 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

[5]Thus for a 56×64 image, there are 195 2D DCT feature vectors.

reasonable assumption if the blocks are small and close or overlapping). To find the deltas for block Y, we apply Eqn. (5.18) to obtain (ignoring the denominator):

$$\begin{aligned} \Delta^h Y &= -X + Z && (5.20) \\ &= -(I_X + I_N) + (I_Z + I_N) && (5.21) \\ &= I_Z - I_X && (5.22) \end{aligned}$$

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*. We interpret these delta coefficients as transitional spatial information (somewhat akin to edges).

DCT-delta feature extraction for a given block is only possible when the block has vertical and horizontal neighbours. Thus processing an image which has Y rows and X columns and using a 50% block overlap results in $N_{D2} = (2\frac{Y}{N} - 3) \times (2\frac{X}{N} - 3)$ *DCT-delta* feature vectors[6].

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

By inspecting Eqns. (5.14) and (5.16), it is evident that the 0-th 2D DCT coefficient will reflect the average pixel value (or the DC level) inside each block and hence will be the most affected by any illumination change. Moreover, by inspecting Figure 5.1 it is evident that the first and second coefficients represent the average horizontal and vertical pixel intensity change, respectively. As such, they will also be significantly affected by any 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*.

[6]Thus for a 56×64 image, there are 143 *DCT-delta* feature vectors.

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:

$$\vec{x} = \left[\begin{array}{cccccccccc} \Delta^h c_0 & \Delta^v c_0 & \Delta^h c_1 & \Delta^v c_1 & \Delta^h c_2 & \Delta^v c_2 & c_3 & c_4 & ... & c_{M-1} \end{array} \right]^T \tag{5.23}$$

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

Thus each *DCT-mod-delta* and *DCT-mod2* feature vector represents transitional spatial information as well as local texture information.

As for *DCT-delta*, *DCT-mod-delta* and *DCT-mod2* feature extraction for a given block is only possible when the block has vertical and horizontal neighbours. Thus processing an image which has Y rows and X columns and using a 50% block overlap results in $N_{D2} = (2\frac{Y}{N} - 3) \times (2\frac{X}{N} - 3)$ *DCT-mod-delta* or *DCT-mod2* feature vectors[7].

5.3.2 Experiments

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

To find the face, we use template matching with several prototype faces[8] of varying dimensions. Using the distance between the eyes as a size measure, an affine transformation is used [41] 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 window, $w(y, x)$, containing the eyes and the nose (the most invariant face area to changes in the expression and hair style) is extracted from the image.

[7] Thus for a 56×64 image, there are 143 *DCT-mod-delta* or *DCT-mod2* feature vectors.

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

For PCA, the dimensionality of the face window is reduced to 40 (choice based on the works by Kirby and Sirovich [66], Samaria [123] and Belhumeur *et al.* [16]).

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.* [34] 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 centers was chosen to be as close as possible to the centers of the blocks used in *DCT-mod2* feature extraction.

In our experiments, we use a sequence of images (video) from the VidTIMIT database (see Chapter 4) for person verification. If the sequence has N_I images, then $N_V = N_I$ for PCA derived features, $N_V = N_I N_G$ for Gabor features, $N_V = N_I N_D$ for 2D DCT and *DCT-mod* features and $N_V = N_I N_{D2}$ for *DCT-delta*, *DCT-mod-delta* and *DCT-mod2* features. To reduce the computational burden during modeling and testing, every second video frame was used. For each feature extraction method, 8-Gaussian client models (GMMs) were generated from features extracted from face windows 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[9]. Taking this into account, an illumination direction change was introduced to face windows 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 window $v(y, x)$ is created by transforming $w(y, x)$ using:

$$\begin{aligned} v(y,x) &= w(y,x) + mx + \delta \qquad (5.24) \\ \text{for:} \quad y &= 0, 1, \cdots, N_Y - 1 \\ x &= 0, 1, \cdots, N_X - 1 \\ \text{where:} \quad m &= \frac{-\delta}{(N_X - 1)/2} \\ \delta &= \text{illumination delta (in pixels)} \end{aligned}$$

[9] As evidenced by the images presented in [71], which were obtained under real-life conditions.

Example face windows for various δ are shown in Figure 5.3. It must be noted that this model of illumination direction change is artificial and restrictive as it does not cover all the effects possible in real life (shadows[10], etc.), but it is useful for providing suggestive results[11].

Figure 5.3: Examples of varying light illumination; left: $\delta = 0$ (no change); middle: $\delta = 40$; right: $\delta = 80$

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 used for simulating impostor accesses against the remaining 35 persons. As in [112], 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 EER.

In the first experiment, we found the performance of the 2D DCT approach on face windows with $\delta = 0$ (i.e., no illumination change) while varying the dimensionality of the feature vectors. The results are presented in Figure 5.4. 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 significantly 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 5.5.

[10]However, the face images presented in [16] show that only extreme illumination direction conditions produce significant shadows, where even humans have trouble recognizing faces.

[11]See also Appendix A for experiments on the Weizmann Database [3].

Method	Time (msec)
PCA	11
DCT	6
Gabor	675
DCT-mod2	8

Table 5.1: Average time taken per face window (results obtained using Pentium III 500 MHz, Linux 2.2.18, gcc 2.96)

In the third experiment we compared the performance of PCA, PCA with histogram equalization pre-processing[12], DCT, Gabor and *DCT-mod2* features for varying δ. Results are presented in Figure 5.6.

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 in an exponential manner as the overlap is increased. Results are shown in Figure 5.7.

Computational burden is an important factor in practical applications, where the amount of required memory and speed of the processor have direct bearing on the final cost. Hence in the final experiment we compared the average time taken to process one face window by PCA, DCT, Gabor and *DCT-mod2* feature extraction techniques. It must be noted that apart from having the transformation data pre-calculated (e.g., β 2D DCT basis functions), no thorough hand optimization of the code was done. Nevertheless, we feel that this experiment provides figures which are at least indicative. Results are listed in Table 5.1.

5.3.3 Discussion and Conclusions

As can be observed in Figure 5.4, the first three 2D DCT coefficients contain a significant amount of person dependent information; thus ignoring them (as in *DCT-mod*) implies a reduction in performance. This is verified in Figure 5.5 where the *DCT-mod* features have

[12]Histogram equalization [24, 41] is often used in an attempt to reduce the effects of varying illumination conditions [69, 89].

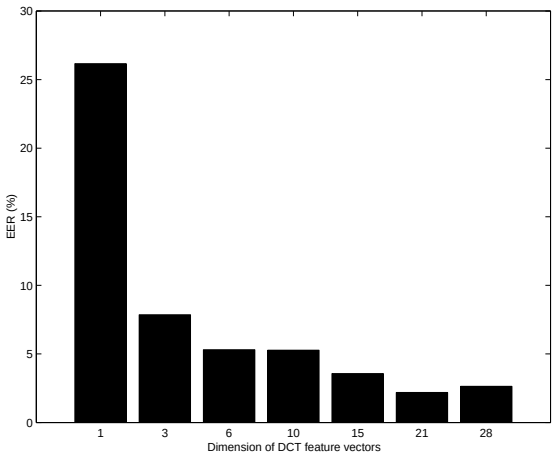

Figure 5.4: Performance for varying dimensionality of 2D DCT feature vectors

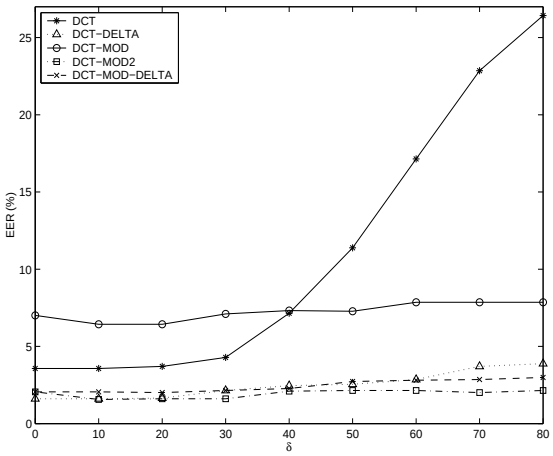

Figure 5.5: Performance of 2D DCT and proposed feature sets

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 $\delta \geq 40$ (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; they 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 (Fig. 5.7) results in a

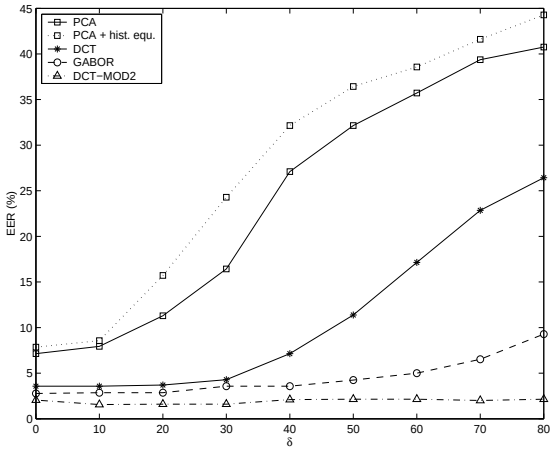

Figure 5.6: Performance of PCA, PCA with histogram equalization pre-processing, DCT, Gabor and *DCT-mod2* feature sets

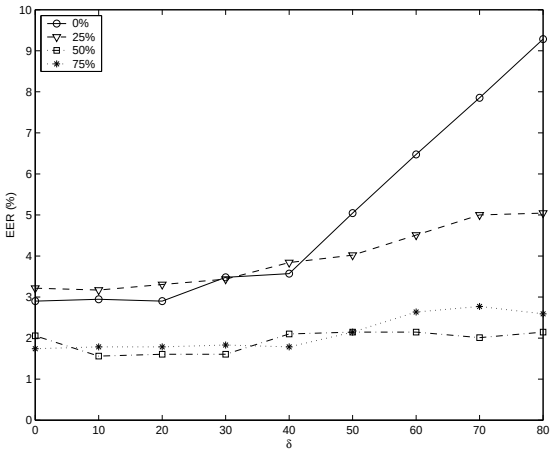

Figure 5.7: Performance of *DCT-mod2* feature set for varying overlap

performance degradation as δ is increased, implying that the assumption that the blocks are corrupted with the same noise has been violated (see Section 5.3.1.4). Increasing the overlap from 50% to 75% had little effect on the performance at the expense of extracting significantly more feature vectors.

By comparing the performance of PCA, PCA with histogram equalization pre-processing, 2D DCT, 2D Gabor and *DCT-mod2* feature sets (Figure 5.6), 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 equalization as pre-processing for PCA increases the error rate in all cases, and most notably offers no help against illumination changes. The results thus suggest that we can order the feature sets, based on their robustness and performance, as follows: *DCT-mod2*, 2D Gabor, 2D DCT, PCA, and lastly, PCA with histogram equalization pre-processing.

From Table 5.1 we can see that 2D Gabor features are the most computationally expensive to calculate, taking about 84 times longer than *DCT-mod2* features. This is due to the size of the 2D Gabor wavelets as well as the need to compute both real and imaginary inner products. Compared to 2D Gabor features, PCA, 2D DCT and *DCT-mod2* features

take a relatively similar amount of time to process one face window.

It must be noted that when using the GMM classifier in conjunction with the 2D Gabor, 2D DCT or *DCT-mod2* features, the spatial relation between major face features (e.g., eyes and nose) is lost. However, excellent performance is still obtained[13], implying that the use of more complex classifiers which preserve spatial relation, such as a pseudo-2D HMM and elastic graph matching, is not necessary. Moreover, due to the loss of the spatial relations, the GMM classifier theoretically has some inbuilt robustness to translation (which may be caused by inaccurate face localization).

It must also be noted that using the introduced illumination change, the center 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.

5.4 Effects of Likelihood Normalization in Face Verification

It seems all current face-based authentication systems (e.g., [34, 62, 102, 142, 146]) effectively follow a thresholding approach[14] to make the final accept or reject decision. The result of comparison of the claimant's features (X) with a model belonging to the person whose identity is being claimed (λ_K) is a matching score or a likelihood. Let us refer to this result as $p(X|\lambda_K)$. Given a threshold t, the claim is accepted when:

$$p(X|\lambda_K) \geq t \tag{5.25}$$

and rejected otherwise.

[13] See also Appendix B.

[14] It must be noted that in [62, 142], the Support Vector Machine (SVM) classifier provides a fixed decision surface in feature space. The decision surface is analogous to a fixed threshold in 1D case, as described by Eqn. (5.25).

The first problem with Eqn. (5.25) is that the threshold is often person-dependent. Finding the threshold involves finding the distribution of true claimant and impostor likelihoods. Because of database size limitations, there is usually a relatively low number of true claimant likelihoods for any single person - thus any resulting performance measure has little statistical significance [33]. One solution is to use a global threshold (i.e., person-independent), which is found using pooled true claimant likelihoods from all persons. However, since the system is now not tuned for each person, the use of a global threshold may result in worse performance.

The second and more important problem with Eqn. (5.25) is that if there is a mismatch between training and testing conditions, the claim may be automatically rejected due to a low likelihood. The mismatch can occur due to an illumination direction change, compression artefacts or white Gaussian noise. While the illumination direction change may be of most concern in security systems, in forensic applications [77] 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, possibly damaged and/or low quality analogue video tape or TV signal corrupted with "static" noise.

In speech-based verification systems it has been found that use of normalized likelihoods together with a global threshold improves performance as well as robustness [117, 118]. By reformulating Eqn. (5.25) in the Bayesian framework (see Chapter 2), the claim is accepted when:

$$\frac{p(X|\lambda_K)}{p(X|\lambda_{\overline{K}})} \geq t \tag{5.26}$$

or

$$p(X|\lambda_K) \geq t\, p(X|\lambda_{\overline{K}}) \tag{5.27}$$

where $p(X|\lambda_{\overline{K}})$ is the result of the claimant's features being compared to an anti-client model ($\lambda_{\overline{K}}$), i.e., the likelihood of the claimant being an impostor. If the testing condition causes $p(X|\lambda_K)$ to decrease, then it is reasonable to suppose that $p(X|\lambda_{\overline{K}})$ will also decrease - thus the ratio of the likelihoods may remain relatively unaffected. In effect, the global threshold is automatically tuned for each person to account for environmental conditions.

As described in Chapter 2, when utilizing a Gaussian Mixture Model (GMM) classifier, there are two popular approaches for finding the impostor likelihood:

1. Background Model Set (BMS) approach [112].

2. Universal Background Model (UBM) approach [115].

The most important difference between the two techniques is that in the latter approach the impostor likelihood is client independent.

In this section we evaluate the effectiveness of the above normalization approaches in a GMM based face verification system, using three different feature sets which are commonly used in recognition systems (PCA derived [150], 2D DCT derived [41], 2D Gabor wavelet derived [74, 75]) and the recently proposed *DCT-mod2* feature set (see Section 5.3.1.5), in four conditions: clean images and images corrupted with an illumination direction change, compression artefacts and white Gaussian noise.

5.4.1 Experiment Setup

The experiment setup is similar to that of Section 5.3.2. The changes are as follows. For experiments involving compression artefacts, face windows extracted from Sessions 2 and 3 were processed by a JPEG codec [158, 159] (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 (PSNR); the average PSNR of the corrupted images is 31.13 dB. Similarly, for TV "static" noise experiments, face windows extracted from Sessions 2 and 3 were corrupted by additive white Gaussian noise, resulting in the PSNR being equal to 26 dB. Example face windows are shown in Figure 5.8.

5.4.2 Experiments and Discussion

In the first experiment, EER performance for PCA features is found using clean and corrupted images, with four classifier configurations: no normalization ($\mathcal{L}(X|\lambda_{\overline{K}}) = 0$), BMS based normalization, UBM based normalization and finally *UBM-alt* normalization, where

Figure 5.8: From left to right: original image, corrupted with illumination change ($\delta = 80$), corrupted with compression artefacts (PSNR=31.7 dB), corrupted with white Gaussian noise (PSNR=26 dB)

the client models are constructed using the EM algorithm (instead of adaptation via MAP) and the impostor likelihood is found using Eqn. (2.39). The *UBM-alt* normalization is used as a reference to deduce whether any performance gains by the UBM normalization are due to MAP training of the models or the process of normalization. When deriving client models from λ_{UBM} (via MAP), only the weights and means were adapted - preliminary experiments showed that adapting the covariance matrices resulted in poorer PCA performance. Results are shown in Figure 5.9.

The second, third and fourth experiments are a repeat of the first, except that 2D DCT, 2D Gabor wavelet and *DCT-mod2* features are used, respectively. Results are presented in Figures 5.10 through 5.12.

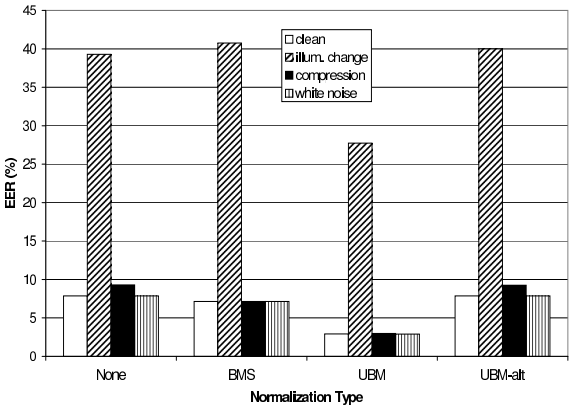

Figure 5.9: Performance using PCA derived features

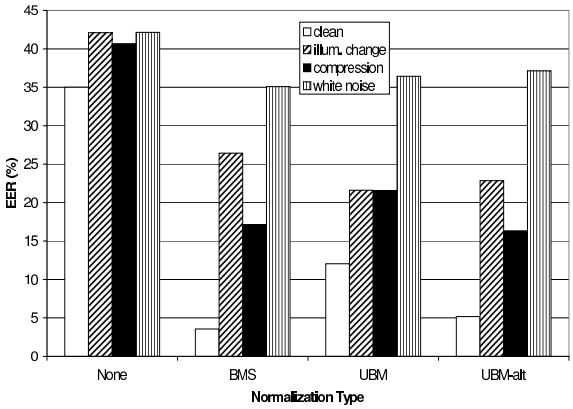

Figure 5.10: Performance using 2D DCT features

When using PCA derived features, BMS based normalization causes minor improvements in performance. For the case of images corrupted using the illumination direction change, the performance is slightly worse. This is in contrast to UBM

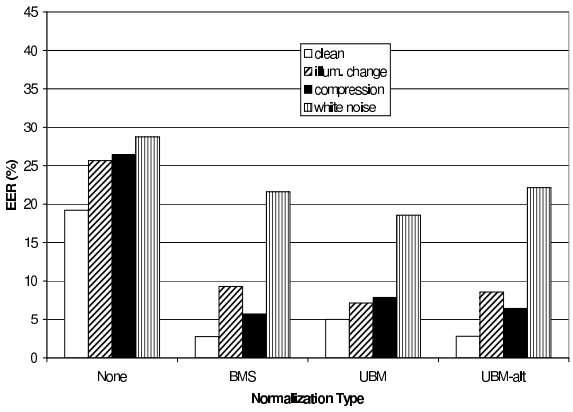

Figure 5.11: Performance using 2D Gabor features

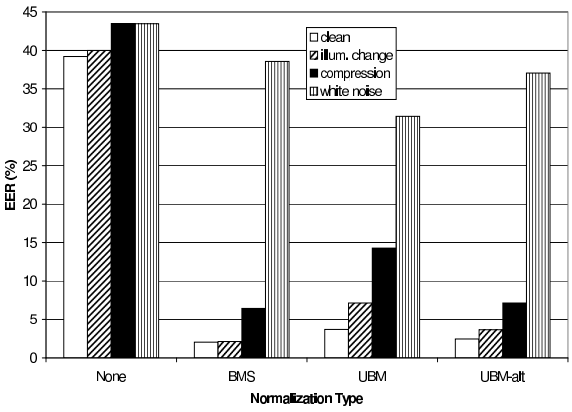

Figure 5.12: Performance using DCT-mod2 features

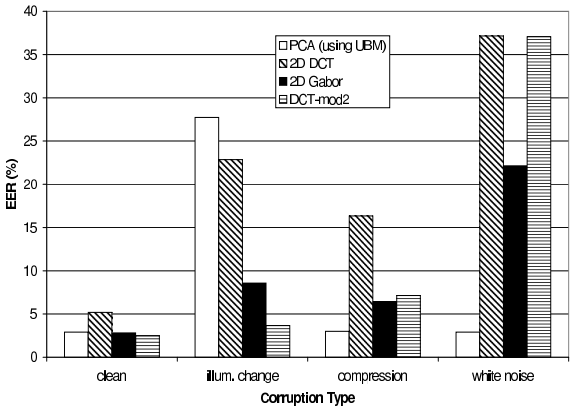

Figure 5.13: Performance of all features. UBM-alt normalization is used for DCT, Gabor and DCT-mod2 features, while UBM is used for PCA derived features

normalization, where it appears that there are significant performance gains in all four image conditions (e.g., when using illumination corrupted images, the EER is reduced from 39.29% to 27.73%). However, by comparing with *UBM-alt*, this improvement can be attributed to MAP training of the client models rather than to likelihood normalization.

As described in Sections 2.3.2.2, data from all clients is used to find λ_{UBM}. In the UBM approach, client models are created by adapting λ_{UBM} (via MAP) using client specific data. This is in contrast to directly computing the client models using the EM algorithm, where only client specific data is used. Effectively there is approximately 30 times more data used during MAP based training than in direct EM based training. Thus the relatively poor performance of the BMS and *UBM-alt* approaches when using PCA derived features can be

attributed to not enough data being available for training with the EM algorithm.

