Histogram Equalization for Robust Text-independent Speaker Verification in Telephone Environments

Prepared by: Marshalleno Skosan

Supervised by: Dr. Daniel J. Mashao

University of Cape Town
Department of Electrical Engineering
March 2005

This dissertation is submitted to the University of Cape Town in fulfilment of the academic requirements for the Degree of Master of Science in Engineering.

Declaration

I declare that this dissertation is my own work. It is being submitted for the degree of Master of Science in Engineering at the University of Cape Town. It has not been submitted before for any degree or examination in any other university.

Signature of Author . . . . . .

Marshalleno Skosan

March 2005

Acknowledgments

I am profoundly grateful to my supervisor, Dr. Daniel J. Mashao, for the assistance he provided throughout the execution of this project. I am especially grateful the many skills that I acquired under his supervision. I am just as grateful to my parents for their relentless support and encouragement throughout my studies. I would not be where I am today if it weren't for them. My colleagues and friends in the STAR and CRG research groups also deserve thanks for the laughs that we shared when the going was tough and, for their all their useful suggestions. A special thank you goes out to all the researchers who personally provided me with useful papers concerning various areas of speaker verification and, all those who reviewed my thesis. Last but not least, I am grateful to Telkom for their financial support throughout my tertiary education. But, above all, I am eternally grateful to the One who watches over my soul. To Him belongs all the glory, honour and praise.

Abstract

While it is common for speaker recognition systems to perform well in ideal conditions, performance degrades when these systems are exposed to adverse conditions. This degradation in performance becomes more evident when speaker recognition systems are trained and tested in different recording conditions. For the technology to prosper, speaker recognition needs to perform reliably regardless of the conditions under which training and testing are done.

This thesis is aimed at mitigating the problem of mismatched training and test conditions by using a technique known as Histogram Equalization. Here, it is used to improve the robustness of a speaker verification system evaluated on the NIST 2000 database. This database contains speech degraded by various artefacts caused by telephone transmission. Histogram Equalization is applied directly to the features extracted from a particular speaker's training and test speech. In so doing, it modifies the underlying feature distributions such that they become less environment-dependent and more consistent across different recording conditions. The technique is shown to lead to a relative improvement in the equal error rate of a speaker verification system employing cepstral mean normalization of more than 11%. A proposed variation of Histogram Equalization, in which the technique is applied to the features extracted from short adjacent segments of speech within an utterance, is also shown to improve performance above that of the original version of the technique.

Key words: Histogram Equalization, robust speaker verification

Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

CDF	– Cumulative Distribution Function
CMN	– Cepstral Mean Normalization
DB-GMM	– Distance-based Gaussian Mixture Model
DCF	– Detection Cost Function
DCT	– Discrete cosine transform
DET	– Detection Error Trade-off Curve
EER	– Equal Error Rate
EIH	– Ensemble Interval Histogram
EM Algorithm	– Expectation-Maximization Algorithm
FAR	– False Accept Rate
FRR	– False Reject Rate
GMM	– Gaussian Mixture Model
H-norm	– Handset Normalization
HEQ	– Histogram Equalization
HMM	– Hidden Markov Model
HT-norm	– Handset-dependent Test Normalization
Hz	– Hertz
LPC	– Linear Predictive Coding
LPCC	– Linear Predictive Cepstral Coefficient
MACV	– Maximum Autocorrelation Value
MFCC	– Mel-frequency Cepstral Coefficient
$MFCC_1$	– The first component of a MFCC feature vector
MLLR	– Maximum Likelihood Linear Regression
MVN	– Mean and Variance Normalization
NIST	– National Institute of Standards and Technology
PDF	– Probability Density Function
RASTA	– Relative Spectral Processing
ROC	– Receiver Operating Characteristic
SS	– Spectral Subtraction
SVM	– Support Vector Machine
T-norm	– Test Normalization
UBM	– Universal Background Model
VAD	– Voice Activity Detector
VTS	– Vector Taylor Series
Z-norm	– Zero Normalization

Chapter 1

Introduction

Recent advances in speech technology have enabled researchers to design speaker recognition systems that are capable of excellent performance in ideal conditions. However, real-world conditions are far from ideal which leads to a deterioration in the performance of these systems. Furthermore, it is highly desirable for these architectures to perform reliably across telephone networks as this would be convenient for most users. Unfortunately, the quality of a speech signal is adversely affected by transmitting it over a telephone network. This degrades the performance of speaker recognition systems operating in telephone environments. This degradation in performance becomes more evident when speaker recognition systems are trained and tested in different recording conditions (e.g., when different telephone handsets are used to collect the training and test speech). This thesis is aimed at improving the robustness of a speaker verification system evaluated on speech degraded by telephone transmission.

1.1 A brief overview of speaker recognition technology

Every day we use numerous keys, cards, badges, pin numbers and passwords to confirm our identities. Even though these mechanisms ensure secure access to various resources and facilities, their major disadvantage is that they can be lost, stolen, forgotten or even counterfeited. Physiological characteristics (e.g., facial features, fingerprints and retinal patterns) and behavioural characteristics (e.g., handwriting and speech) on the other hand are to a large extent specific to each individual and can also be used as a means of authentication. These characteristics are collectively referred to as *biometrics* and have the additional advantage that they are based on *who one is* and not on *what one remembers or possesses* [1]. For this reason they do not suffer from the same problems encountered when using the possession- or knowledge-based authentication mechanisms mentioned previously. Biometric person recognition however, does not completely solve

the problem of person authentication – bad illumination in face recognition, cuts and bruises in fingerprint recognition and background noise in speaker recognition are all pitfalls of the technology [1]. As such, the design, implementation and deployment of such systems are a non-trivial task.