The rest of the discussion concerns 2D DCT, 2D Gabor or *DCT-mod2* features. Here, use of likelihood normalization is important in order to obtain good performance when using a global threshold. The performance gains are quite staggering - e.g., for *DCT-mod2* features, the EER drops from 39.2% to 2.05% when using clean images and the BMS normalization approach. In all image conditions use of likelihood normalization provides better performance than without normalization. Generally, the BMS approach obtains the best performance, closely followed by *UBM-alt* and lastly by UBM.

Since the BMS and *UBM-alt* approaches obtain better performance than UBM on clean images, there is an implication that the client models derived from λ_{UBM} are not precise enough (thus MAP adaptation is not sufficiently "tuning" the λ_{UBM} for each client). However, this imprecision allows some leeway in the drift of features, as evidenced by the case of 2D Gabor and *DCT-mod2* features obtained from face images corrupted by white Gaussian noise: here the UBM approach provides the best results.

Since there is a large number of feature vectors extracted, there is no data shortage problem as experienced with PCA features when using EM algorithm based training (e.g., for *DCT-mod2*, there is 143 feature vectors extracted from each video frame, resulting in an average of 45474 training vectors per person). Thus when testing with images corrupted other than with white Gaussian noise, training the client models using the EM algorithm for training is better than using MAP adaptation.

It must be noted that the UBM-alt approach obtains results which are close to the BMS approach, with the advantage of being client-independent. Clearly, using client-independent normalization (as opposed to the BMS approach) greatly simplifies the implementation of the verification system.

These experiments also allow us to compare the relative robustness of all the features. Results are shown in Figure 5.13, where in every case except for PCA, the *UBM-alt* normalization approach is used. For PCA we use the UBM approach, as it provides the most reliable client models. It can be seen that PCA derived features are the most affected by the illumination direction change, while being the least affected by compression artefacts

and white Gaussian noise. This is in contrast to 2D DCT and *DCT-mod2* features, which are the most affected by white Gaussian noise. 2D Gabor wavelets provide intermediary performance between 2D DCT and PCA features. *DCT-mod2* features are the least affected by the illumination direction change, followed by 2D Gabor wavelets and 2D DCT features. After PCA, 2D Gabor wavelets are the least affected by compression artefacts, followed by *DCT-mod2* and distantly by 2D DCT. As can be seen, there is no feature type which is immune to all corruption types. However, it can be argued that the best overall performance is obtained by 2D Gabor wavelets.

5.4.3 Conclusion

Current face verification systems achieve the final accept or reject decision using a fixed threshold (or decision surface) and thus do not take into account a mismatch between training and testing conditions, where use of corrupted face images can lead to a false rejection of the claimant. We have evaluated the effectiveness of several likelihood normalization approaches (suited to the GMM classifier) which automatically tune the threshold to account for the condition of test images. Results on the VidTIMIT database, using test images corrupted by an illumination direction change, compression artefacts and white Gaussian noise, suggest that likelihood normalization approach has little effect when using PCA derived features, while the BMS, UBM and *UBM-alt* approaches are useful when using 2D DCT, 2D Gabor wavelet or *DCT-mod2* features. Out of the three, the *UBM-alt* approach is the most useful, as it provides performance which is close to the best approach (BMS) while having the advantage of being client-independent.

5.5 Enhancement of the PCA Approach via DCT-mod2

As shown in Sections 5.3.2 and 5.4.2, PCA derived features are sensitive to changes in the illumination direction. However, in Section 5.4.2 it was shown that the PCA approach is quite robust to compression artefacts and white Gaussian noise. We propose to solve the fragility of PCA derived features to the illumination direction change 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 then performed. We will show that this *enhanced PCA* technique retains all the positive aspects of traditional PCA, while also being robust to changes in the illumination direction. This approach differs to that of Belhumeur *et al.* [16] where training images in varying illumination conditions are required. It also differs from utilizing an edge detector as the preprocessor (as used by Moghaddam and Pentland [86], resulting in a drop in performance) since local texture information is retained.

5.5.1 Enhanced PCA

As described in Section 5.3.2, 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. (5.8)] is now different from the usual result. Because of this, use of PCA derived features results in poor performance under varying illumination conditions.

In the proposed *enhanced PCA* approach, a given face image is processed using *DCT-mod2* feature extraction to produce pseudo-image $\hat{F}$, which is then used in place of F by traditional PCA feature extraction (described in Section 5.3.1.1). Since the *DCT-mod2* feature vectors are largely robust to illumination changes, features obtained via the *enhanced PCA* should also be robust to illumination changes.

Formally, a given image is analyzed on a block by block basis, where the blocks are overlapping by 50%. Each block has N rows and N columns, where $N = 8$. Since *DCT-mod2* feature extraction for a given block is only possible when the block has vertical and horizontal neighbours, processing an image which has Y rows and X columns results in $(2\frac{Y}{N} - 3) \times (2\frac{X}{N} - 3)$ *DCT-mod2* feature vectors[15]. Let us now construct the pseudo image:

[15]Thus for a 56×64 image, there are 11×13 *DCT-mod2* feature vectors

$$\hat{F} = \begin{bmatrix} \vec{c}^{\,(\Delta b,\Delta a)} & \vec{c}^{\,(\Delta b,2\Delta a)} & \vec{c}^{\,(\Delta b,3\Delta a)} & \dots \\ \vec{c}^{\,(2\Delta b,\Delta a)} & \vec{c}^{\,(2\Delta b,2\Delta a)} & \vec{c}^{\,(2\Delta b,3\Delta a)} & \dots \\ \vec{c}^{\,(3\Delta b,\Delta a)} & \vec{c}^{\,(3\Delta b,2\Delta a)} & \vec{c}^{\,(3\Delta b,3\Delta a)} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{5.28}$$

where $\vec{c}^{\,(n\Delta b,n\Delta a)}$ 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. Because each *DCT-mod2* feature vector is $M+3$ dimensional, matrix $\hat{F}$ has $(M+3)(2\frac{Y}{N}-3)$ rows and $(2\frac{X}{N}-3)$ columns.

5.5.2 Experiments and Discussion

The experiment setup is similar to that of Section 5.4.1. The changes are as follows. For experiments involving compression artefacts, the average PSNR of the corrupted face windows extracted from Sessions 2 and 3 ranges from 45.66 to 31.13 dB. Similarly, for TV "static" noise experiments, face windows extracted from Sessions 2 and 3 were corrupted by additive white Gaussian noise, with the PSNR ranging from 40 to 15.5 dB. Taking into account the results presented in Section 5.4.2, the BMS normalization approach is utilized when using *DCT-mod2* features alone, while the UBM normalization approach is utilized for PCA and *enhanced PCA* derived features.

In the first experiment we compared the performance of the *enhanced PCA* derived features with that of the traditional PCA derived features and *DCT-mod2* features, using faces corrupted by the illumination change. Results are presented in Figure 5.14. As can be observed, *enhanced PCA* derived features are largely immune to the illumination direction change and on clean data obtain the same performance as traditional PCA based features. It must be noted that in this case the *DCT-mod2* features obtain better performance than the both types of PCA features.

The second experiment was a repeat of the first, except the faces were corrupted by the JPEG codec. Results are shown in Figure 5.15. Both traditional and *enhanced PCA* features 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. The performance then rapidly degrades as the PSNR is lowered, becoming worse than both of the PCA approaches at a PSNR equal to 33.3 dB.

The third experiment was also a repeat of the first, except that this time the faces were corrupted by additive white Gaussian noise (simulating TV "static" noise). The results are presented in Figure 5.16. 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 both of the PCA approaches at a PSNR of 36.5 dB and becomes unusable at a PSNR of 22.5 dB.

While the additive noise greatly distorts the image, the average pixel intensity remains largely the same. Thus the robustness of both types of PCA approaches stems from the dot product operation, where a given face is projected onto an eigenface. The final dot product remains largely the same for both clean and corrupted images; similar reasoning can be applied for the case of images corrupted with compression artefacts. In contrast, *DCT-mod2* features describe only a small section of the face and hence are easily affected by additive noise.

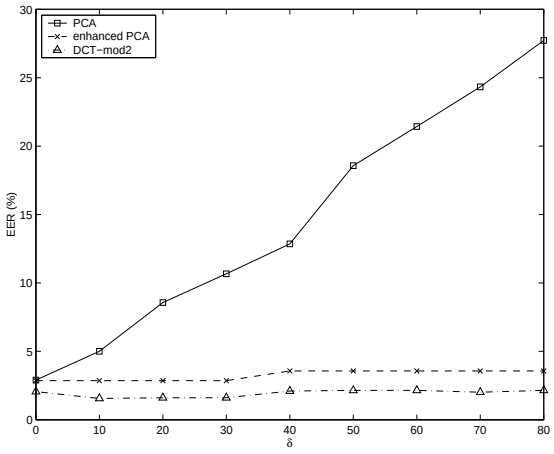

Figure 5.14: Performance for varying illumination direction

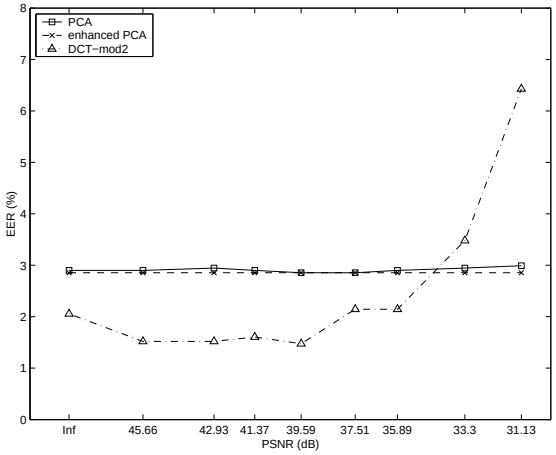

Figure 5.15: Performance for faces corrupted with compression artefacts

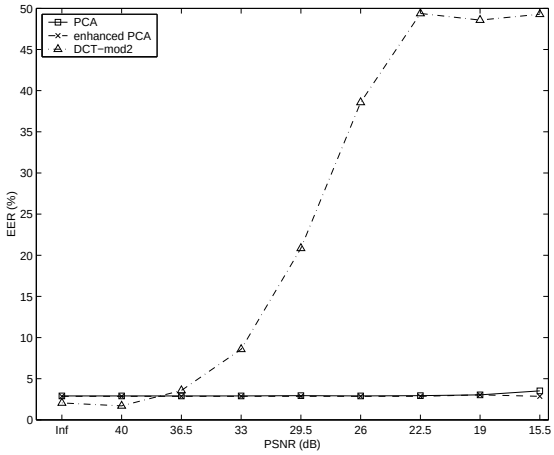

Figure 5.16: Performance for faces corrupted with white Gaussian noise

5.6 Extension of DCT-mod2 with K=2 and Various Windows

In Section 5.3.1.4 delta coefficients were calculated using $K = 1$ [see Eqns. (5.18) and (5.19)] and a rectangular window, which amounted to finding the differences between 2D DCT coefficients obtained from neighbouring blocks. In this section we shall extend the *DCT-mod2* approach with $K = 2$ (which increases the number of blocks used in deriving a *DCT-mod2* feature vector) and various windows. The performance of each configuration is then evaluated on faces corrupted by an illumination change, compression artefacts and white Gaussian noise.

5.6.1 Experiments

By inspecting Eqns. (5.18) & (5.19) and assuming that a rectangular window is used, it can be seen that for $K = 2$, 2D DCT coefficients from blocks with $k = -2$ and $k = 2$ have the largest contribution to the final value. Since this may not be optimal, we shall study two additional windows:

- Window B, where $\vec{h} = [\ 0.5\ 1.0\ 1.0\ 1.0\ 0.5\ ]^T$, causing all 2D DCT coefficients to have equal contribution

- Window C, where $\vec{h} = [\ 0.25\ 1.0\ 1.0\ 1.0\ 0.25\ ]^T$, causing the 2D DCT coefficients from the outer blocks to have smaller contribution

We shall refer to the rectangular window ($\vec{h} = [\ 1.0\ 1.0\ 1.0\ 1.0\ 1.0\ ]^T$) as Window A.

Using the experimental setup described in Section 5.5.2 the performance of *DCT-mod2* with $K = 2$ and the three windows is found using faces corrupted by an illumination change, compression artefacts and white Gaussian noise. The results are presented in Figures 5.17 through 5.19.

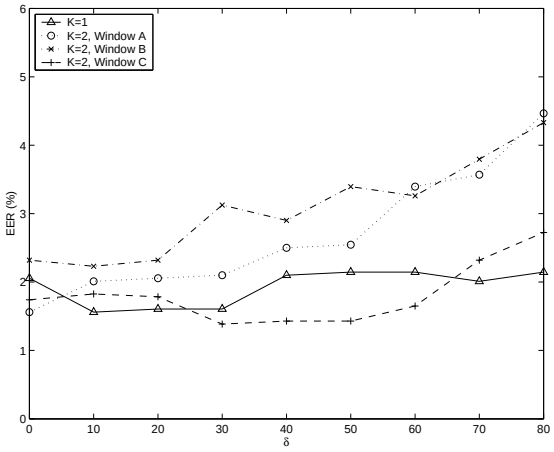

Figure 5.17: Performance for varying illumination direction

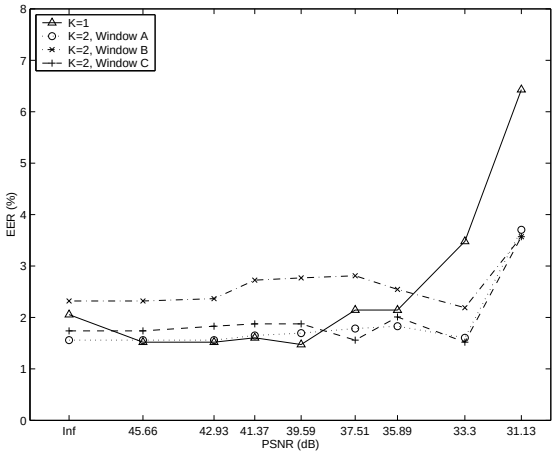

Figure 5.18: Performance for faces corrupted with compression artefacts

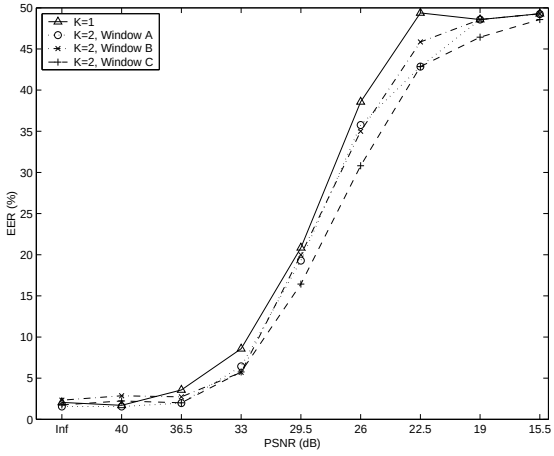

Figure 5.19: Performance for faces corrupted with white Gaussian noise

5.6.2 Discussion and Conclusions

As can be seen in Figure 5.17, where the faces have been corrupted by the illumination direction change, using $K = 2$ with Window C results in performance which is quite similar to that when using $K = 1$. When using $K = 2$ with Windows A & B, the performance degrades somewhat for large illumination direction changes. This is not unexpected, since the dominant (or equally important) blocks used in deriving a *DCT-mod2* feature vector are now significantly farther apart; thus the assumption that the blocks are corrupted with the same noise (see Section 5.3.1.4) may not hold anymore.

In Figure 5.18 it can be observed that for PSNR $\leq$ 33.3 dB, the use of $K = 2$ with any window increases the robustness of *DCT-mod2* features to compression artefacts. For larger PSNRs, the differences in performance are minor when compared to $K = 1$.

When utilizing faces corrupted with white Gaussian noise (Figure 5.19), there are minor performance differences between $K = 1$ and $K = 2$ for PSNR $\geq$ 36.5 dB. As the PSNR drops below 36.5 dB, use of $K = 2$ with any window generally results in better performance than when using $K = 1$. In this case, Window C has the best performance improvement. However, the performance is still far worse than the standard and the enhanced PCA approaches (see Section 5.5.1).

5.7 Summary

In this chapter we first reviewed important publications in the field of face recognition (Section 5.2). 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.

In Section 5.3 a new feature set (termed *DCT-mod2*) was proposed; the feature set utilizes 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 database suggest that the proposed feature set is the most robust, followed by (in order of robustness and performance): 2D Gabor wavelets, 2D DCT coefficients and PCA (eigenface) derived features. Moreover, compared to Gabor wavelets, the *DCT-mod2* feature set is over 80 times quicker to compute.

In Section 5.4 the effects of likelihood normalization in face verification were studied. Current face verification systems use a fixed threshold (or decision surface) to make the final accept or reject decision; this approach does not take into account a mismatch between training and testing conditions, where use of corrupted face images can lead to a false rejection of the claimant. To account for varying image conditions, the decision threshold can be automatically tuned through the use of likelihood normalization. The effectiveness of three likelihood normalization approaches, the Background Model Set (BMS), the Universal Background Model (UBM) and an alternate version of UBM, denoted as UBM-alt, was evaluated. Experiments using face images corrupted by an illumination direction change, compression artefacts and white Gaussian noise, show that likelihood normalization has little effect when using PCA derived features, while all three normalization approaches provide significant performance improvements when using 2D DCT, 2D Gabor wavelet or *DCT-mod2* features. Out of the three, the UBM-alt approach is the most useful, as it provides performance which is close to the best approach (BMS) while having the advantage of being client-independent. The results also show that while PCA derived features are greatly affected by an illumination direction change, they are quite immune to compression artefacts and white Gaussian noise.

In Section 5.5 we proposed to solve the fragility of PCA derived features to the illumination direction change 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 then performed. We showed that the *enhanced PCA* technique retains all the positive aspects of traditional PCA, while also being robust to changes in the illumination direction.

In Section 5.6, the *DCT-mod2* approach was extended by increasing the number of

blocks used in deriving each feature vector; moreover, windowing was introduced, allowing the variation of the contribution of each block. Results show that depending on the window used, the modified feature set is less robust compared to the original feature set when using face images corrupted with an illumination direction change; however, the modified set is more robust to compression artefacts and white Gaussian noise.

Chapter 6

Verification Using Speech and Face Information

6.1 Abstract

In this chapter we first review important concepts in the field of information fusion (Section 6.2), followed by a review of previous work on audio-visual person recognition (Section 6.3). In Section 6.4 it is shown that the weighted summation fusion approach is equivalent to a post-classifier[1] which utilizes a linear decision surface. This equivalence indicates that for a multi-expert adaptive system it is a fallacy to report the performance in noisy conditions in terms of EER (Section 6.5). Several standard non-adaptive fusion approaches are evaluated, obtaining non-optimal performance in noisy conditions (Section 6.6). Several new methods for combining speech and face information in noisy conditions are proposed, namely: a weight adjustment procedure, which explicitly measures the quality of the speech signal (Section 6.7.1); a modification to the Bayesian post-classifier, allowing the adjustment of the degree of contribution of each expert to the final verification decision (Section 6.7.2); a structurally noise resistant piece-wise linear post-classifier, which attempts to minimize the effects of noisy conditions via structural constraints on the decision boundary (Section 6.8.1); and a modification to the Bayesian post-classifier, which also attempts to impose structural constraints (Section 6.8.2).

Experimental results show that the proposed weight adjustment procedure outperforms

[1] a *post-classifier* makes the final verification decision based on the opinions of several modality experts; it is also known as a *decision stage*.

a recently published adaptive approach. Moreover, in noisy conditions, the noise resistant piece-wise linear post-classifier has similar performance to that of the proposed weight adjustment procedure, with the advantage of having a fixed (non-adaptive) structure.

Publications resulting from this research: [126, 127, 128, 129, 130, 135, 139, 140].

6.2 Introduction to Information Fusion

Broadly speaking, the term *information fusion* encompasses any area which deals with utilizing 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. It is a relatively new research area, with pioneering publications tracing back to early 1980s [14, 99, 147, 148].

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

- Utilizing 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 utilize 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 [141]). Several species of snakes combine infrared information with visual information when hunting for prey [58, 79].

This section is a review of the most important and common approaches to information fusion. In literature information fusion is often divided into three main categories (namely, sensor data level fusion, feature level fusion and decision fusion) [49, 58]. However, it is more intuitive to classify it into two main categories: *pre-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, 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 on each possible decision.

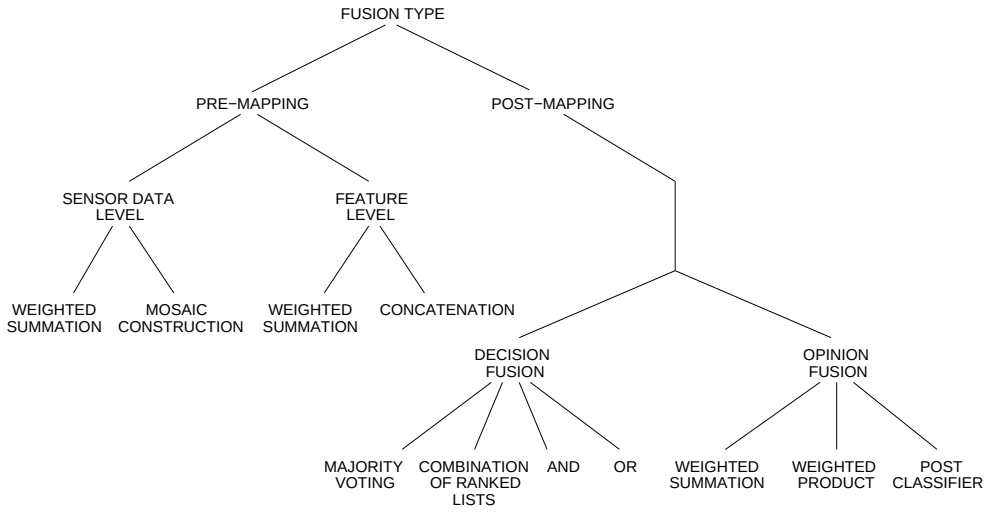

Figure 6.1: Non-exhaustive tree of fusion types

Silsbee and Bovik [141] refer to *pre-mapping fusion* and *post-mapping fusion* as *pre-categorical integration* and *post-categorical integration*, respectively, while Wark [157] refers to the terms as *input level* or *early* fusion and *classifier level* or *late* fusion, respectively.

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.

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. Review of important

milestones in the field of information fusion in audio-visual person recognition is presented in Section 6.3.

6.2.1 Pre-mapping Fusion: Sensor Data Level

In sensor data level fusion [49], 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). It must be emphasized that the data must first be *commensurate*[2], 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 [58].

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) 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, 49, 78], where a new feature vector is constructed by concatenating two or more feature vectors (e.g., to combine audio and visual features - see Figure 6.2).

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 is a problem when

[2] *commensurate*: having a common measure; equal in measure or extent; proportionate [160].

combining speech and visual feature vectors[3]. The third downside is the dimensionality of the resulting feature vector, which can lead to the "curse of dimensionality" problem [35]. Due to the above problems, in many cases the post-mapping fusion approach is preferred (described in Sections 6.2.3 and 6.2.4).

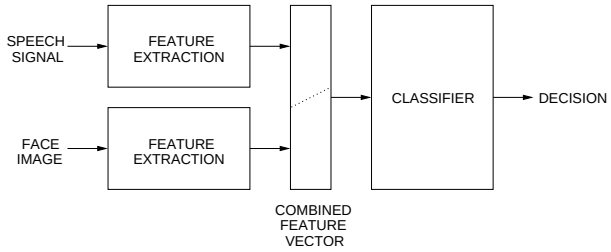

Figure 6.2: Conceptual example of classification using feature level fusion

6.2.3 Post-Mapping Fusion: Decision Fusion

In decision fusion [49, 58], 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 recognizing a particular set of classes while being "bad" at recognizing a different set of classes. Thus a combination of classifiers may overcome the "bad" properties of each classifier [54, 68].

6.2.3.1 Majority Voting

In majority voting [44, 58, 107], 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

[3]For example, speech feature vectors are usually extracted at a rate of 100 per second (see Chapter 3) 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 [149]).

approach; an odd number of classifiers is required to prevent ties; moreover, the number of classifiers must be greater than the number of classes (possible decisions) to ensure a decision is reached. See Figure 6.3 for a conceptual example of classification using majority voting.

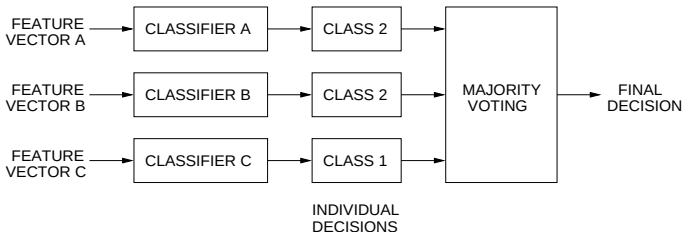

Figure 6.3: Conceptual example of classification using majority voting

6.2.3.2 Ranked List Combination

In ranked list combination [2, 54, 107], 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 [54], 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; see Figure 6.4 for an example.

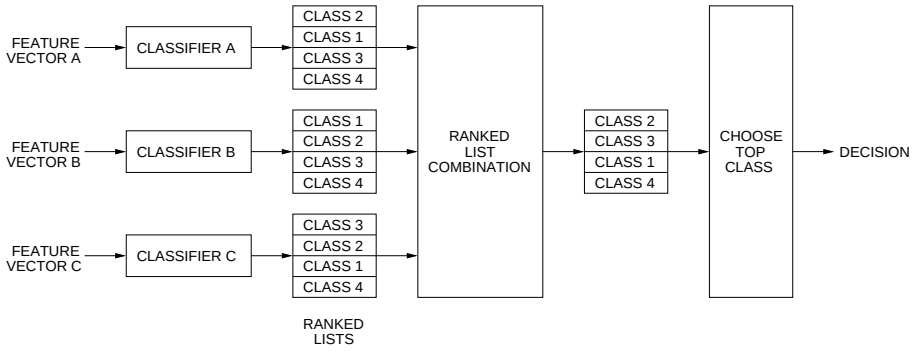

Figure 6.4: Conceptual example of classification using ranked list combination

6.2.3.3 AND Fusion

In AND fusion [79, 153], 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 we would like to detect the presence of a true claimant, this translates to a high FR% and low FA%).

6.2.3.4 OR Fusion

In OR fusion [79, 153], 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 (in a person verification scenario, where we would like to detect the presence of a true claimant, this translates to a low FR% and high FA%).

6.2.4 Post-Mapping Fusion: Opinion Fusion

In opinion fusion [49, 58, 154], 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, where 0 indicates the lowest opinion and 1 the highest opinion. It must be noted that 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).

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.

It must be noted that often in literature (e.g., [49, 58, 154]) the term "decision fusion" also encompasses opinion fusion. However, since each expert provides an opinion and not a decision, the term "decision fusion" is in this case a misnomer.

Opinions can be combined using weighted summation or weighted product approaches (described in Sections 6.2.4.1 and 6.2.4.2, respectively) before using a classification criterion, such as the MAX operator (which selects the class with the highest opinion), to reach a decision (see Figure 6.5). Alternatively, a post-classifier (Section 6.2.4.3) 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 [35].

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 are selected to reflect the reliability and discrimination ability of each expert. Thus 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*.

6.2.4.1 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 constraint $\sum_{i=1}^{N_E} w_i = 1$. When all the weights are equal, Eqn. (6.1) reduces to an arithmetic mean operation. The weighted summation approach is also known as *linear opinion pool* [6] and *sum rule* [5, 68].

When thinking of the experts as elaborate discriminant functions, the weighted summation approach is somewhat analogous to the linear combination of Gaussian functions in the Gaussian Mixture Model based classifier (described Chapter 2) [35].

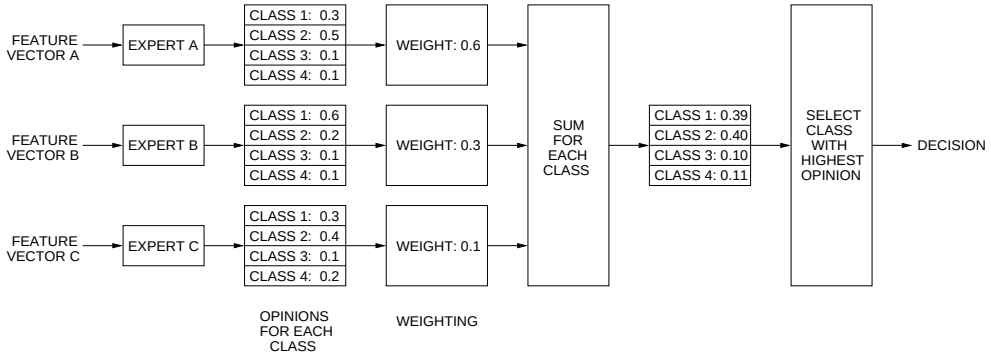

Figure 6.5: Conceptual example of classification using weighted summation

6.2.4.2 Weighted Product Fusion

The opinions can be interpreted as *a posteriori* probabilities in the Bayesian framework [21]. Assuming the experts are independent, the opinions regarding class j from N_E experts can be combined using a product rule:

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

Moreover, to account for varying discrimination ability and reliability of each expert, weighting is introduced:

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

When all the weights are equal, Eqn. (6.3) reduces to a geometric mean operation. The weighted product approach is also known as *logarithmic opinion pool* [6] and *product rule* [5, 68].

There are two downsides to weighted product fusion: the first is that 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[4]. The second downside is that the independence assumption is only strictly valid when each expert is using independent features.

[4]From a different point of view, the effect of setting the opinion close to zero may be desirable in a high security application; for example, when either the speech or the face expert gives a very low opinion.

6.2.4.3 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*[5]. It must be noted that the opinions do not necessarily need 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 only dependent on the number of experts [17]. Each expert provides only one opinion, indicating the likelihood that a given claimant is the true claimant. Thus a low opinion suggests that the claimant is an impostor, while a high opinion suggests that the claimant is the true claimant. The post-classifier then provides a decision surface in N_E-dimensional space, separating the impostor and true claimant classes[6]. See Figure 6.6 for a conceptual example of classification using a post-classifier.

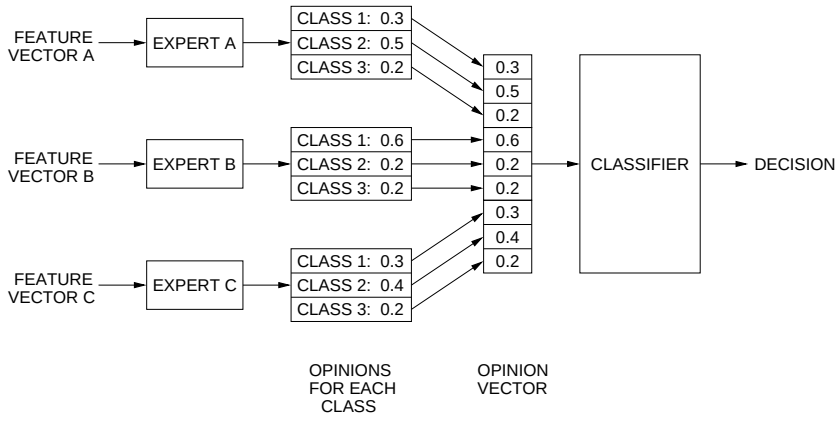

Figure 6.6: Conceptual example of classification using a post-classifier

[5]In the identification scenario, the described post-classifier is a natural extension of the approach presented in [7]. In the verification scenario it has been implemented by Ben-Yacoub *et al.* [17] as a binary classifier.

[6]see Figure 6.8 for example decision surfaces.

6.3 Previous Work in Audio-Visual Person Recognition

This section provides an overview of the most important contributions in the field of audio-visual person recognition. It concentrates on the verification task while briefly touching on the identification task. Almost all of the work reviewed here used different databases and/or different experimental setup (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. It is assumed that the reader is familiar with the concepts presented in Section 6.2.

The review 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 is 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.* [28] in 1993 and Brunelli *et al.* [20, 21] in 1995.

In [28], Chibelushi *et al.* combined information from still face profile images and speech using a form of weighted summation fusion:

$$f = w_1 o_1 + w_2 o_2 \tag{6.4}$$

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 (i.e., a low opinion suggests that the claimant is an impostor, while a high opinion suggests that the claimant is the true claimant). Since there is a constraint on the weights ($\sum_{i=1}^{2} w_i = 1$), Eqn. (6.4) reduces to:

$$f = w_1 o_1 + (1 - w_1) o_2 \tag{6.5}$$

The verification decision was reached via thresholding the fused opinion, f. When using the speech expert alone (i.e., $w_1 = 1$), an 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 the EER was reduced to 1.5%.

In [20], Brunelli *et al.* combined the opinions from a face expert (which utilized geometric features obtained from static frontal face images) and a speech expert using the weighted product approach:

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

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 [21], Brunelli *et al.* used two speech experts (for static and delta features) and three face experts (for the eye, nose and mouth areas of the face) for person identification. The weighted product approach was used to fuse the opinions, with the weights found automatically via 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.* [32] 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 utilized. 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.

In [67], Kittler *et al.* 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. No results were provided for using more than six images. The results suggest that using a video sequence of the face, rather than one image, provides superior performance.

In [68], Kittler *et al.* attempted to provide theoretical foundations for common fusion approaches such as the summation and product methods. However, by the authors' own admission, the foundations utilized assumptions which are "unrealistic in most applications". Experimental 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 [78] 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.

Jourlin *et al.* [63, 64] 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.

Hong and Jain [55] 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 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 (i.e., the fingerprint expert was significantly slower than the face expert). It was shown that in all tested cases fusion led to better performance than using either expert alone.

Ben-Yacoub *et al.* [17] 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 [154] also investigated various binary classifiers for opinion fusion as well as the majority voting and 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. Verlinde also attempted to develop a piece-wise linear classifier but obtained poor results.

In [155], Wark *et al.* 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). Experimental 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 Wark refers to as *catastrophic fusion* [157]). The authors proposed a statistically inspired method of *a priori* weight selection (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 and no results were reported for SNR levels below 10 dB; 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:

$$w_1 = \frac{\zeta_2}{\zeta_1 + \zeta_2} \tag{6.7}$$

where

$$\zeta_i = \sqrt{\frac{\sigma_{i,true}^2}{N_{true}} + \frac{\sigma_{i,imp}^2}{N_{imp}}} \tag{6.8}$$

where, for the i-th expert, ζ_i is the standard error [29] of the difference between sample means $\mu_{i,true}$ and $\mu_{i,imp}$ of opinions for true and impostor claims, respectively, $\sigma_{i,true}^2$ and $\sigma_{i,imp}^2$ 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 is a constraint on the weights ($\sum_{i=1}^{2} w_i = 1$), 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; thus a low standard error indicates better performance.

6.3.2 Adaptive Approaches

In [156] Wark *et al.* extended the work presented in [155] (see above) by proposing a heuristic method to adjust the weights. Experimental 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 [155] (above).

In further work [157], 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:

$$w_1 = \left[\frac{\zeta_2}{\zeta_1 + \zeta_2}\right]\left[\frac{\kappa_1}{\kappa_1 + \kappa_2}\right] \tag{6.9}$$

where $\frac{\zeta_2}{\zeta_1+\zeta_2}$ was found using Eqn. (6.8) during training and

$$\kappa_i = \frac{|\mathcal{M}(o_i)_{i,true} - \mathcal{M}(o_i)_{i,imp}|}{\mu_{i,true}} \tag{6.10}$$

was found during testing. Wark referred to κ_i as an *a posteriori* 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 Mahalanobis distance [35] 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.

Similarly, $\mathcal{M}(o_i)_{i,imp} = \frac{(o_i-\mu_{i,imp})^2}{\sigma^2_{i,imp}}$ is the one dimensional Mahalanobis 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.

6.4 Equivalence of the Weighted Summation and Post-Classifier Approaches

From the descriptions presented in Section 6.2, it is evident that the weighted summation approach is the most flexible fusion technique for most applications. In a verification application, which utilizes N_E experts, opinion fusion is accomplished using:

$$f = \sum_{i=1}^{N_E} w_i o_i \tag{6.11}$$

where o_i is the opinion of the i-th expert (in the [0,1] interval), with corresponding weight w_i (also in the [0,1] interval). Each 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). The verification 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. (6.11) can be modified to:

$$F(\vec{o}) = \vec{w}^{\,T}\vec{o} - t \tag{6.12}$$

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

It can be seen that Eqn. (6.12) is a form of a linear discriminant function [35], indicating that the procedure of weighted summation and 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.

6.5 Performance Measurement of Multi-Expert Systems

The equivalency described in Section 6.4 has several implications on the measurement of performance. As described in Section 2.4, the EER is traditionally used as a measure of expected performance of a verification system. In a single expert configuration this amounts to selecting the appropriate *a posteriori* threshold; in a multi-expert scenario this translates to selecting appropriate *a posteriori* parameters for the post-classifier (in the above case $\vec{w}$ 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 [157]). 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. (6.11)] 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 adaptive and non-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 standard practice of using the *a posteriori* threshold in single-expert systems); performance in noisy conditions is then reported in terms of FA% & FR%, where the post-classifier parameters are fixed (in non-adaptive systems), or automatically varied (in adaptive systems). The results are also reported graphically, by combining FA% & FR% into one number; this is accomplished by using a quantity referred to as Total Error (TE), defined as:

$$\text{TE} = \text{FA\%} + \text{FR\%} \tag{6.13}$$

6.6 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 (Section 6.6.2) and SVM (Section 6.6.3) post-classifiers), for combining speech and face information.

6.6.1 Mapping Opinions to the [0,1] Interval

The experiments reported throughout this chapter utilize the following method (inspired by [63]) 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 sigmoïd:

$$o_i = \frac{1}{1 + \exp[-\tau_i(o_{i,\text{orig}})]} \tag{6.14}$$

where

$$\tau_i(o_{i,\text{orig}}) = \frac{o_{i,\text{orig}} - (\mu_i - 2\sigma_i)}{2\sigma_i} \tag{6.15}$$

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, 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 [35]. Eqn. (6.15) maps the opinions to the $[-2, 2]$ interval, which corresponds to the approximately linear portion of the sigmoïd in Eqn. (6.14). The sigmoïd is necessary to take care of outliers and situations where the assumptions do not hold entirely.

6.6.2 Bayesian Post-Classifier

A Bayesian post-classifier has been previously used by Ben-Yacoub *et al.* [17] and Abdeljaoued [1]. The classifier is similar to the classifier described in Chapter 2. The only difference is that rather than using multiple observation vectors $X = \{\vec{x}_i\}_{i=1}^{N_V}$ in Eqn. (2.17), a single opinion vector $\vec{o}^T = [\ o_i\ ]_{i=1}^{N_E}$ is used. Formally, the decision rule described in Eqn. (2.17) is expressed as:

$$\text{chosen class} = \begin{cases} C_1 & \text{if} \quad \Lambda(\vec{o}) \geq t \\ C_2 & \text{otherwise} \end{cases} \tag{6.16}$$

where C_1 and C_2 are the true claimant and impostor classes, respectively. By following Eqns. (2.14) and (2.16), $\Lambda(\vec{o})$ expands to:

$$\Lambda(\vec{o}) = \log \tilde{p}(\vec{o}|C_1) - \log \tilde{p}(\vec{o}|C_2) \tag{6.17}$$

where $\tilde{p}(\vec{o}|C_j)$ is a parametric representation of $p(\vec{o}|C_j)$. We shall utilize GMMs to provide the parametric representation of the distribution of opinions:

$$\Lambda(\vec{o}) = \log \tilde{p}(\vec{o}|\lambda_{\text{true}}) - \log \tilde{p}(o_i|\lambda_{\text{imp}}) \tag{6.18}$$

where, λ_{true} and λ_{imp} are the GMM parameters of the distribution of opinions for true and impostor claims.

6.6.3 Support Vector Machine Post-Classifier

The Support Vector Machine (SVM) [152] has been previously used by Ben-Yacoub *et al.* [17] as a post-classifier. While an in-depth description of SVM is beyond the scope of this

section, important points are summarized. For more detail, the reader is referred to [23].

The SVM is based on the principle of Structural Risk Minimization (SRM) as opposed to Empirical Risk Minimization (ERM) used in classical learning approaches. Under ERM, the criteria for an optimal decision surface depends on the classifier structure used (e.g., k-Nearest Neighbour or Maximum Likelihood). Without testing on a separate data set, it is unknown which decision surface has the best generalization capability. Under SRM, the decision surface has to satisfy a structural requirement which is thought to obtain the best generalization 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 to the closest points of the two classes; we interpret the meaning of the margin as a measure of generalization capability. Thus using the SRM principle, the optimal decision surface has the maximum *margin*.

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

$$S = \left\{ \ (\vec{o}_i, y_i) \mid \vec{o}_i \in \mathbb{R}^{N_E}, y_i \in \{-1, +1\} \ \right\}_{i=1}^{N_V} \tag{6.19}$$

The SVM uses the following function, which implements the optimal decision surface in SRM sense [152], to map a given vector to its label space (i.e., -1 or $+1$):

$$f(\vec{o}) = \text{sign}\left(\sum_{i=1}^{N_V} \alpha_i y_i K(\vec{o}_i, \vec{o}) + b \right) \tag{6.20}$$

where vectors $\vec{o}_i$ with corresponding $\alpha_i > 0$ are known as *support vectors*. $K(\vec{d}, \vec{e})$ is a positive definite symmetric kernel function, subject to Mercer's condition [23, 152]. $\vec{\alpha}^T = [\alpha_i]_{i=1}^{N_V}$ is found by minimizing (via quadratic programming):

$$-\sum_{i=1}^{N_V} \alpha_i + \frac{1}{2} \vec{\alpha}^T D \vec{\alpha} \tag{6.21}$$

subject to constraints:

$$\vec{\alpha}^T \vec{y} = 0 \tag{6.22}$$

$$\alpha_i \in [0, C] \ \forall \ i \tag{6.23}$$

where, $\vec{y}^T = [\ y_i\ ]_{i=1}^{N_V}$ and, typically, $C = 1000$. The elements of D are defined as:

$$D_{ij} = y_i y_j K(\vec{o}_i, \vec{o}_j) \tag{6.24}$$

The parameter b is estimated after $\vec{\alpha}$ has been found [23]. The kernel function $K(\vec{d}, \vec{e})$ implements a dot product in a high dimensional space, $\mathbb{R}^h$ (where $h > N_E$), which improves separability of the data [121]. Popular kernels used for pattern recognition problems are [23]:

$$K(\vec{d}, \vec{e}) = (\vec{d}^{\,T} \vec{e} + 1)^p \tag{6.25}$$

$$K(\vec{d}, \vec{e}) = \exp(-\gamma ||\vec{d} - \vec{e}||^2) \tag{6.26}$$

$$K(\vec{d}, \vec{e}) = \tanh(\kappa \vec{d}^{\,T} \vec{e} - \delta) \tag{6.27}$$

Eqn. (6.25) is a p-th degree polynomial, Eqn. (6.26) is a Radial Basis Function (RBF) while Eqn. (6.27) is a hyperbolic tangent (sigmoïd), often used in Artificial Neural Networks [35]. Ben-Yacoub *et al.* [17] obtained the best results using the kernel defined by Eqn. (6.25).

The experiments reported in this section utilize the SVM engine developed by Joachims [59]. In a verification system there is generally more training data for the impostor class than the true claimant class; thus a misclassification 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 minimizes the *total* misclassification rate (subject to SRM constraints), 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.

6.6.4 Experiment Setup and Results

The experiments were done on the VidTIMIT database (see Chapter 4). The speech and frontal face experts are described in detail in Chapters 3 and 5, respectively. Both experts used eight-Gaussian client models, utilized the BMS likelihood normalization technique, and were trained using utterances from Session 1. Based on the results from Section 3.4.2, CMS+Δ+MACV feature extraction was used for speech signals. The PCA based approach (eigenfaces) was used for frontal face feature extraction; it was selected since it extracts one feature vector per video frame, allowing feature vector concatenation fusion to be used.

To find the performance, Sessions 2 and 3 were used for obtaining expert opinions of known impostor and true claims. Four utterances, each from 8 fixed persons (4 male and 4 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. 10 background models were selected from the 35 client models ($N_\Phi = N_\Psi = 10$). In total, there were 1120 impostor and 140 true claims.

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 [78]. Thus in the experiments reported in this section, the information loss is avoided by utilizing the former approach of artificially increasing the video frame rate.

Speech signals were corrupted by additive white Gaussian noise, with the SNR varying from 28 to -8 dB. Following the arguments presented in Section 6.5, post-classifier parameters (when using multiple experts) and the decision threshold (when using feature vector concatenation) were found to obtain performance as close as possible to EER on

clean test data. Opinion mapping parameters (Section 6.6.1) were also found on clean test data.

The parameters for weighted summation fusion were found via an exhaustive search procedure. As done by the speech expert, the feature vectors resulting from feature vector concatenation were processed by the VAD (described in Section 3.3.5).

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 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 $\vec{o} = [\ o_1\ o_2\ ]^T$. When used alone, the face expert obtained an EER of 7.14% (TE = 14.28).

For the Bayesian post-classifier, results using GMMs with 1, 2 and 3 Gaussians are reported. For the SVM post-classifier, results for the polynomial kernel [see Eqn. (6.25)] and the RBF kernel [see Eqn. (6.26)] are reported. When using the sigmoïd kernel [see Eqn. (6.27)], the SVM engine failed to converge. Performance of SVM using the polynomial kernel was evaluated with the parameter p equal to 1, 2 and 3. For the RBF kernel, γ was set to 1.

Results are presented in terms of FA% and FR% in Tables 6.1 through 6.10 and in terms of TE in Figure 6.7. For the Bayesian post-classifier, the TE is plotted only for the case of single-Gaussian GMMs, as this configuration provided the best overall result. Similarly for the SVM post-classifier, the TE is plotted only for the case of polynomial kernel with $p = 2$.

Figures 6.8 to 6.13 show the distribution of opinion vectors in clean and noisy (SNR=-8dB) conditions, with the decision boundaries used by the three approaches.

SNR (dB)	∞	28	24	20	16	12	8	4	0	-4	-8
FA%	12.41	12.14	12.14	12.05	12.32	12.68	12.50	10.89	9.20	8.04	8.39
FR%	12.14	12.14	12.14	12.14	12.14	10.71	13.57	27.14	44.29	67.14	81.43

Table 6.1: Performance of the speech expert

SNR (dB)	∞	28	24	20	16	12	8	4	0	-4	-8
FA%	10.00	9.91	9.91	9.91	9.91	9.91	9.91	10.00	10.00	9.91	9.91
FR%	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00

Table 6.2: Performance of feature vector concatenation fusion

SNR (dB)	∞	28	24	20	16	12	8	4	0	-4	-8
FA%	2.86	2.95	3.04	3.04	2.95	2.86	2.59	2.41	1.96	1.07	0.89
FR%	2.86	2.86	2.86	3.57	5.71	6.43	6.43	8.57	12.86	26.43	41.43

Table 6.3: Performance of weighted summation fusion

SNR (dB)	∞	28	24	20	16	12	8	4	0	-4	-8
FA%	3.57	3.57	3.75	3.75	3.75	3.75	3.75	3.66	2.77	2.05	1.61
FR%	3.57	3.57	2.86	2.86	2.86	2.86	5.00	6.43	12.14	27.14	45.71

Table 6.4: Performance of the Bayesian post-classifier, 1-Gaussian GMMs

SNR (dB)	∞	28	24	20	16	12	8	4	0	-4	-8
FA%	2.95	2.95	3.12	3.21	3.12	2.95	2.68	2.41	1.96	1.07	0.80
FR%	2.86	2.86	2.14	2.86	4.29	4.29	5.71	7.86	13.57	32.14	48.57

Table 6.5: Performance of the Bayesian post-classifier, 2-Gaussian GMMs

SNR (dB)	∞	28	24	20	16	12	8	4	0	-4	-8
FA%	2.95	3.04	3.12	3.12	3.21	2.95	2.95	2.23	1.96	0.98	0.62
FR%	2.86	2.86	2.14	2.86	2.86	3.57	5.71	8.57	14.29	33.57	49.29

Table 6.6: Performance of the Bayesian post-classifier, 3-Gaussian GMMs

SNR (dB)	∞	28	24	20	16	12	8	4	0	-4	-8
FA%	2.95	2.95	3.12	3.04	2.95	2.86	2.59	2.41	1.96	1.16	0.98
FR%	2.86	3.57	3.57	4.29	5.71	6.43	6.43	8.57	12.86	25.00	40.71

Table 6.7: Performance of the SVM post-classifier using polynomial kernel, $p = 1$

SNR (dB)	∞	28	24	20	16	12	8	4	0	-4	-8
FA%	3.12	3.12	3.12	3.04	3.04	3.04	3.30	3.04	2.77	1.70	1.16
FR%	2.86	3.57	3.57	3.57	5.71	5.71	6.43	7.14	10.71	19.29	33.57

Table 6.8: Performance of the SVM post-classifier using polynomial kernel, $p = 2$

SNR (dB)	∞	28	24	20	16	12	8	4	0	-4	-8
FA%	2.86	2.86	3.12	3.04	3.04	3.04	2.86	2.59	2.23	1.34	1.07
FR%	2.86	3.57	2.86	4.29	5.71	6.43	6.43	8.57	10.71	23.57	38.57

Table 6.9: Performance of the SVM post-classifier using polynomial kernel, $p = 3$

SNR (dB)	∞	28	24	20	16	12	8	4	0	-4	-8
FA%	3.04	2.86	3.12	3.12	3.04	3.04	2.86	2.59	2.50	1.34	1.07
FR%	2.86	2.86	2.86	4.29	5.71	6.43	6.43	8.57	11.43	20.71	37.14

Table 6.10: Performance of the SVM post-classifier using RBF kernel

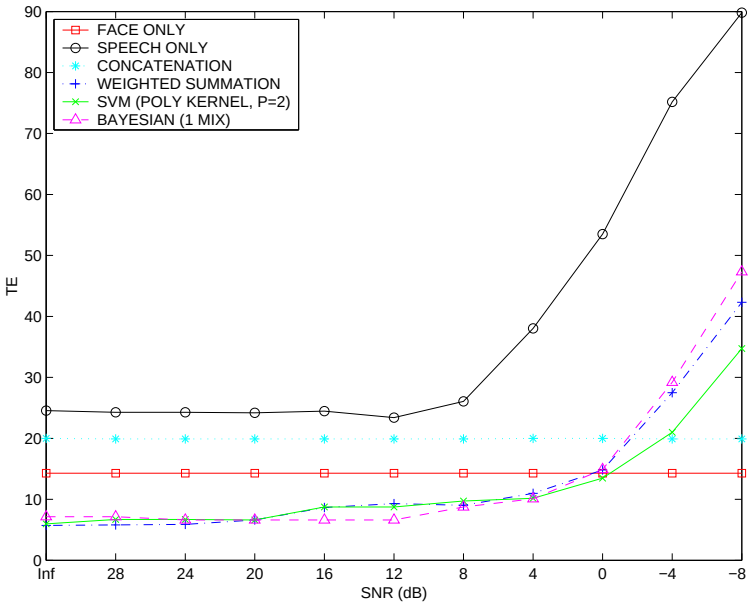

Figure 6.7: Performance of various non-adaptive fusion approaches

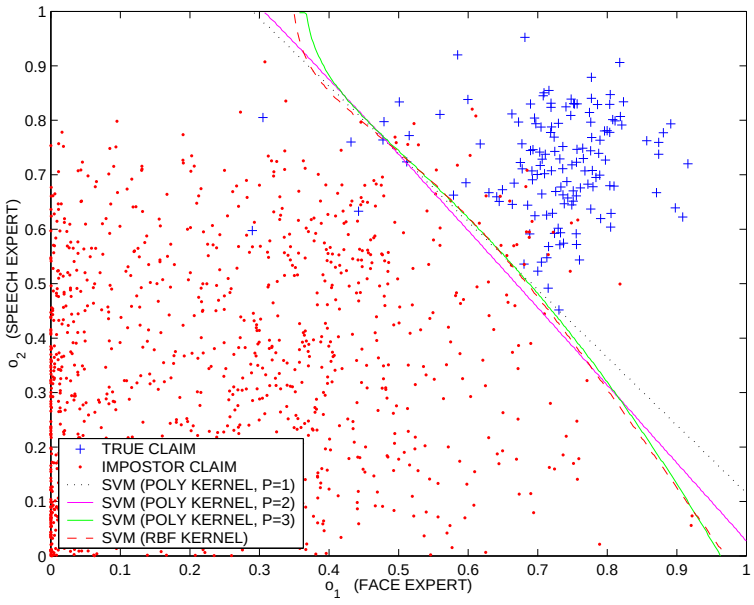

Figure 6.8: Decision boundaries used by SVM (various kernels) and distribution of opinion vectors for true & impostor claims using clean speech

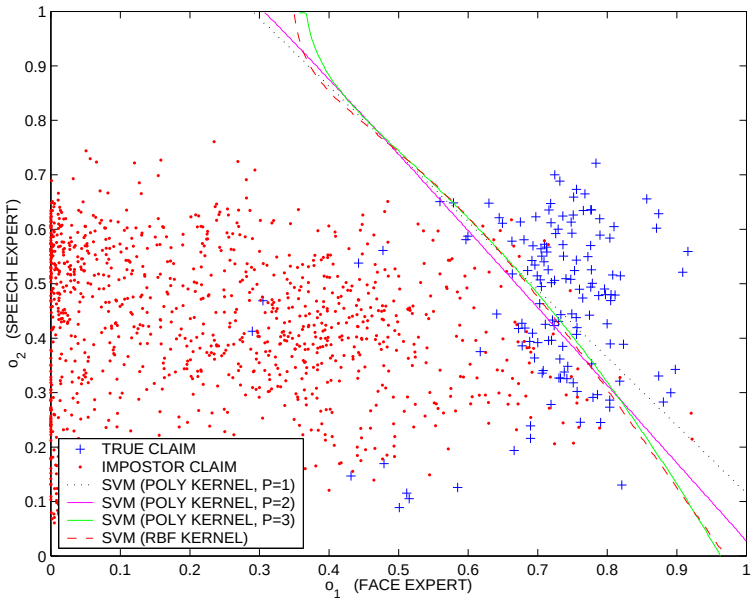

Figure 6.9: As per Figure 6.8 but using noisy speech

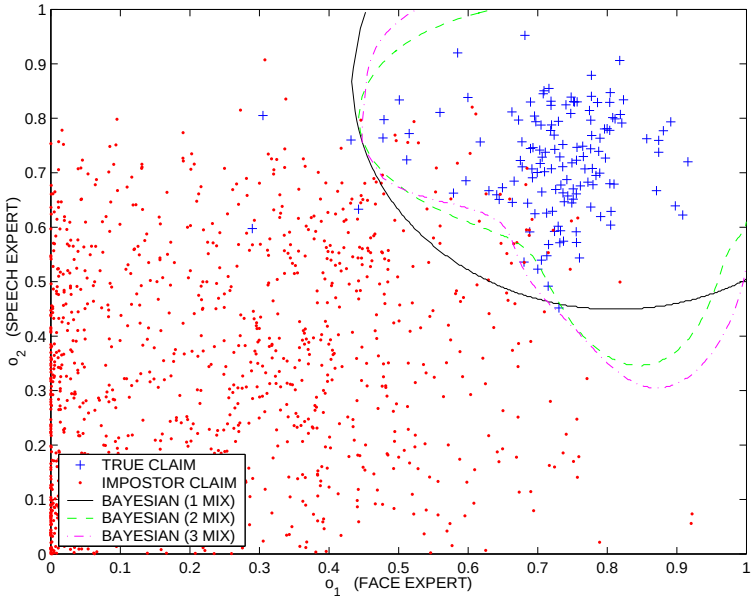

Figure 6.10: Decision boundaries used by Bayesian post-classifier and distribution of opinion vectors for true & impostor claims using clean speech

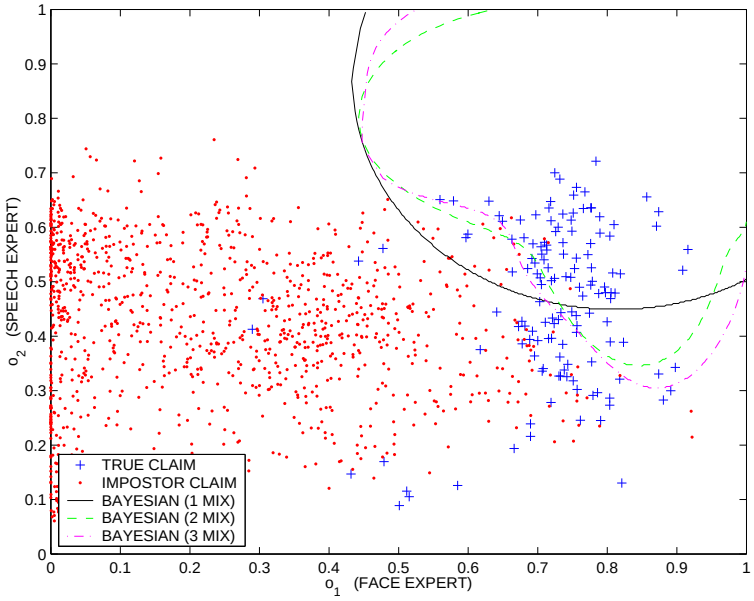

Figure 6.11: As per Figure 6.10 but using noisy speech

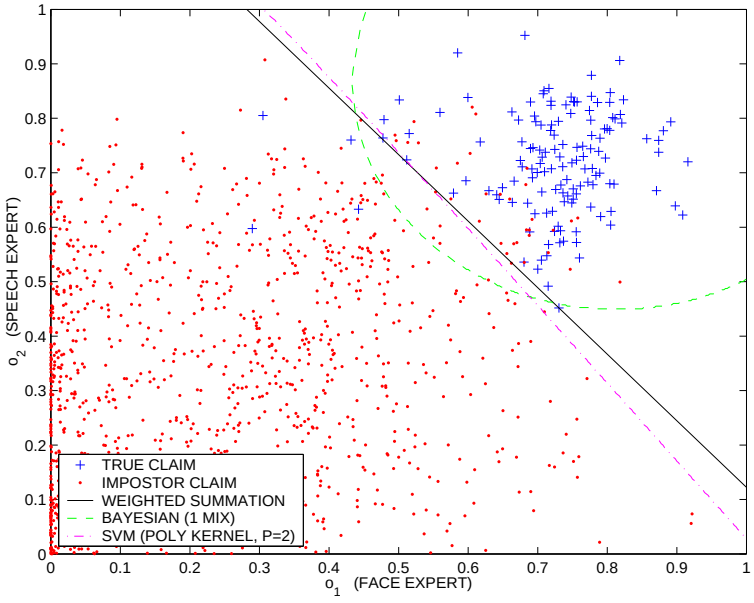

Figure 6.12: Decision boundaries used by the weighted summation approach and Bayesian & SVM post-classifiers, and distribution of opinion vectors for true & impostor claims using clean speech

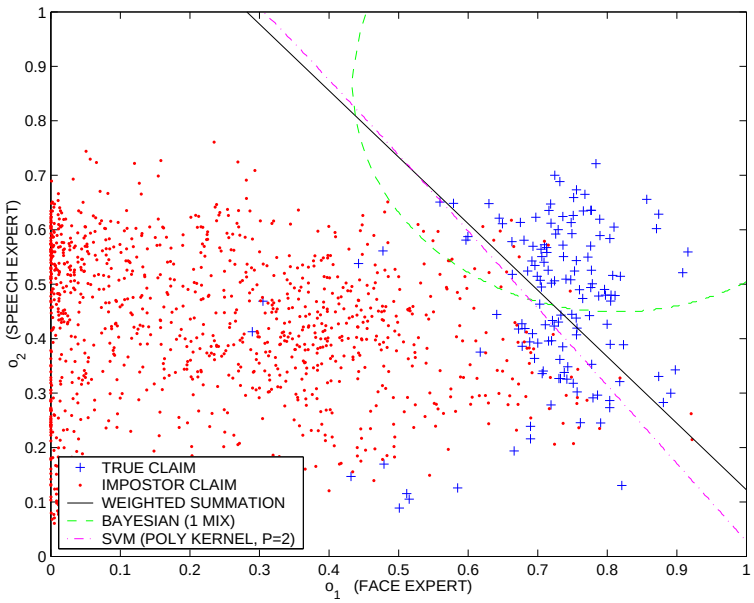

Figure 6.13: As per Figure 6.12 but using noisy speech

6.6.5 Discussion

6.6.5.1 Effect of Noisy Conditions on 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 6.8 and 6.9, noisy conditions cause the mean of the true claim opinion distribution to move toward the o_1 axis while the variance of the impostor opinion distribution is decreased along the o_2 axis. The changes can be explained by analyzing Eqn. (2.16):

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

where C_1 and C_2 are the true claimant and impostor classes, respectively, X is a set of feature vectors and $\mathcal{L}(X|C_j)$ is the average log likelihood function, defined in Eqn. (2.14).

Let us suppose a true claim has been made. In clean conditions $\mathcal{L}(X|C_1)$ will be high while $\mathcal{L}(X|C_2)$ will be low, causing $\Lambda(X)$ to be high. Due to the mismatch between training and testing conditions, the feature vectors drift away from the feature space described by

the true claimant model (parametric model representing C_1), causing $\mathcal{L}(X|C_1)$ to decrease. If $\mathcal{L}(X|C_2)$ decreases by the same amount as $\mathcal{L}(X|C_1)$, then $\Lambda(X)$ is relatively unchanged. However, to model possible impostors, the parametric model representing C_2 (i.e., the impostor model) may cover a wide area of the feature space (see Section 2.3.2). 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 impostor model, causing $\mathcal{L}(X|C_2)$ to decrease by a smaller amount, which in turn causes $\Lambda(X)$ to decrease.

Let us now suppose that several impostor claims have been made. In clean conditions $\mathcal{L}(X|C_1)$ will be low while $\mathcal{L}(X|C_2)$ will be high, causing $\Lambda(X)$ to be low. Recall that additive white noise is used to cause the mismatch between training and testing conditions and that the speech expert utilizes MFCC feature vectors which represent the instantaneous Fourier spectrum (see Section 3.3.1). As the noise level is increased, the spectrum becomes flatter, causing the MFCC feature vectors obtained from different impostors to be similar. Hence the mismatch causes the variance of $\mathcal{L}(X|C_2)$ to be reduced. The true claimant model does not represent the impostor feature space, indicating that $\mathcal{L}(X|C_1)$ should be consistently low for impostor claims. Thus if the variance of $\mathcal{L}(X|C_2)$ is reduced, then the variance of $\Lambda(X)$ should also be reduced.

While Figures 6.8 and 6.9 were obtained by corrupting the speech signals with additive white Gaussian noise, we would expect a similar movement of the mean of the true claim opinion distribution for other noise types. Generally any noise types alters the features obtained, which would cause $\mathcal{L}(X|C_1)$ to decrease, and as explained above, this leads to a decrease of $\Lambda(X)$.

6.6.5.2 Effect of Noisy Conditions on Performance

As shown in Figure 6.8, utilizing different kernels for the SVM post-classifier results in similar decision boundaries.

For the polynomial kernel with $p = 2$, the decision boundary is the furthest away from the distribution of opinion vectors for true claims, translating to less misclassification of true claim opinions in noisy conditions (where the true claim opinion vectors have "moved"

toward the o_1 axis), than when using other kernels (see Figure 6.9).

For the Bayesian post-classifier, single-Gaussian GMMs provide the best performance in noisy conditions. As for the SVM post-classifier, this can be attributed to the decision boundary used. As can be observed in Figures 6.10 and 6.11, the decision boundaries for 2- and 3-Gaussian models envelop the true claimant opinion distribution more closely than single-Gaussian models, resulting in better performance in clean conditions. However, in noisy conditions the performance is poorer, since more true claimant opinion vectors are misclassified.

The remainder of the discussion assumes that the SVM post-classifier uses the polynomial kernel with $p = 2$ and that the Bayesian post-classifier uses single-Gaussian GMMs.

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 4 dB. As the SNR decreases further, the SVM post-classifier is the least affected, followed by the weighted summation approach and finally the Bayesian post-classifier. The differences in performance in noisy conditions can be attributed to the decision boundaries used by each approach, shown in Figures 6.12 and 6.13. It can be seen that the SVM post-classifier utilizes a decision surface which results in the least misclassifications of true claimant opinion vectors in noisy conditions.

The performance of the feature concatenation fusion approach stays relatively constant while the SNR is lowered. However, for all SNRs the performance is worse than the face expert, indicating that while feature concatenation fusion is 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 information to the classifier. 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.

6.7 Performance of Adaptive Approaches in Noisy Conditions

In this section we evaluate the performance of several adaptive opinion fusion methods, namely weighted summation fusion with Wark's weight selection (described in Section 6.3.2), weighted summation fusion with the weight adjustment proposed in Section 6.7.1 and the Modified Bayesian Classifier described in Section 6.7.2 also using the proposed weight adjustment method.

6.7.1 Proposed Weight Adjustment Method

As shown in Section 3.4.2, the MFCC features are most susceptible to changes in the SNR. We exploit this to detect the amount of mismatch between the training and testing conditions, and hence adjust the weight of the speech expert accordingly.

Every time a speech utterance is recorded, it is preceded by a short segment which contains only ambient noise. N_{noise} MFCC feature vectors from the noise segment from each training utterance[7] are used to construct a global background noise GMM, λ_{noise}. Given a test utterance, N_{noise} MFCC feature vectors, $\{\vec{x_i}\}_{i=1}^{N_{\text{noise}}}$, representing the noise segment, are used to estimate the utterance's quality by measuring the mismatch from λ_{noise} as follows:

$$q = \frac{1}{N_{\text{noise}}} \sum_{i=1}^{N_{\text{noise}}} \log p(\vec{x_i}|\lambda_{\text{noise}}) \tag{6.28}$$

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

$$q_{\text{map}} = \frac{1}{1 + \exp[-a(q - b)]} \tag{6.29}$$

[7]Thus if there is 10 training utterances, $10 \times N_{\text{noise}}$ MFCC feature vectors are used.

where a and b describe the shape of the sigmoïd. The values of a and b are selected so that q_{map} is close to one for clean training utterances and close to zero for training utterances artificially corrupted with noise (thus 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 $w_{2,\text{apriori}}$ for the speech expert (found for clean conditions), the adapted weight for the speech expert is found using:

$$w_2 = q_{\text{map}} w_{2,\text{apriori}} \tag{6.30}$$

Since we are using a two modal system, there is a $\sum_{i=1}^{2} w_i = 1$ constraint on the weights. Thus the corresponding weight for the video expert is then found using: $w_1 = 1 - w_2$.

6.7.2 Modified Bayesian Post-Classifier

As explained in Section 6.6.2, the Bayesian post-classifier is similar to the classifier described in Chapter 2. The main difference is that instead of using multiple observation vectors $X = \{\vec{x}_i\}_{i=1}^{N_V}$ in (2.17), a single opinion vector $\vec{o}^T = [\ o_i\ ]_{i=1}^{N_E}$ is used. Formally, the decision rule described in (2.17) is expressed as:

$$\text{chosen class} = \begin{cases} C_1 & \text{if} \quad \Lambda(\vec{o}) \geq t \\ C_2 & \text{otherwise} \end{cases} \tag{6.31}$$

where C_1 and C_2 are the true claimant and impostor classes, respectively. By following Eqns. (2.14) and (2.16), $\Lambda(\vec{o})$ expands to:

$$\Lambda(\vec{o}) = \log \tilde{p}(\vec{o}|C_1) - \log \tilde{p}(\vec{o}|C_2) \tag{6.32}$$

where $\tilde{p}(\vec{o}|C_j)$ is a parametric representation of $p(\vec{o}|C_j)$. By assuming the opinions are independent (a reasonable assumption when dealing with experts processing speech and face data), Eqn. (6.32) is modified to:

$$\Lambda(\vec{o}) = \log \left[\prod_{i=1}^{N_E} \tilde{p}(o_i|C_1) \right] - \log \left[\prod_{i=1}^{N_E} \tilde{p}(o_i|C_2) \right] \tag{6.33}$$

which is simplified to:

$$\Lambda(\vec{o}) = \sum_{i=1}^{N_E} \log \tilde{p}(o_i|C_1) - \sum_{i=1}^{N_E} \log \tilde{p}(o_i|C_2) \tag{6.34}$$

We shall utilize GMMs to provide the parametric representation of the distribution of opinions for each expert:

$$\Lambda(\vec{o}) = \sum_{i=1}^{N_E} \log \tilde{p}(o_i|\lambda_{i,\text{true}}) - \sum_{i=1}^{N_E} \log \tilde{p}(o_i|\lambda_{i,\text{imp}}) \tag{6.35}$$

where, for the i-th expert, $\lambda_{i,\text{true}}$ and $\lambda_{i,\text{imp}}$ are the GMM parameters of the distribution of opinions for true and impostor claims. To allow a decrease of the contribution of an expert which is affected by noisy conditions, the Bayesian classifier is further modified by introducing weighting:

$$\Lambda(\vec{o}) = \sum_{i=1}^{N_E} w_i \log \tilde{p}(o_i|\lambda_{i,\text{true}}) - \sum_{i=1}^{N_E} w_i \log \tilde{p}(o_i|\lambda_{i,\text{imp}}) \tag{6.36}$$

where the weights have a $\sum_{i=1}^{N_E} w_i = 1$ constraint.

6.7.3 Experimental Setup and Results

The experimental setup is similar to the one described in Section 6.6.4. Based on manual observation of plots of speech signals from the VidTIMIT database, N_{noise} was set to 30 for the proposed adaptive weight adjustment method [see Eqn. (6.28)]. A single Gausian for λ_{noise} proved sufficient in preliminary experiments. The sigmoïd parameters a and b [in Eqn. (6.29)] were obtained by observing how q in Eqn. (6.28) decreased as the SNR was lowered on utterances in Session 1 (i.e., training utterances). The resulting value of q_{map} in Eqn. (6.29) was close to one for clean utterances and close to zero for utterances with an SNR of -8 dB. Based on the results presented in Section 6.6.4, single-Gaussian GMMs were used for the modified Bayesian post-classifier. The weights and threshold were found via an exhaustive search procedure.

Results in terms of FA% and FR% in Tables 6.11 through 6.13, and in terms of TE in Figure 6.14.

SNR (dB)	∞	28	24	20	16	12	8	4	0	-4	-8
FA%	3.39	3.39	3.57	3.48	3.39	3.66	3.57	3.21	2.32	1.61	0.98
FR%	3.57	3.57	2.86	2.86	2.86	3.57	5.00	7.14	11.43	25.71	44.29

Table 6.11: Performance of weighted summation fusion using Wark's weight selection

SNR (dB)	∞	28	24	20	16	12	8	4	0	-4	-8
FA%	2.86	2.95	3.12	3.04	2.95	2.86	2.86	3.30	4.38	5.27	5.62
FR%	2.86	3.57	3.57	5.00	5.71	6.43	7.14	7.86	8.57	10.00	10.00

Table 6.12: Performance of weighted summation fusion using proposed weight adjustment

SNR (dB)	∞	28	24	20	16	12	8	4	0	-4	-8
FA%	3.21	3.39	3.57	3.66	3.66	3.57	3.66	4.02	5.80	7.68	7.95
FR%	2.86	3.57	3.57	4.29	4.29	5.71	6.43	6.43	6.43	6.43	6.43

Table 6.13: Performance of modified Bayesian post-classifier using proposed weight adjustment

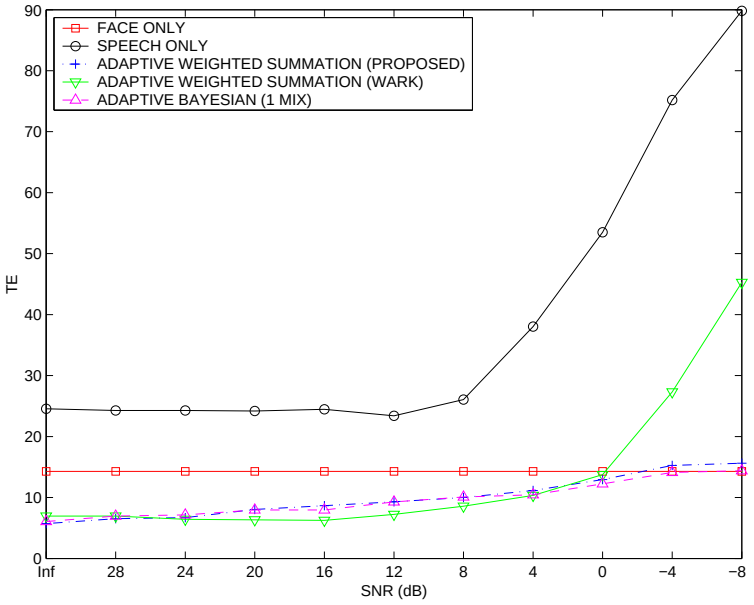

Figure 6.14: Performance of various adaptive fusion approaches

6.7.4 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. (6.10)]. However, the variance of the impostor opinion distribution decreased (as discussed in Section 6.6.5.1) causing the variance of the distances to decrease, while their mean stayed relatively constant. Thus Wark's *a posteriori* confidences (κ) for impostor claims changed relatively little as the SNR was lowered, leading to poor performance. By comparing Tables 6.3 & 6.11 and Figures 6.7 & 6.14 it can be observed that the performance of Wark's approach is similar to the non-adaptive weighted summation approach.

When the proposed weight adjustment method is used in either the weighted summation approach or the modified Bayesian classifier, the performance gently deteriorates as the SNR is lowered, becoming slightly worse than the performance of the face expert (TE=14.28) at a SNR of -8 dB. While the performance is much better than non-adaptive approaches, the proposed approach is noise type dependent (as described in Section 6.7.1).

In terms of TE, the modified Bayesian classifier has very similar performance as the weighted summation approach, at the expense of being significantly more complex due to the use of GMMs to model the true claimant and impostor opinion distributions. However, by comparing Tables 6.12 and 6.13, it can be seen that the performance of the modified Bayesian classifier is closer to EER performance throughout the SNR range than the weighted summation approach.

6.8 Structurally Noise Resistant Post-Classifiers

Inspired by the SRM principle used in SVM (see Section 6.6.3) and by observing the movement of opinion vectors due to presence of noise (see Figures 6.8 and 6.9 and Section 6.6.5.1), a structurally noise resistant piece-wise linear (PL) post-classifier is developed (Section 6.8.1). As the name suggests, the decision surface used by the post-classifier is designed so the contribution of FR errors from the movement of opinion vectors is minimized.

Moreover, the Bayesian classifier presented in Section 6.6.2 is modified to introduce a similar structural constraint (Section 6.8.2). The performance of the two proposed post-classifiers is evaluated in Section 6.8.3.

6.8.1 Piece-Wise Linear Post-Classifier Definition

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

$$g(\vec{o}) = \begin{cases} a(\vec{o}) & \text{if } o_2 \geq o_{2,int} \\ b(\vec{o}) & \text{otherwise} \end{cases} \tag{6.37}$$

where $\vec{o} = [\ o_1\ o_2\ ]^T$ is a 2-dimensional opinion vector,

$$a(\vec{o}) = m_1 o_1 - o_2 + c_1 \tag{6.38}$$

$$b(\vec{o}) = m_2 o_1 - o_2 + c_2 \tag{6.39}$$

and $o_{2,int}$ is the threshold for selecting whether to use $a(\vec{o})$ or $b(\vec{o})$. Figure 6.15 shows an example of the decision surface. The verification decision is reached as follows. The claim is accepted when $g(\vec{o}) \leq 0$ (i.e., true claimant); the claim is rejected when $g(\vec{o}) > 0$ (i.e., impostor).

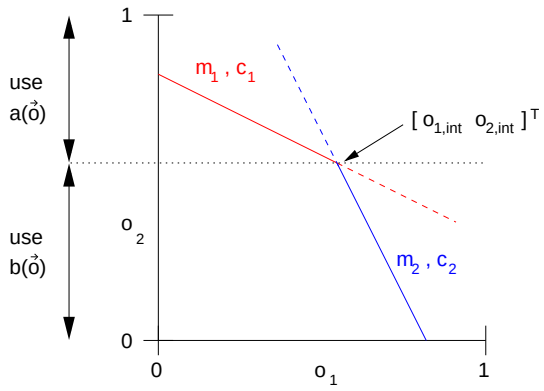

Figure 6.15: Example decision surface of the PL classifier

The first segment of the decision boundary can be described by $a(\vec{o}) = 0$, which reduces Eqn. (6.38) to:

$$0 = m_1 o_1 - o_2 + c_1 \tag{6.40}$$

$$\text{hence} \quad o_2 = m_1 o_1 + c_1 \tag{6.41}$$

If we assume o_2 is a function of o_1, Eqn. (6.41) is simply the description of a line [144], 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 $\vec{o}_{int} = [\ o_{1,int}\ o_{2,int}\ ]^T$; moreover, when the two lines intersect, $a(\vec{o}_{int}) = b(\vec{o}_{int}) = 0$. Hence

$$o_{2,int} = m_1 o_{1,int} + c_1 \tag{6.42}$$

$$\text{and} \quad o_{2,int} = m_2 o_{1,int} + c_2 \tag{6.43}$$

therefore,

$$m_2 o_{1,int} + c_2 = m_1 o_{1,int} + c_1 \tag{6.44}$$

$$m_2 o_{1,int} - m_1 o_{1,int} = c_1 - c_2 \tag{6.45}$$

$$o_{1,int} = \frac{c_1 - c_2}{m_2 - m_1} \tag{6.46}$$

Since $o_{2,int}$ represents the value of o_2 at which the two lines intersect, substituting (6.46) into (6.43) yields:

$$o_{2,int} = m_2 \left(\frac{c_1 - c_2}{m_2 - m_1} \right) + c_2 \tag{6.47}$$

6.8.1.1 Structural Constraints and Training

By observing Figures 6.8 and 6.9 it can be seen that 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 surface which minimizes the increase of FR errors due to this movement. Structurally, this requirement translates to a decision surface that is as steep as possible; moreover, to keep consistency with the experiments done in Sections 6.6 and 6.7, 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 have to be as large as possible.

3. The EER criterion must be satisfied.

Let $\lambda = \{m_1, c_1, m_2, c_2\}$ be the set of PL classifier parameters. Given an initial solution, described in Section 6.8.1.2, the downhill simplex optimization method [96, 104] can be used to find the final parameters. The following function is minimized:

$$\varepsilon(\lambda) = \epsilon_1(\lambda) + \epsilon_2(\lambda) + \epsilon_3(\lambda) \tag{6.48}$$

where $\epsilon_1(\lambda)$ through $\epsilon_3(\lambda)$ (defined below) represent constraints 1-3 described above, respectively.

$$\epsilon_1(\lambda) = \gamma_1 + \gamma_2 \tag{6.49}$$

$$\text{where } \gamma_j = \begin{cases} |o_{j,int}| & \text{if } o_{j,int} < 0 \text{ or } o_{j,int} > 1 \\ 0 & \text{otherwise} \end{cases} \tag{6.50}$$

where $o_{1,int}$ and $o_{2,int}$ are found using Eqns. (6.46) and (6.47), respectively,

$$\epsilon_2(\lambda) = \left|\frac{1}{m_1}\right| + \left|\frac{1}{m_2}\right| \tag{6.51}$$

and finally

$$\epsilon_3(\lambda) = \left|\frac{\text{FA}\%}{100\%} - \frac{\text{FR}\%}{100\%}\right| \tag{6.52}$$

6.8.1.2 Initial Solution of PL Parameters

The initial solution for λ 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.(2.23)], indicating that it can be characterized 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. 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 6.16):

$$P_1 = (\ x_1\ ,\ y_1\ ) = (\ \mu_1\ ,\ \mu_2 + 2\sigma_2\ ) \tag{6.53}$$

$$P_2 = (\ x_2\ ,\ y_2\ ) = \left(\ \mu_1 + 2\sigma_1 \cos\left[\frac{\pi}{4}\right]\ ,\ \mu_2 + 2\sigma_2 \sin\left[\frac{\pi}{4}\right]\ \right) \tag{6.54}$$

$$P_3 = (\ x_3\ ,\ y_3\ ) = (\ \mu_1 + 2\sigma_1\ ,\ \mu_2\ ) \tag{6.55}$$

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

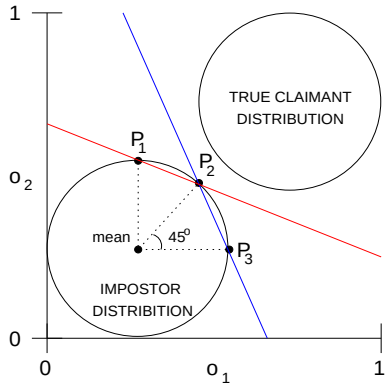

Figure 6.16: Points used in the initial solution of PL classifier parameters

$$m_1 = \frac{y_2 - y_1}{x_2 - x_1} \tag{6.56}$$

$$c_1 = y_1 - m_1 x_1 \tag{6.57}$$

Similarly, the gradient (m_2) and the intercept (c_2) for the second line can be found using:

$$m_2 = \frac{y_3 - y_2}{x_3 - x_2} \tag{6.58}$$

$$c_2 = y_2 - m_2 x_2 \tag{6.59}$$

The initial solution for real data is shown in Figure 6.18.

6.8.2 Modified Bayesian Post-Classifier (Mark II)

In Figure 6.12 it can be seen that the decision boundaries made by the Bayesian post-classifier (described in Section 6.6.2) envelop the true claimant opinion distribution. The downward movement of the vectors due to noisy conditions crosses the boundary and causes the high FR% observed in Tables 6.4 to 6.6. If the decision boundary was forced to envelop the distribution of opinion vectors for impostor claims, the increase in FR% would be reduced. This can be accomplished by modifying Eqn. (6.17) to use only the impostor likelihood:

$$\Lambda(\vec{o}) = -\log \tilde{p}(\vec{o}|C_2) \tag{6.60}$$

i.e., $\log \tilde{p}(\vec{o}|C_1) = 0$, where C_1 and C_2 are the true claimant and impostor classes, respectively.

6.8.3 Performance Evaluation

In this section the performance of the proposed PL and modified Bayesian (Mark II) post-classifiers is evaluated and compared against the performance of the weighted summation approach using fixed and adaptive weights (using the weight update algorithm described in Section 6.7.1). The experimental setup is the same as described in Section 6.7. Results in terms of FA% & FR% are presented in Tables 6.14 to 6.17 and in terms of TE in Figure 6.17. The decision boundaries are shown in Figures 6.18 to 6.21.

SNR (dB)	∞	28	24	20	16	12	8	4	0	-4	-8
FA%	4.29	4.20	4.29	4.29	4.38	4.29	4.38	4.38	4.38	3.75	3.48
FR%	4.29	4.29	5.00	5.00	5.71	6.43	6.43	6.43	7.86	9.29	10.71

Table 6.14: Performance of the PL post-classifier

SNR (dB)	∞	28	24	20	16	12	8	4	0	-4	-8
FA%	4.46	4.20	4.02	4.11	4.11	3.84	3.66	3.48	2.95	2.05	1.88
FR%	4.29	5.00	5.00	5.71	6.43	6.43	7.14	10.71	12.14	21.43	23.57

Table 6.15: Performance of the modified Bayesian post-classifier (Mark II), 1-Gaussian GMM

SNR (dB)	∞	28	24	20	16	12	8	4	0	-4	-8
FA%	4.64	4.73	4.73	4.55	4.29	4.11	3.84	3.39	3.04	2.68	3.21
FR%	4.29	5.00	4.29	4.29	5.71	7.14	8.57	10.00	11.43	16.43	18.57

Table 6.16: Performance of the modified Bayesian post-classifier (Mark II), 2-Gaussian GMM

SNR (dB)	∞	28	24	20	16	12	8	4	0	-4	-8
FA%	4.29	4.29	4.29	4.38	4.02	4.20	3.75	3.30	3.39	2.68	2.05
FR%	4.29	4.29	2.86	5.71	5.71	7.86	10.71	10.71	13.57	16.43	20.71

Table 6.17: Performance of the modified Bayesian post-classifier (Mark II), 3-Gaussian GMM

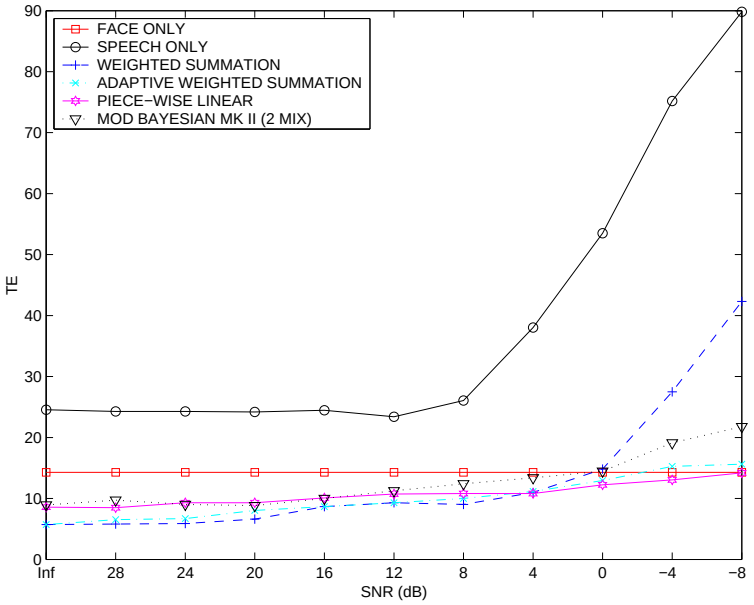

Figure 6.17: Performance of the PL and modified Bayesian (Mark II) post-classifiers compared to fixed and adaptive weighted summation fusion

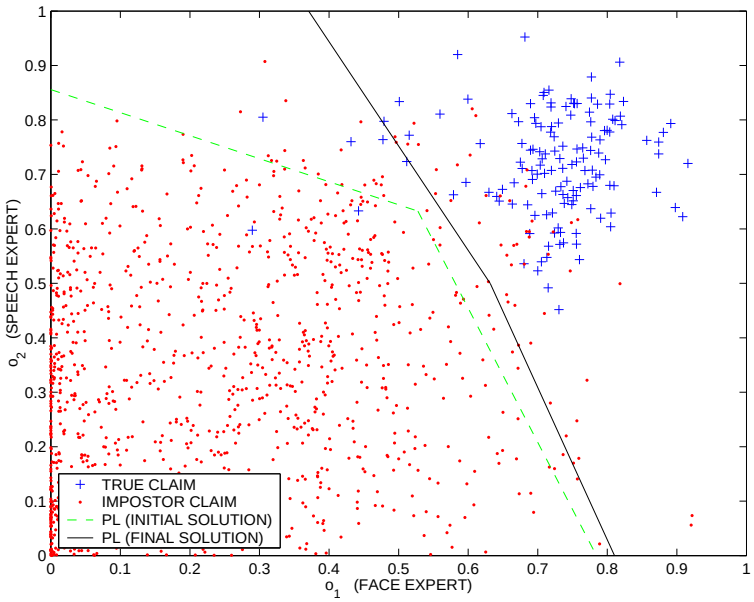

Figure 6.18: Initial and final decision boundaries used by PL post-classifier and distribution of opinion vectors for true & impostor claims using clean speech

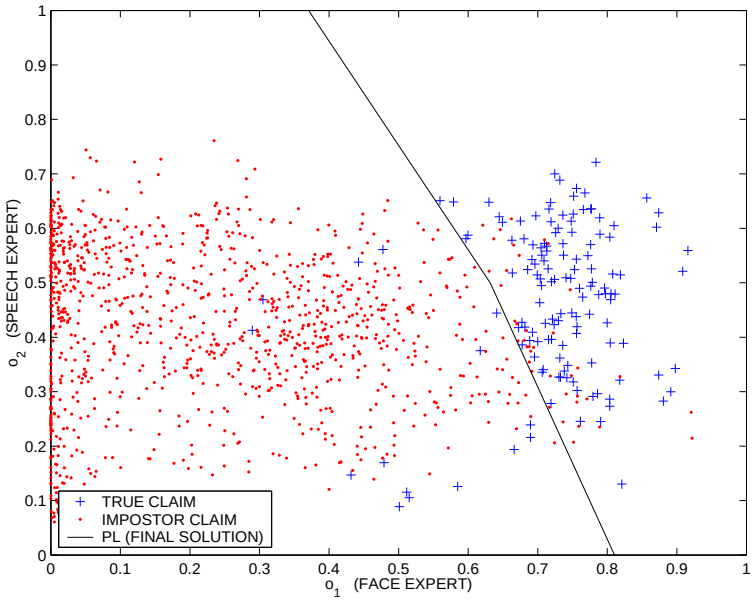

Figure 6.19: Final decision boundaries used by PL post-classifier and distribution of opinion vectors for true & impostor claims using noisy speech

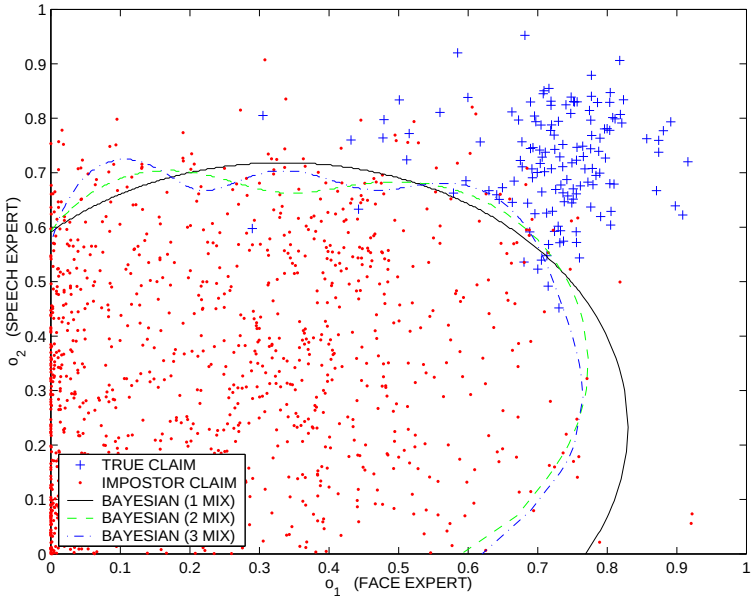

Figure 6.20: Decision boundaries used by modified Bayesian post-classifier (Mark II) and distribution of opinion vectors for true & impostor claims using clean speech

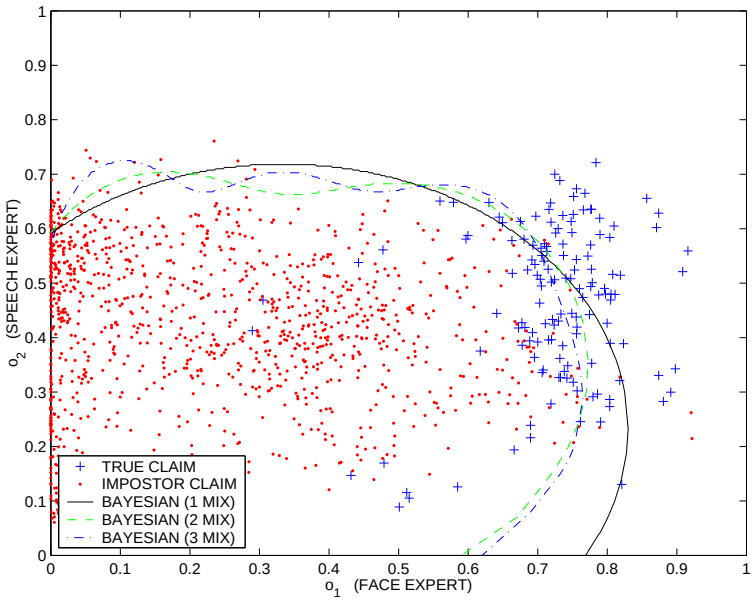

Figure 6.21: As per Figure 6.20, but using noisy speech

6.8.4 Discussion

The decision boundary used by the PL post-classifier effectively takes into account the movement of opinion vectors due to noisy conditions. In clean and low noise conditions the weighted summation fusion (using both fixed and adaptive weights) outperforms the PL post-classifier. However, in high noise conditions (SNR $\leq$ 0) the PL post-classifier obtains better performance than the fixed approach and has similar performance as the adaptive approach, with the advantage of having a fixed (non-adaptive) structure. Moreover, unlike the 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.

For the modified Bayesian post-classifier (Mark II), increasing the number of Gaussians from 1 to 2 results in a decision boundary which envelops the impostor opinion distribution tighter and reduces the number of misclassifications of true claim opinions in noisy conditions. Increasing the number of Gaussians from 2 to 3 has relatively little effect on both the decision boundary and performance in noisy conditions.

Similar to the PL post-classifier, the performance of the modified Bayesian post-classifier is poorer than weighted summation fusion (using both fixed and adaptive weights) in clean and low noise conditions. For SNR $\leq$ -4 dB, it is better than non-adaptive weighted summation, but worse than the PL post-classifier and adaptive weighted summation.

By comparing Tables 6.15 to 6.17 with Tables 6.4 to 6.6, it can be observed that in clean and low noise conditions the performance of the modified Bayesian post-classifier is not as good as the normal Bayesian post-classifier. However, for SNR $\leq$ -4 dB, it is significantly better.

6.9 Chapter Summary

This chapter provided a review of important concepts in the field of *information fusion* as well as a review of previous work on audio-visual person recognition. It has been shown that the weighted summation fusion approach is equivalent to a post-classifier which utilizes a

linear decision surface. The equivalency indicated that for a multi-expert adaptive system it is a fallacy to report the performance in noisy conditions in terms of EER. Evaluation of several standard non-adaptive fusion approaches showed that they result in non-optimal performance in noisy conditions.

Several new methods for combining speech and face information in noisy conditions were proposed, namely: a weight adjustment procedure, which explicitly measures the quality of the speech signal; a modification to the Bayesian post-classifier, allowing the adjustment of the degree of contribution of each expert to the final verification decision; a structurally noise resistant piece-wise linear post-classifier, which attempts to minimize the effects of noisy conditions via structural constraints on the decision boundary; and a modification to the Bayesian post-classifier, which also attempts to impose structural constraints.

Experimental results showed that the proposed weight adjustment procedure outperforms a recently published adaptive approach. Moreover, in noisy conditions, the noise resistant piece-wise linear post-classifier has similar performance to that of the proposed weight adjustment procedure, with the advantage of having a fixed (non-adaptive) structure.

A few words of caution: since there are only four test utterances per person, the size of the test set is rather limited; thus results presented in this chapter may not generalize.

Chapter 7

Conclusions and Further Work

7.1 Chapter Summary and Conclusions

This chapter summarizes the work presented in this thesis and presents the main conclusions that have been drawn from the work; future research is also suggested.

7.1.1 Chapter 2: Gaussian Mixture Model Based Classifier

The chapter began by using Bayesian Decision Theory to derive a decision machine (classifier) used in a verification system. The machine was then implemented using the Gaussian Mixture Model (GMM) approach. The k-means, Expectation Maximization (EM) and maximum a posteriori (MAP) adaptation algorithms, used for finding GMM parameters, were described. Two methods for finding the impostor likelihood were presented: the Background Model Set (BMS) and Universal Background Model (UBM). Next, error measures for finding the performance of a verification system were described. The chapter was concluded by a discussion on implementation issues, where practical limitations and experimental requirements were taken into account.

7.1.2 Chapter 3: Speech Based Verification

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. A recently proposed feature set, termed Maximum Auto-Correlation Values (MACVs), which utilizes 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 database 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 [112]. Further experiments showed that the performance degradation of a verification system used in noisy conditions can be reduced through the use of MACV features.

7.1.3 Chapter 4: VidTIMIT database

The chapter described two previous multi-modal databases: M2VTS and XM2VTS. Their limitations were discussed, such as the size and cost (which was quite prohibitive for XM2VTS). The VidTIMIT database, created by the author while taking into account the problems with M2VTS and XM2VTS databases, was then described.

7.1.4 Chapter 5: Face Based Verification

In this chapter we first reviewed important publications 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; the feature set utilizes 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 changes. Results on the multi-session VidTIMIT database suggest that the proposed feature set is the most robust, followed by (in order of robustness and performance): 2D Gabor wavelets, 2D DCT coefficients and PCA (eigenface) derived features. Moreover, compared to Gabor wavelets, the *DCT-mod2* feature set is over 80 times quicker to compute.

The effects of likelihood normalization in face verification were studied. Current face verification systems use a fixed threshold (or decision surface) to make the final accept or reject decision; this approach does not take into account a mismatch between training and testing conditions, where use of corrupted face images can lead to a false rejection of the claimant. To account for varying image conditions, the decision threshold can be automatically tuned through the use of likelihood normalization. The effectiveness of three likelihood normalization approaches, the Background Model Set (BMS), the Universal Background Model (UBM) and an alternate version of UBM, denoted as UBM-alt, was evaluated. Experiments using face images corrupted by an illumination change, compression artefacts and white Gaussian noise, show that likelihood normalization has little effect when using PCA derived features, while all three normalization approaches provide significant performance improvements when using 2D DCT, 2D Gabor wavelet or DCT-mod2 features. Out of the three, the UBM-alt approach is the most useful, as it provides performance which is close to the best approach (BMS) while having the advantage of being client-independent. The results also show that while PCA derived features are greatly affected by an illumination change, they are quite immune to compression artefacts and white Gaussian noise.

We proposed to solve the fragility of PCA derived features to the illumination direction change 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 then performed. We showed that the *enhanced PCA* technique retains all the positive aspects of traditional PCA, while also being robust to changes in the illumination direction.

Finally, the *DCT-mod2* approach was extended by increasing the number of blocks used in deriving each feature vector; moreover, windowing was introduced, allowing the variation of the contribution of each block. Results show that depending on the window used, the modified feature set is less robust compared to the original feature set when using face images corrupted with an illumination direction change; however, the modified set is more robust to compression artefacts and white Gaussian noise.

7.1.5 Chapter 6: Fusion of Speech and Face Information

This chapter provided a review of important concepts in the field of *information fusion* as well as a review of previous work on audio-visual person recognition. It has been shown that the weighted summation fusion approach is equivalent to a post-classifier which utilizes a linear decision surface. The equivalency indicated that for a multi-expert adaptive system it is a fallacy to report the performance in noisy conditions in terms of EER. Evaluation of several standard non-adaptive fusion approaches showed that they result in non-optimal performance in noisy conditions.

Several new methods for combining speech and face information in noisy conditions were proposed, namely: a weight adjustment procedure, which explicitly measures the quality of the speech signal; a modification to the Bayesian post-classifier, allowing the adjustment of the degree of contribution of each expert to the final verification decision; a structurally noise resistant piece-wise linear post-classifier, which attempts to minimize the effects of noisy conditions via structural constraints on the decision boundary; and a modification to the Bayesian post-classifier, which also attempts to impose structural constraints.

Experimental results showed that the proposed weight adjustment procedure outperforms a recently published adaptive approach. Moreover, in noisy conditions, the noise resistant piece-wise linear post-classifier has similar performance to that of the proposed weight adjustment procedure, with the advantage of having a fixed (non-adaptive) structure.

7.2 Suggested Future Research

In this thesis we have presented a number of approaches aimed at increasing the robustness of speech based, face based and multi-modal (speech and face) verification systems. In keeping with this line of research, this section presents possible avenues for further study.

In Chapter 3 experiments on the NTIMIT database have shown that the Maximum Auto-Correlation Value (MACV) feature set reduces the effects of noisy conditions; it would be interesting to see if the results extend to the Switchboard database [33].

In speech recognition systems, it has been recently been shown that Spectral Subband Centroid (SSC) features [40, 98] and biologically inspired Zero-Crossing with Peak Amplitude (ZCPA) features [65] are quite robust to the effects of additive noise. While the speaker verification task is significantly different from the speech recognition task, the SSC and ZCPA features may still contain person-dependent information; thus it would be interesting to evaluate their usefulness for robust person verification purposes.

The *DCT-mod2* facial feature extraction approach proposed in Section 5.3 can be extended to utilize diagonally neighbouring blocks, possibly resulting in more robustness to non-linear illumination changes and white Gaussian noise; moreover, a more thorough analysis of which 2D DCT basis functions are the most useful for face verification could lead to a reduction in the dimensionality of *DCT-mod2* feature vectors.

The structurally noise resistant piece-wise linear post-classifier presented in Section 6.8.1 could be extended to handle more than two modality experts; its decision surface can be modified to account for more than one modality expert being affected by environmental conditions; moreover, rather than using the general downhill simplex optimization method [96, 104], a dedicated optimization algorithm could be developed.

It can be reasonably expected that the performance of a face verification system (or expert), trained on face images without eye glasses, would decrease if test face images contained eye glasses. To increase robustness, the multi-modal system can be extended by adding lip region expert; the system can be made even more robust by detecting that a test face image contains eye glasses and decreasing the contribution of the face expert accordingly.

Appendix A

Experiments on the Weizmann Database

The experiments described in Section 5.3.2 utilize an artificial illumination direction change. In this appendix we shall compare the performance of 2D DCT, 2D Gabor wavelet and *DCT-mod2* feature sets (see Section 5.3) on the Weizmann Database [3], which has more realistic illumination direction changes.

It must be noted that the database 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 database is not suited for verification experiments, but some suggestive results can still be obtained.

The experimental setup is similar to that described in Section 5.3.2. 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) by means of maximum *a posteriori* (MAP) adaptation [43, 115]. 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 database, 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 2.3.2.2.

Since PCA based feature extraction produces one feature vector per image (see Section

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 A.1: Results on the Weizmann Database, quoted in terms of approximate EER (%).

5.3.1.1), there is insufficient training data to reliably train the client models. Thus PCA based feature extraction is not evaluated in this appendix.

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 *a posteriori* decision threshold was set to obtain performance as close as possible to EER. Results are presented in Table A.1.

As can be observed, no method is immune to the changes in the illumination direction. However *DCT-mod2* features are the least affected, followed by Gabor features and lastly DCT features.

Appendix B

Face Areas Modeled by the GMM

A typical example of the face areas modeled by each Gaussian (in an 8-Gaussian GMM) is shown in Figure B.1, 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 center blocks used in *DCT-mod2* feature extraction (see Section 5.3.1.5). 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 modeled by each Gaussian is generally guided by the degree of smoothness of the area; this leads 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 2.3.1) is in effect a probabilistic clustering procedure, where similar features are represented by each Gaussian.

Figure B.2 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 B.1 and B.2 it can be seen that *fdrd1*'s model selects similar areas in *fdrd1*'s and *mbdg0*'s face images. Thus 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.

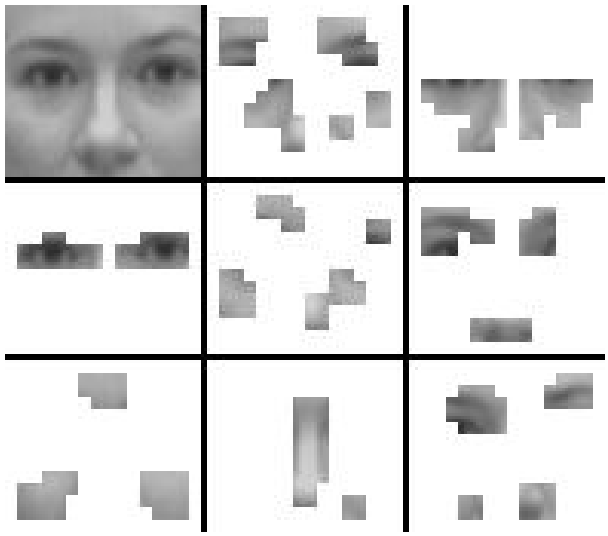

Figure B.1: Typical example of 8-Gaussian GMM face modeling. Top left: original image of subject *fdrd1*; other squares: areas modeled by each Gaussian in *fdrd1*'s model (*DCT-mod2* feature extraction).

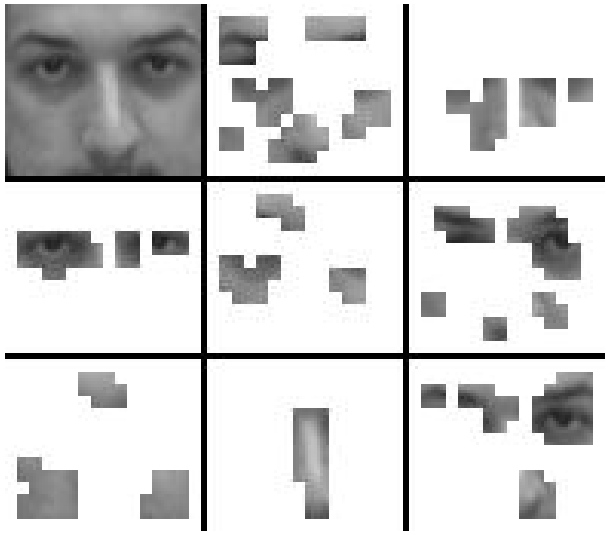

Figure B.2: Top left: original image of subject *mbdg0*; other squares: areas selected by *fdrd1*'s Gaussians.

Appendix C

EM Algorithm for Gaussian Mixture Models

In the Gaussian Mixture Model (GMM) approach, a D-dimensional observation vector $\vec{x}$ is modelled by:

$$p(\vec{x}|\Theta) = \sum_{m=1}^{M} w_m p(\vec{x}|\theta_m) \tag{C.1}$$

where $\sum_{m=1}^{M} w_m = 1$, $w_m \geq 0$ and $p(\vec{x}|\theta_m)$ is a multivariate Gaussian probability density function with parameter set $\theta_m = \{\vec{\mu}_m, \Sigma_m\}$:

$$p(\vec{x}|\theta_m) = \mathcal{N}(\vec{x}; \vec{\mu}_m, \Sigma_m) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_m|^{\frac{1}{2}}} \exp\left[\frac{-1}{2}(\vec{x} - \vec{\mu}_m)^T \Sigma_m^{-1} (\vec{x} - \vec{\mu}_m)\right] \tag{C.2}$$

where $\vec{\mu}_m$ is the mean vector and Σ_m is the covariance matrix. Thus the complete parameter set for Eqn. (C.1) is expressed as $\Theta = \{w_m, \theta_m\}_{m=1}^{M}$. Our aim is to find Θ so the likelihood function

$$p(X|\Theta) = \prod_{i=1}^{N} p(\vec{x}_i|\Theta) \tag{C.3}$$

is maximized. Here, $X = \{\vec{x}_i\}_{i=1}^{N}$ is the set of training data.

The Expectation-Maximization (EM) algorithm [31, 109] is a likelihood function optimization technique, often used in the pattern recognition literature [35]. 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" $\vec{x}_i$. Thus $y_i \in [1, M]\ \forall\ i$ and $y_i = m$ if the i-th feature vector ($\vec{x}_i$) was "generated" by the m-th mixture component. If we know the values for Y, then Eqn. (C.3) can be modified to:

$$p(X, Y|\Theta) = \prod_{i=1}^{N} w_{y_i} p(\vec{x}_i|\theta_{y_i}) \tag{C.4}$$

As its name suggests, the EM algorithm is comprised of two steps: expectation, followed by maximization. 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 $X = \{\vec{x}_i\}_{i=1}^N$ and current parameter estimates, $\Theta^{[k]}$ (where k indicates the iteration number):

$$Q(\Theta, \Theta^{[k]}) = E\left[\log p(X, Y|\Theta)\ |\ X, \Theta^{[k]}\right] \tag{C.5}$$

Since Y is a random variable with distribution $p(\boldsymbol{y}|X, \Theta^{[k]})$, Eqn. (C.5) can be written as:

$$Q(\Theta, \Theta^{[k]}) = \int_{\boldsymbol{y}\in\Upsilon} \log p(X, \boldsymbol{y}|\Theta)\ p(\boldsymbol{y}|X, \Theta^{[k]})\ \ d\boldsymbol{y} \tag{C.6}$$

where $\boldsymbol{y}$ is an instance of the missing data and Υ is the space of values $\boldsymbol{y}$ can take on. The maximization step then maximizes the expectation:

$$\Theta^{[k+1]} = \arg\max_{\Theta} Q(\Theta, \Theta^{[k]}) \tag{C.7}$$

The expectation and maximization steps are iterated until convergence, or when the increase in likelihood falls below a pre-defined threshold. As can be seen in Eqn. (C.6), we require $p(\boldsymbol{y}|X, \Theta^{[k]})$. We can define it as follows:

$$p(\boldsymbol{y}|X, \Theta^{[k]}) = \prod_{i=1}^{N} p(y_i|\vec{x}_i, \Theta^{[k]}) \tag{C.8}$$

Given initial parameters[1] $\Theta^{[k]}$, we can compute $p(\vec{x}_i|\theta_m^{[k]})$. Moreover, we can interpret the mixing weights (w_m) as *a priori* probabilities of each mixture component

[1]Parameters for $k = 0$ can be found via the k-means algorithm [35, 76] (see also Section 2.3.1.1).

[i.e., $w_m = p(m|\Theta^{[k]})$]. Hence we can apply Bayes' rule [35] to obtain:

$$p(y_i|\vec{x}_i, \Theta^{[k]}) = \frac{p(\vec{x}_i|\theta_{y_i}^{[k]})p(y_i|\Theta^{[k]})}{p(\vec{x}_i|\Theta^{[k]})} \quad (C.9)$$

$$= \frac{p(\vec{x}_i|\theta_{y_i}^{[k]})p(y_i|\Theta^{[k]})}{\sum_{n=1}^{M} p(\vec{x}_i|\theta_n^{[k]})p(n|\Theta^{[k]})} \quad (C.10)$$

Expanding Eqn. (C.6) yields:

$$Q(\Theta, \Theta^{[k]}) = \int_{\boldsymbol{y}\in\Upsilon} \log p(X, \boldsymbol{y}|\Theta)\ p(\boldsymbol{y}|X, \Theta^{[k]})\ d\boldsymbol{y} \quad (C.11)$$

$$= \sum_{\boldsymbol{y}\in\Upsilon} \log \prod_{i=1}^{N} w_{y_i} p(\vec{x}_i|\theta_{y_i}) \prod_{j=1}^{N} p(y_j|\vec{x}_j, \Theta^{[k]}) \quad (C.12)$$

$$= \sum_{y_1=1}^{M}\sum_{y_2=1}^{M}\cdots\sum_{y_N=1}^{M}\sum_{i=1}^{N} \log\left[w_{y_i} p(\vec{x}_i|\theta_{y_i})\right] \prod_{j=1}^{N} p(y_j|\vec{x}_j, \Theta^{[k]}) \quad (C.13)$$

It can be shown [18] that Eqn. (C.13) can be simplified to:

$$Q(\Theta, \Theta^{[k]}) = \sum_{m=1}^{M}\sum_{i=1}^{N} \log[w_m p(\vec{x}_i|\theta_m)]\ p(m|\vec{x}_i, \Theta^{[k]}) \quad (C.14)$$

$$= \sum_{m=1}^{M}\sum_{i=1}^{N} \log[w_m]\ p(m|\vec{x}_i, \Theta^{[k]}) + \sum_{m=1}^{M}\sum_{i=1}^{N} \log[p(\vec{x}_i|\theta_m)]\ p(m|\vec{x}_i, \Theta^{[k]}) \quad (C.15)$$

$$= Q_1 + Q_2 \quad (C.16)$$

Hence Q_1 and Q_2 can be maximized separately, to obtain w_m and $\theta_m = \{\vec{\mu}_m, \Sigma_m\}$, respectively. To find the expression which maximizes w_m, we need to introduce the Lagrange multiplier [35] λ, 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:

$$\frac{\partial Q_1}{\partial w_m} = 0 \quad (C.17)$$

$$\therefore\ 0 = \frac{\partial}{\partial w_m}\left\{\sum_{m=1}^{M}\sum_{i=1}^{N} \log[w_m]\ p(m|\vec{x}_i, \Theta^{[k]}) + \lambda\left[(\sum_m w_m) - 1\right]\right\} \quad (C.18)$$

$$= \sum_{i=1}^{N} \frac{1}{w_m}\ p(m|\vec{x}_i, \Theta^{[k]}) + \lambda \quad (C.19)$$

Let us rearrange Eqn. (C.19) so we can obtain a value for λ:

$$-\lambda w_m = \sum_{i=1}^{N} p(m|\vec{x}_i, \Theta^{[k]}) \quad (C.20)$$

Summing both sides over m yields:

$$-\lambda \sum_m w_m = \sum_{i=1}^{N} \sum_m p(m|\vec{x}_i, \Theta^{[k]}) \tag{C.21}$$

$$-\lambda 1 = \sum_{i=1}^{N} 1 \tag{C.22}$$

$$\lambda = -N \tag{C.23}$$

By substituting Eqn. (C.23) into Eqn. (C.19) we obtain:

$$N = \sum_{i=1}^{N} \frac{1}{w_m}\, p(m|\vec{x}_i, \Theta^{[k]}) \tag{C.24}$$

$$\therefore\ w_m = \frac{1}{N} \sum_{i=1}^{N} p(m|\vec{x}_i, \Theta^{[k]}) \tag{C.25}$$

To find expressions which maximize $\vec{\mu}_m$ and Σ_m, let us now expand Q_2:

$$Q_2 = \sum_{m=1}^{M} \sum_{i=1}^{N} \log[p(\vec{x}_i|\theta_m)]\ p(m|\vec{x}_i, \Theta^{[k]}) \tag{C.26}$$

$$= \sum_{m=1}^{M} \sum_{i=1}^{N} \left[-\frac{1}{2} \log(|\Sigma_m|) - \frac{1}{2} (\vec{x}_i - \vec{\mu}_m)^T \Sigma_m^{-1} (\vec{x}_i - \vec{\mu}_m) \right] p(m|\vec{x}_i, \Theta^{[k]}) \tag{C.27}$$

where $-\frac{D}{2}\log(2\pi)$ was omitted since it vanishes when taking a derivative with respect to $\vec{\mu}_m$ or Σ_m^{-1}. To find the expression which maximizes μ_m, we need to take the derivative of Q_2 with respect to $\vec{\mu}_m$, and set the result to zero:

$$\frac{\partial Q_2}{\partial \vec{\mu}_m} = 0 \tag{C.28}$$

$$0 = \frac{\partial}{\partial \vec{\mu}_m} \left\{ \sum_{m=1}^{M} \sum_{i=1}^{N} \left[-\frac{1}{2} \log(|\Sigma_m|) - \frac{1}{2} (\vec{x}_i - \vec{\mu}_m)^T \Sigma_m^{-1} (\vec{x}_i - \vec{\mu}_m) \right] p(m|\vec{x}_i, \Theta^{[k]}) \right\} \tag{C.29}$$

At this point let us recall some results from matrix theory. Lütkepohl [80] states that $\frac{\partial \vec{z}^T A \vec{z}}{\partial \vec{z}} = (A + A^T)\vec{z}$, $(A^{-1})^T = (A^T)^{-1}$ and if A is symmetric, then $A = A^T$. Since Σ_m is symmetric, Eqn. (C.29) reduces to:

$$0 = \sum_{i=1}^{N} -\frac{1}{2} 2 \Sigma_m^{-1} (\vec{x}_i - \vec{\mu}_m) p(m|\vec{x}_i, \Theta^{[k]}) \tag{C.30}$$

$$= \sum_{i=1}^{N} \left[-\Sigma_m^{-1} \vec{x}_i p(m|\vec{x}_i, \Theta^{[k]}) + \Sigma_m^{-1} \vec{\mu}_m p(m|\vec{x}_i, \Theta^{[k]}) \right] \tag{C.31}$$

$$\therefore\ \sum_{i=1}^{N} \Sigma_m^{-1} \vec{\mu}_m p(m|\vec{x}_i, \Theta^{[k]}) = \sum_{i=1}^{N} \Sigma_m^{-1} \vec{x}_i p(m|\vec{x}_i, \Theta^{[k]}) \tag{C.32}$$

multiply both sides by Σ_m:

$$\sum_{i=1}^{N} \vec{\mu}_m p(m|\vec{x}_i, \Theta^{[k]}) = \sum_{i=1}^{N} \vec{x}_i p(m|\vec{x}_i, \Theta^{[k]}) \quad \text{(C.33)}$$

$$\therefore \ \vec{\mu}_m = \frac{\sum_{i=1}^{N} \vec{x}_i p(m|\vec{x}_i, \Theta^{[k]})}{\sum_{i=1}^{N} p(m|\vec{x}_i, \Theta^{[k]})} \quad \text{(C.34)}$$

Lütkepohl [80] states that: $|A^{-1}| = |A|^{-1}$ and $\mathrm{tr}(AB) = \mathrm{tr}(BA)$. Since $\mathrm{tr}(\vec{z}A\vec{z}^T) = \mathrm{tr}(\text{scalar})$, we can rewrite Eqn. (C.27) as:

$$Q_2 = \sum_{m=1}^{M} \sum_{i=1}^{N} \left[\frac{1}{2} \log(|\Sigma_m^{-1}|) - \frac{1}{2} \mathrm{tr}(\Sigma_m^{-1}(\vec{x}_i - \vec{\mu}_m)(\vec{x}_i - \vec{\mu}_m)^T) \right] p(m|\vec{x}_i, \Theta^{[k]}) \quad \text{(C.35)}$$

Let us quote some more results from Lütkepohl [80]: $\frac{\partial \log(|A|)}{\partial A} = (A^T)^{-1}$ and $\frac{\partial \mathrm{tr}(BA)}{\partial B} = A^T$. Moreover, we note that $\vec{z}\vec{z}^T$ is a symmetric matrix. To find an expression which maximizes Σ_m, we can take the derivative of Eqn. (C.35) with respect to Σ_m^{-1} and set the result to zero:

$$0 = \frac{\partial Q_2}{\partial \Sigma_m^{-1}} \quad \text{(C.36)}$$

$$= \frac{\partial}{\partial \Sigma_m^{-1}} \left\{ \sum_{m=1}^{M} \sum_{i=1}^{N} \left[\frac{1}{2} \log(|\Sigma_m^{-1}|) - \frac{1}{2} \mathrm{tr}\left(\Sigma_m^{-1}(\vec{x}_i - \vec{\mu}_m)(\vec{x}_i - \vec{\mu}_m)^T\right) \right] p(m|\vec{x}_i, \Theta^{[k]}) \right\} \quad \text{(C.37)}$$

$$= \sum_{i=1}^{N} \left[\frac{1}{2}\Sigma_m - \frac{1}{2}(\vec{x}_i - \vec{\mu}_m)(\vec{x}_i - \vec{\mu}_m)^T \right] p(m|\vec{x}_i, \Theta^{[k]}) \quad \text{(C.38)}$$

$$\quad \text{(C.39)}$$

thus

$$\frac{1}{2}\Sigma_m \sum_{i=1}^{N} p(m|\vec{x}_i, \Theta^{[k]}) = \frac{1}{2} \sum_{i=1}^{N} (\vec{x}_i - \vec{\mu}_m)(\vec{x}_i - \vec{\mu}_m)^T p(m|\vec{x}_i, \Theta^{[k]}) \quad \text{(C.40)}$$

$$\therefore \ \Sigma_m = \frac{\sum_{i=1}^{N} (\vec{x}_i - \vec{\mu}_m)(\vec{x}_i - \vec{\mu}_m)^T p(m|\vec{x}_i, \Theta^{[k]})}{\sum_{i=1}^{N} p(m|\vec{x}_i, \Theta^{[k]})} \quad \text{(C.41)}$$

In summary,

$$w_m^{[k+1]} = \frac{1}{N} \sum_{i=1}^{N} p(m|\vec{x}_i, \Theta^{[k]}) \quad \text{(C.42)}$$

$$\vec{\mu}_m^{[k+1]} = \frac{\sum_{i=1}^{N} \vec{x}_i \ p(m|\vec{x}_i, \Theta^{[k]})}{\sum_{i=1}^{N} p(m|\vec{x}_i, \Theta^{[k]})} \quad \text{(C.43)}$$

$$\Sigma_m^{[k+1]} = \frac{\sum_{i=1}^{N} (\vec{x}_i - \vec{\mu}_m^{[k+1]})(\vec{x}_i - \vec{\mu}_m^{[k+1]})^T p(m|\vec{x}_i, \Theta^{[k]})}{\sum_{i=1}^{N} p(m|\vec{x}_i, \Theta^{[k]})} \quad \text{(C.44)}$$

where

$$p(m|\vec{x}_i, \Theta^{[k]}) = \frac{p(\vec{x}_i|\theta_m^{[k]})p(m|\Theta^{[k]})}{\sum_{n=1}^{M} p(\vec{x}_i|\theta_n^{[k]})p(n|\Theta^{[k]})} \tag{C.45}$$

which can be explicitly stated as:

$$p(m|\vec{x}_i, \Theta^{[k]}) = \frac{\mathcal{N}(\vec{x}_i; \vec{\mu}_m^{[k]}, \Sigma_m^{[k]})w_m^{[k]}}{\sum_{n=1}^{M} \mathcal{N}(\vec{x}_i; \vec{\mu}_n^{[k]}, \Sigma_n^{[k]})w_n^{[k]}} \tag{C.46}$$

If we let $l_{m,i} = p(m|\vec{x}_i, \Theta^{[k]})$ and $L_m = \sum_{i=1}^{N} l_{m,i}$, we can restate Eqns. (C.42) through (C.44) as:

$$w_m^{[k+1]} = \frac{L_m}{N} \tag{C.47}$$

$$\vec{\mu}_m^{[k+1]} = \frac{1}{L_m} \sum_{i=1}^{N} \vec{x}_i \; l_{m,i} \tag{C.48}$$

$$\Sigma_m^{[k+1]} = \frac{1}{L_m} \sum_{i=1}^{N} (\vec{x}_i - \vec{\mu}_m^{[k+1]})(\vec{x}_i - \vec{\mu}_m^{[k+1]})^T l_{m,i} \tag{C.49}$$

Bibliography

[1] Y. Abdeljaoued, "Fusion of Person Authentication Probabilities by Bayesian Statistics", *Proc. Second International Conf. on Audio- and Video-based Biometric Person Authentication*, Washington D.C., 1999, pp. 172-175.

[2] B. Achermann and 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 and S. Ullman, "Face Recognition: The Problem of Compensating for Changes in Illumination Direction", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 19, 1997, pp. 721-732.

[4] A. Adjoudani and C. Benoît, "Audio-Visual Speech Recognition Compared Across Two Architectures", *Proc. 4th European Conf. Speech Communication and Technology*, Madrid, Spain, 1995, Vol. 2, pp. 1563-1567.

[5] L. A. Alexandre, A. C. Campilho and M. Kamel, "On combining classifiers using sum and product rules", *Pattern Recognition Letters*, Vol. 22, 2001, pp. 1283-1289.

[6] H. Altiçay and M. Demirekler, "An information theoretic framework for weight estimation in the combination of probabilistic classifiers for speaker identification", *Speech Communication*, Vol. 30, 2000, pp. 255-272.

[7] H. Altiçay and M. Demirekler, "Comparison of Different Objective Functions for Optimal Linear Combination of Classifiers for Speaker Identification", *Proc. IEEE*

International Conf. Acoustics, Speech and Signal Processing, Salt Lake City, 2001, Vol. 1, pp. 401-404.

[8] R. Auckenthaler, M. Carey and H. Lloyd-Thomas, "Score Normalization for Text-Independent Speaker Verification Systems", *Digital Signal Processing*, Vol. 10, 2000, pp. 42-54.

[9] B. S. Atal, *Automatic Speaker Recognition Based on Pitch Contours*, PhD Thesis, Polytechnic Institute of Brooklyn, 1968.

[10] B. S. Atal and M. R. Schroeder, "Predictive Coding of Speech Signals", *Report of the 6th International Congress on Acoustics*, Tokyo, 1968.

[11] 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.

[12] W. Atkins, "A testing time for face recognition technology", *Biometric Technology Today*, Vol. 9, No. 3, 2001, pp. 8-11.

[13] R. Balchandran, V. Ramanujam and R. Mammone, "Channel Estimation and Normalization by Coherent Spectral Averaging For Robust Speaker Verification", *Proc. 6th European Conf. Speech Communication and Technology*, Budapest, 1999, pp. 755-758.

[14] Y. Barniv and 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] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, 1997, pp. 711-720.

[17] S. Ben-Yacoub, Y. Abdeljaoued and E. Mayoraz, "Fusion of Face and Speech Data for Person Identity Verification", *IEEE Trans. on Neural Networks*, Vol. 10, No. 5, 1999, pp. 1065-1074.

[18] 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.

[19] R. Brunelli and T. Poggio, "Face Recognition: Features versus Templates", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 15, No. 10, 1993, pp. 1042-1052.

[20] R. Brunelli, D. Falavigna, T. Poggio and L. Stringa, "Automatic Person Recognition Using Acoustic and Geometric Features", *Machine Vision & Applications*, Vol. 8, 1995, pp. 317-325.

[21] R. Brunelli and D. Falavigna, "Person Identification Using Multiple Cues", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 17, 1995, pp. 955-965.

[22] M. M. Buechner, "Eye of the Beholder", *Time Australia*, 27 November 2000 (No. 47), pp. 89-92.

[23] 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.

[24] K. R. Castleman, *Digital Image Processing*, Prentice-Hall, USA, 1996.

[25] C. Champod and D. Meuwly, "The inference of identity in forensic speaker recognition", *Speech Communication*, Vol. 31, No. 2-3, 2000, pp. 193-203.

[26] 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.

[27] L-F. Chen, H-Y. Liao, J-C. Lin and 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.

[28] C. C. Chibelushi, F. Deravi and J. S. Mason, "Voice and Facial Image Integration for Speaker Recognition", *IEEE International Symposium and Multimedia Technologies and Future Applications*, Southampton, UK, 1993.

[29] E. Caulcott, *Significance tests*, Routledge & Kegan Paul, London, 1973.

[30] J. Daugman, "The importance of being random: statistical principles of iris recognition", *Pattern Recognition* (in publication).

[31] A.P. Dempster, N.M. Laird and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *J. Royal Statistical Soc., Ser. B*, Vol. 39, No. 1, 1977, pp. 1-38.

[32] U. Dieckmann, P. Plankensteiner and T. Wagner, "SESAM: A biometric person identification system using sensor fusion", *Pattern Recognition Letters*, Vol. 18, No. 9, 1997, pp. 827-833.

[33] G. R. Doddington, M. A. Przybycki, A. F. Martin and D. A. Reynolds, "The NIST speaker recognition evaluation - Overview, methodology, systems, results, perspective", *Speech Communication*, Vol. 31, No. 2-3, 2000, pp. 225-254.

[34] B. Duc, S. Fischer and J. Bigün, "Face Authentication with Gabor Information on Deformable Graphs", *IEEE Trans. Image Processing*, Vol. 8, No. 4, 1999, pp. 504-516.

[35] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, John Wiley & Sons, USA, 2001.

[36] S. Eickeler, S. Müller and G. Rigoll, "Recognition of JPEG Compressed Face Images Based on Statistical Methods", *Image and Vision Computing*, Vol. 18, No. 4, 2000, pp. 279-287.

[37] K. R. Farrell, "Text-Dependent Speaker Verification Using Data Fusion", *Proc. IEEE International Conf. Acoustics, Speech and Signal Processing*, Detroit, Michigan, 1995, Vol. 1, pp. 349-352.

[38] S. Furui, "Cepstral Analysis Technique for Automatic Speaker Verification", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. 29, No. 2, 1981, pp. 254-272.

[39] S. Furui, "Recent Advances in Speaker Recognition", *Pattern Recognition Letters*, Vol. 18, No. 9, 1997, pp. 859-872.

[40] B. Gajic and K. K. Paliwal, "Robust Feature Extraction Using Subband Spectral Centroid Histograms", *Proc. International Conf. on Acoustics, Speech and Signal Processing*, Salt Lake City, 2001, pp. 85-88.

[41] R. C. Gonzales and R. E. Woods, *Digital Image Processing*, Addison-Wesley, Reading, Massachusetts, 1993.

[42] J-L. Gauvain and C-H. Lee, "Bayesian learning for hidden Markov model with Gaussian Mixture state observation densities", *Speech Communication*, Vol. 11, 1992, pp. 205-213.

[43] J-L. Gauvain and C-H. Lee, "Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains", *Proc. IEEE Trans. Speech and Audio Processing*, Vol. 2, No. 2, 1994. pp. 291-298.

[44] D. Genoud, F. Bimbot, G. Gravier and G. Chollet, "Combining methods to improve speaker verification", *Proc. 4th International Conf. Spoken Language Processing*, Philadelphia, 1996, Vol. 3, pp. 1756-1759.

[45] H. Gish and M. Schmidt, "Text-independent speaker identification", *IEEE Signal Processing Magazine*, Vol. 11, No. 4, 1994, pp. 18-32.

[46] M. A. Grudin, "On internal representations in face recognition systems", *Pattern Recognition*, Vol. 33, No. 7, 2000, pp. 1161-1177.

[47] J. A. Haigh and J. S. Mason, "A voice activity detector based on cepstral analysis", *Proc. European Conf. Speech Communication and Technology*, 1993, Vol. 2, pp. 1103-1106.

[48] J. A. Haigh, "Voice Activity Detection for Conversational Analysis", Masters Thesis, University of Wales, 1994.

[49] D. L. Hall and J. Llinas, "Multisensor Data Fusion" in: *Handbook of Multisensor Data Fusion* (editors: D. L. Hall and J. Llinas), CRC Press, USA, 2001, pp. 1-1 - 1-10.

[50] C-C. Han, H-Y. M. Liao, G-J. Yu and L-H. Chen, "Fast face detection via morphology-based pre-processing", *Pattern Recognition*, Vol. 33, No. 10, 2000, pp. 1701-1712.

[51] *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*, Cambridge University Press, 1999.

[52] H. Hermansky, N. Morgan, A. Bayya and P. Kohn, "RASTA-PLP Speech Analysis Technique", *Proc. IEEE International Conf. Acoustics, Speech and Signal Processing*, San Francisco, 1992, Vol. 1, pp. 121-124.

[53] H. Hermansky and N. Morgan, "RASTA Processing of Speech", *IEEE Trans. Speech and Audio Processing*, Vol. 2, No. 4, 1994, pp. 578-589.

[54] T. K. Ho, J. J. Hull and S. N. Srihari, "Decision Combination in Multiple Classifier Systems", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 16, No. 1, 1994, pp. 66-75.

[55] L. Hong and A. Jain, "Integrating Faces and Fingerprints for Personal Identification", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 20, No. 12, 1998, pp. 1295-1306.

[56] X. Huang, A. Acero and H-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*, Prentice Hall PTR, New Jersey, 2001.

[57] Q. Huo, C. Chan, C-H. Lee, "Bayesian Adaptive Learning of the Parameters of Hidden Markov Model for Speech Recognition", *Proc. IEEE Trans. Speech and Audio Processing*, Vol. 3, No. 5, 1995, pp. 334-345.

[58] S. S. Iyengar, L. Prasad and H. Min, *Advances in Distributed Sensor Technology*, Prentice Hall PTR, New Jersey, 1995.

[59] T. Joachims, "Making Large-Scale SVM Learning Practical" in: *Advances in Kernel Methods - Support Vector Learning* (editors: B. Schölkopf, C. Burges and A. Smola), MIT-Press, 1999.

[60] N. L. Johnson and F. C. Leone, *Statistics and Experimental Design in Engineering and the Physical Sciences* (Vol. 1), John Wiley & Sons, USA, 1977.

[61] C. Jankowski, A. Kalyanswamy, S. Basson and J. Spitz, "NTIMIT: A Phonetically Balanced, Continuous Speech Telephone Bandwidth Speech Database", *Proc. International Conf. Acoustics, Speech and Signal Processing*, Albuquerque, 1990, Vol. 1, pp. 109-112.

[62] K. Jonsson, J. Kittler, Y.P. Li and J. Matas, "Support Vector Machines for Face Authentication", *Image and Vision Computing*, Vol. 20, No. 5-6, 2002, pp. 369-375.

[63] P. Jourlin, J. Luettin, D. Genoud and H. Wassner, "Acoustic-labial speaker verification", *Pattern Recognition Letters*, Vol. 18, No. 9, 1997, pp. 853-858.

[64] P. Jourlin, J. Luettin, D. Genoud and H. Wassner, "Integrating Acoustic and Labial Information for Speaker Indentification and Verification", *Proc. 5th European Conf. Speech Communication and Technology*, Rhodes, Greece, 1997, Vol. 3, pp. 1603-1606.

[65] D-S. Kim, S-Y. Lee and R. M. Kil, "Auditory Processing of Speech Signals for Robust Speech Recognition in Real-World Noisy Environments", *IEEE Trans. Speech and Audio Processing*, Vol. 7, No. 1, 1999, pp. 55-69.

[66] M. Kirby and L. Sirovich, "Application of the Karhunen-Loève Procedure for the Characterization of Human Faces", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 12, No. 1, 1990, pp. 103-108.

[67] J. Kittler, J. Matas, K. Johnsson and M. U. Ramos Sánchez, "Combining evidence in personal identity verification systems", *Pattern Recognition Letters*, Vol. 18, No. 9, 1997, pp. 845-852.

[68] J. Kittler, M. Hatef, R. P. W. Duin and J. Matas, "On Combining Classifiers", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 20, No. 3, 1998, pp. 226-239.

[69] L. H. Koh, S. Ranganath and Y. V. Venkatesh, "An integrated automatic face detection and recognition system", *Pattern Recognition*, Vol. 35, No. 6, 2002, pp. 1259-1273.

[70] T. Kohonen, "The self-organizing map", *Proceedings of the IEEE*, Vol. 78, No. 9, 1990, pp. 1464-1480.

[71] C. Kotropoulos, A. Tefas and I. Pitas, "Frontal Face Authentication Using Morphological Elastic Graph Matching", *IEEE Trans. Image Processing*, Vol. 9, No. 4, 2000, pp. 555-560.

[72] C. Kotropoulos, A. Tefas and 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.

[73] S. Lawrence, C. L. Giles, A. C. Tsoi and A. D. Back, "Face Recognition: A Convolutional Neural-Network Approach", *IEEE Trans. Neural Networks*, Vol. 8, No. 1, 1997, pp. 98-113.

[74] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. v.d. Malsburg, R. P. Würtz and W. Konen, "Distortion Invariant Object Recognition in the Dynamic Link Architecture", *IEEE Trans. Computers*, Vol. 42, No. 3, 1993, pp. 300-311.

[75] T. S. Lee, "Image Representation Using 2D Gabor Wavelets", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 18, No. 10, 1996, pp. 959-971.

[76] Y. Linde, A. Buzo and R. M. Gray, "An Algorithm for Vector Quantization", *IEEE Trans. Communications*, Vol. 28, No. 1, 1980, pp. 84-95.

[77] M. Lockie (editor), "Facial verification bureau launched by police IT group", *Biometric Technology Today*, Vol. 10, No. 3, 2002, pp. 3-4.

[78] J. Luettin, "Visual Speech and Speaker Recognition", *PhD Thesis*, Department of Computer Science, University of Sheffield, 1997.

[79] R. C. Luo and M. G. Kay, "Introduction" in: *Multisensor Integration and Fusion for Intelligent Machines and Systems* (editors: R. C. Luo and M. G. Kay), Ablex Publishing Corporation, Norwood, NJ, 1995, pp. 1-26.

[80] H. Lütkepohl, *Handbook of Matrices*, John Wiley & Sons, UK, 1996.

[81] J. Makhoul, "Spectral Analysis of Speech by Linear Prediction", *IEEE Trans. Acoustics Speech and Signal Processing*, Vol. 21, No. 3, 1973, pp. 140-148.

[82] J. Markowitz, "Speech systems work together in harmony", *Biometric Technology Today*, Vol. 9, No. 4, 2001, pp. 7-8.

[83] J. Matas, K. Jonsson and J. Kittler, "Fast face localisation and verification", *Image and Vision Computing*, Vol. 17, No. 8, 1999, pp. 757-581.

[84] 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.

[85] I. Miller, J. E. Freund and R. A. Johnson, *Probability and Statistics for Engineers* (4th edition), Prentice-Hall, USA, 1990.

[86] B. Moghaddam and A. Pentland, "Probabilistic Visual Learning for Object Representation", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, 1997, pp. 696-710.

[87] T. K. Moon, "Expectation-maximization Algorithm", *IEEE Signal Processing Magazine*, Vol. 13, No. 6, 1996, pp. 47-60.

[88] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*, Prentice Hall, Upper Saddle River, New Jersey, 2000.

[89] H. Moon, and P. J. Phillips, "Computational and performance aspects of PCA-based face-recognition algorithms", *Perception*, Vol. 30, 2001, pp. 303-321.

[90] K. Messer, J. Matas, J. Kittler, J. Luettin and G. Maitre, "XM2VTSDB: The Extended M2VTS Database", *Proc. Second International Conf. on Audio- and Video-based Biometric Person Authentication*, Washington D.C., 1999, pp. 72-77.

[91] Tom M. Mitchell, *Machine Learning*, WCB/McGraw-Hill, USA, 1997.

[92] B. C. J. Moore, "Frequency Analysis and Masking", in: *Hearing* (editors: D. A. Eddins and D. M. Green), Academic Press, USA, 1995.

[93] B. C. J. Moore, "Information Extraction and Perceptual Grouping in the Auditory System", in: *Human and Machine Perception: Information Fusion* (editors: V. Cantoni, V. D. Gesù, A. Setti and D. Tegolo), Plenum Press, New York, 1997.

[94] J. A. Moorer, "The Optimum Comb Method of Pitch Period Analysis of Continuous Digitized Speech", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. 22, No. 3, 1974, pp. 330-338.

[95] A. V. Nefian and M. H. Hayes, "Hidden Markov Models for Face Recognition", *Proc. International Conf. on Acoustics, Speech and Signal Processing*, Seattle, 1998, Vol. 5, pp. 2721-2724.

[96] J. A. Nelder and R. Mead, "A simplex method for function minimization", *The Computer Journal*, Vol. 7, No. 4, 1965, pp. 308-313.

[97] K. K. Paliwal, "Speech Processing Techniques" in: *Advances in Speech, Hearing and Language Processing* (editor: W. A. Ainsworth), Vol. 1, 1990, pp. 1-78.

[98] K. K. Paliwal, "Spectral Subband Centroid Features for Speech Recognition", *Proc. International Conf. on Acoustics, Speech and Signal Processing*, Seattle, Washington, 1998, Vol. 2, pp. 617-620.

[99] L. F. Pau, "Fusion of multisensor data in pattern recognition" in: *Pattern recognition theory and applications* (editors: J. Kittler, K. S. Fu and L. F. Pau), D Reidel Publ., Dordrecht, Holland, 1982.

[100] J. Picone, "Signal Modeling Techniques in Speech Recognition", *Proceedings of the IEEE*, Vol. 81, No. 9, 1993, pp. 1215-1247.

[101] S. Pigeon and L. Vandendorpe, "The M2VTS Multimodal Face Database (Release 1.00)", *Proc. First International Conf. on Audio- and Video-based Biometric Person Authentication*, Crans-Montana, Switzerland, 1997, pp. 403-409.

[102] S. Pigeon and L. Vandendorpe, "Image-based multi-modal face authentication", *Signal Processing*, Vol. 69, No. 1, 1998, pp. 59-79.

[103] T. W. Parsons, *Voice and Speech Processing*, McGraw-Hill, USA, 1987.

[104] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in C: the Art of Scientific Computing*, Cambridge University Press, Cambridge, 1992.

[105] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications* (3rd ed.), Prentice Hall, USA, 1996.

[106] L. Rabiner and B-H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall PTR, New Jersey, 1993.

[107] V. Radová and J. Psutka, "An Approach to Speaker Identification using Multiple Classifiers", *Proc. IEEE Conf. Acoustics, Speech and Signal Processing*, Munich, Germany, 1997, Vol. 2, pp. 1135-1138.

[108] B. Raducanu, M. Graña, F.X. Albizuri and A. d'Anjou, "Face localization based on the morphological multiscale fingerprints", *Pattern Recognition Letters*, Vol. 22, No. 3-4, 2001, pp. 359-371.

[109] R. A. Redner and H. F. Walker, "Mixture Densities, Maximum Likelihood and the EM Algorithm", *SIAM Review*, Vol. 26, No. 2, 1984, pp. 195-239.

[110] D. A. Reynolds, "A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification", *Technical Report 967*, Lincoln Laboratory, Massachusetts Institute of Technology, 1993.

[111] D. A. Reynolds, "Experimental Evaluation of Features for Robust Speaker Identification", *IEEE Trans. Speech and Audio Processing*, Vol. 2, No. 4, 1994, pp. 639-643.

[112] D. A. Reynolds, "Speaker Identification and Verification Using Gaussian Mixture Speaker Models", *Speech Communication*, Vol. 17, No. 1-2, 1995, pp. 91-108.

[113] D. A. Reynolds, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models", *IEEE Trans. Speech and Audio Processing*, Vol. 3, No. 1, 1995, pp. 72-83.

[114] D. A. Reynolds, "Comparison of Background Normalization Methods for Text-Independent Speaker Verification", *Proc. 5th European Conf. Speech Communication and Technology*, Rhodes, Greece, 1997, Vol. 2, pp. 963-966.

[115] D. Reynolds, T. Quatieri and R. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models", *Digital Signal Processing*, Vol. 10, No. 1-3, 2000, pp. 19-41.

[116] C. P. Robert, *The Bayesian Choice: A Decision-Theoretic Motivation*, Springer-Verlag, New York, 1994.

[117] A. E. Rosenberg, J. DeLong, C-H. Lee, B-H. Juang and F. K. Soong, "The Use of Cohort Normalized Scores for Speaker Verification", *Proc. International Conf. Spoken Language Processing*, Alberta, 1992, Vol. 1, pp. 599-602.

[118] A.E. Rosenberg and S. Parthasarathy, "Speaker Background Models for Connected Digit Password Speaker Verification", *Proc. International Conf. Acoustics, Speech and Signal Processing*, Atlanta, 1996, Vol. 1, pp. 81-84.

[119] M. J. Ross, "Average Magnitude Diffrence Function Pitch Extractor", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. 22, No. 5, 1974, pp. 353-362.

[120] A. Ross, A. Jain and J-Z. Qian, "Information Fusion in Biometrics", *Proc. 3rd Audio- and Video-Based Biometric Person Authentication*, Halmstad, 2001, pp. 354-359.

[121] V. Roth and V. Steinhage, "Nonlinear Discriminant Analysis using Kernel Functions", Technical Report Nr IAI-TR-99-7 (ISSN 0944-8535), University of Bonn, 1999.

[122] H. A. Rowley, S. Baluja and T. Kanade, "Neural Network-Based Face Detection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 20, No. 1, 1998, pp. 23-38.

[123] F. Samaria, *Face Recognition Using Hidden Markov Models*, PhD Thesis, University of Cambridge, 1994.

[124] D. O'Shaughnessy, *Speech communications: human and machine* (2nd ed.), IEEE Press, New York, 2000.

[125] F. K. Soong and A. E. Rosenberg, "On the Use of Instantaneous and Transitional Spectral Information in Speaker Recognition", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. 36, No. 6, 1988, pp. 871-879.

[126] C. Sanderson and K. K. Paliwal, "Multi-Modal Person Verification System Based on Face Profiles and Speech", *Fifth International Symposium on Signal Processing and its Applications*, Brisbane, 1999, Vol. 2, pp. 947-950.

[127] C. Sanderson and K. K. Paliwal, "Adaptive Multi-Modal Person Verification System", *Proc. First IEEE Pacific-Rim Conf. on Multimedia*, Sydney, 2000, pp. 210-213.

[128] C. Sanderson and K. K. Paliwal, "Training Method of a Piecewise Linear Classifier for a Multi-Modal Person Verification System", *Proc. Eighth Australian International Conf. on Speech Science and Technology*, Canberra, 2000, pp. 312-317.

[129] C. Sanderson and K. K. Paliwal, "Noise Compensation in a Multi-Modal Verification System", *Proc. International Conf. on Acoustics, Speech and Signal Processing*, Salt Lake City, 2001, Vol. 1, pp. 157-160.

[130] C. Sanderson and K. K. Paliwal, "Information Fusion for Robust Speaker Verification", *Proc. 7th European Conf. Speech Communication and Technology*, Aalborg, 2001, Vol. 2, pp. 755-758.

[131] C. Sanderson and K. K. Paliwal, "Robust Face-Based Identity Verification", *Proc. Microelectronic Engineering Research Conf.* Brisbane, Australia, 2001.

[132] C. Sanderson and K. K. Paliwal, "Polynomial Features for Robust Face Authentication", *Proc. International Conf. Image Processing*, Rochester, 2002. pp. 997-1000 (Vol. 3).

[133] C. Sanderson and K. K. Paliwal, "Likelihood Normalization for Face Authentication in Variable Recording Conditions", *Proc. International Conf. Image Processing*, Rochester, 2002, pp. 301-304 (Vol. 1).

[134] C. Sanderson and K. K. Paliwal, "Fast Feature Extraction Method for Robust Face Verification", *IEE Electronics Letters*, Vol. 38, No. 25, 2002, pp. 1648-1650.

[135] C. Sanderson and 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.

[136] C. Sanderson and K. K. Paliwal, "Features for Robust Face-based Identity Verification", *Signal Processing*, Vol. 83, No. 5, 2003, pp. 931-940.

[137] C. Sanderson and K. K. Paliwal, "Fast Features for Face Authentication Under Illumination Direction Changes", submitted to *Pattern Recognition Letters* on 7-Feb-2002.

[138] C. Sanderson and K. K. Paliwal, "Likelihood Normalization for Face Verification in Variable Image Conditions", submitted to *Image and Vision Computing* on 19-Mar-2002.

[139] C. Sanderson and K. K. Paliwal, "Structurally Noise Resistant Classifier for Multi-Modal Person Verification", submitted to *Pattern Recognition Letters* on 21-Jun-2002.

[140] C. Sanderson and K. K. Paliwal, "Automatic Person Verification Using Speech and Face Information", submitted to *Digital Signal Processing* on 4-Aug-2002.

[141] P. Silsbee and A. Bovik, "Computer Lipreading for Improved Accuracy in Automatic Speech Recognition", *IEEE Trans. Speech and Audio Processing* Vol. 4, No. 5, 1996, pp. 337-351.

[142] F. Smeraldi and J. Bigün, "Retinal vision applied to facial features detection and face authentication", *Pattern Recognition Letters*, Vol. 23, No. 4, 2002, pp. 463-473.

[143] B. Sonesson, "The functional anatomy of the speech organs" in: *Manual of Phonetics* (editor: B. Malmberg), North-Holland, Amsterdam, 1968, pp. 45-75.

[144] E. W. Swokowski, *Calculus* (5th ed.), PWS-Kent, USA, 1991.

[145] A. Tefas, C. Kotropoulos and I. Pitas, "Face Authentication by Using Elastic Graph Matching and Support Vector Machines", *Proc. International Conf. Acoustics, Speech and Signal Processing*, Istanbul, 2000, pp. 2409-2412 (Vol. 4).

[146] A. Tefas, C. Kotropoulos and I. Pitas, "Using Support Vector Machines to Enhance the Performance of Elastic Matching for Frontal Face Authentication", *IEEE Trans. Pattern Analysis and Machine Intelligence* Vol. 23, No. 7, 2001, pp. 735-745.

[147] R. R. Tenney and N. R. Sandell Jr., "Detection with Distributed Sensors", *IEEE Trans. on Aerospace and Electronic Syst.*, Vol. 17, 1981, pp. 98-101.

[148] R. R. Tenney and N. R. Sandell Jr., "Strategies for Distributed Decisionmaking", *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 11, 1981, pp. 527-537.

[149] T. Thong and Y. C. Jenq, "Hardware and Architecture" in: *Handbook for Digital Signal Processing* (editors: S. K. Mitra and J. F. Kaiser), John Wiley & Sons, USA, 1993, pp. 721-781.

[150] M. Turk and A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, 1991, pp. 71-86.

[151] D. Valentin, H. Abdi, A. J. O'Toole and G. W. Cottrell, "Connectionist Models of Face Processing: A Survey", *Pattern Recognition*, Vol. 27, No. 9, 1994, pp. 1209-1230.

[152] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

[153] P. K. Varshney, *Distributed Detection and Data Fusion*, Springer-Verlag, New York, 1997.

[154] P. Verlinde, *A Contribution to Multi-Modal Identity Verification Using Decision Fusion*, PhD Thesis, Department of Signal and Image Processing, Telecom Paris, France, 1999.

[155] T. Wark, S. Sridharan and V. Chandran, "Robust Speaker Verification via Fusion of Speech and Lip Modalities", *Proc. International Conf. Acoustics, Speech and Signal Processing*, Phoenix, Arizona, 1999, Vol. 6, pp. 3061-3064.

[156] T. Wark, S. Sridharan and V. Chandran, "Robust Speaker Verification via Asynchronous Fusion of Speech and Lip Information", *Proc. 2nd International Conf. Audio- and Video-based Biometric Person Authentication*, Washington, D.C., 1999, pp. 37-42.

[157] 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.

[158] G. K. Wallace, "The JPEG Still Picture Compression Standard", *Communications of the Association for Computing Machinery*, Vol. 34, No. 4, 1991, pp. 30-44.

[159] G. K. Wallace, "The JPEG still picture compression standard", *IEEE Trans. Consumer Electronics*, Vol. 38, No. 1, 1992, pp. xviii-xxxiv.

[160] *Webster's Revised Unabridged Dictionary*, MICRA Inc., Plainfield, NJ, 1998.

[161] B. Wildermoth and K. K. Paliwal, "Use of Voicing and Pitch Information for Speaker Recognition", *Proc. 8th Australian International Conf. Speech Science and Technology*, Canberra, 2000, pp. 324-328.

[162] K-W. Wong, K-M. Lam and W-C. Siu, "An efficient algorithm for human face detection and facial feature extraction under different conditions", *Pattern Recognition*, Vol. 34, No. 10, 2001, pp. 1993-2004.

[163] J. D. Woodward, "Biometrics: Privacy's Foe or Privacy's Friend?", *Proceedings of the IEEE*, Vol. 85, No. 9, 1997, pp. 1480-1492.

[164] J. Zhang, Y. Yan and M. Lades, "Face Recognition: Eigenface, Elastic Matching, and Neural Nets", *Proceedings of the IEEE*, Vol. 85, No. 9, 1997, pp. 1423-1435.