The primary focus of this study is on the use of speech as a means of automatically determining an individual's identity. This is referred to as *speaker recognition* [2-4]. Simply put, speaker recognition "*is the general term used to include all of the many tasks of discriminating people based on the sound of their voices*" [4]. Reynolds [3] more precisely states that "*the goal of automatic speaker recognition systems is to extract, characterize and recognize information in the speech signal conveying speaker identity*". This is in contrast to speech recognition in which the goal is to automatically extract the word sequence in the speech signal so as to produce a textual output [5]. One of the advantages of using speech to determine an individual's identity is that speech is our most natural means of interacting with each other. Thus, speaker recognition, in contrast to other biometric identification techniques, is generally regarded as being less intrusive to perform – there is no need to place one's head in a specific position so that some system can scan one's iris for example. In addition, there is already a well established infrastructure in place for transmitting speech from one point to another: the ubiquitous telephone network. This makes the large scale deployment of speaker recognition technology easier and more cost-effective than that of other biometric identification systems as no special equipment is needed (only the speaker recognition system of course). In the following section a brief overview of the basic components and operation of a speaker recognition system is presented.

1.1.1 How does it work?

Speaker recognition research has been conducted for more than four decades. This ranges back to the visual *spectrogram*[1] comparisons made by Kersta in the early 1960's [6] to the sophisticated statistical pattern matching techniques employed in contemporary speaker recognition systems [7]. As a result, speaker recognition can be regarded as a subset of the larger area of *pattern recognition*. Pattern recognition is defined as "*the study of how machines can observe the environment, learn to distinguish patterns of interest from their background, and make sound and reasonable decisions about the categories of the patterns*" [8]. According to the same reference, pattern recognition generally involves three aspects: (1) data acquisition and pre-processing; (2) data representation; and (3) decision making. In addition, at any time, a pattern recognition system is

[1] A spectrogram is a three-dimensional display of a speech signal. It shows how the speech intensity in different frequency bands changes over time [36].

either in one of two modes of operation: the training mode or the test mode. In the training mode, the system "learns" the categories to which the input training patterns belong and, in the test mode, patterns are classified according to their similarity to these categories. The following is a discussion of pattern recognition as it pertains to speaker recognition.

Figure 1-1 depicts a generic speaker recognition system. Illustrated are the two modes of operation as well as all of the pattern recognition aspects mentioned previously (each of these is represented as an independent component). However, in speaker recognition terminology, data acquisition is referred to as *feature extraction* or *speech parameterisation* (which occurs in the *front-end*), data representation is referred to as *speaker modelling* and decision-making is often referred to as *classification* (which together with speaker modelling occurs in the *back-end*). In the training mode new speakers are enrolled into the system and in the test mode the recognition of speakers takes place.

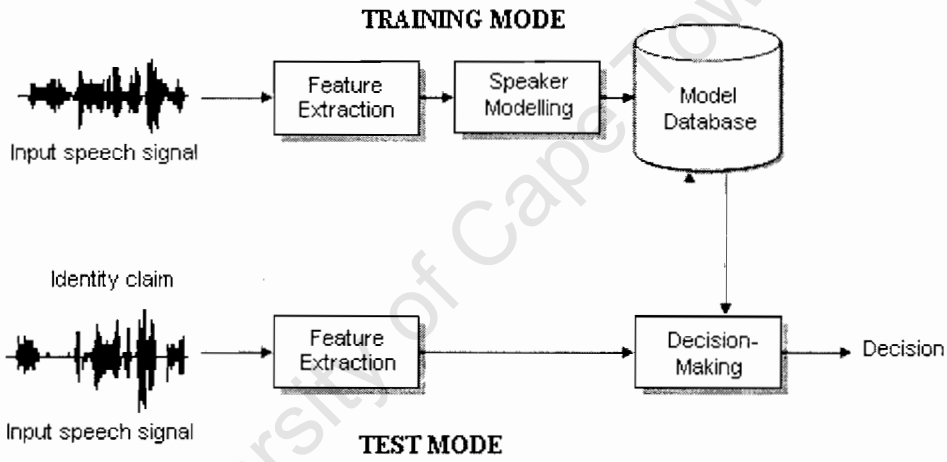

Figure 1-1: A generic speaker recognition system

The purpose of feature extraction is to extract speaker-dependent information from a raw speech signal and, in the process, convert the speech signal into a more compact and efficient representation. The output of this component is a sequence of feature vectors where the individual elements of each feature vector are known as *features*. Desirable attributes of the features used for speaker recognition are that they should [9, 10]:

- differentiate between speakers while being tolerant of intra-speaker variabilities (like the speaker's health or mood),
- occur naturally and frequently in the speech signal,
- be easily measurable from the speech signal,
- be minimally affected by ambient noise and transmission over communication channels,
- be stable over time, and
- not be predisposed to mimicry by impostors.

As illustrated by Figure 1-1, feature extraction takes place in both the training mode and the test mode. In the training mode the features generated from the input speech signal are fed into the speaker modelling component. This component is responsible for creating a model of each speaker's speech characteristics. In so doing, the system "learns" the speaker's voice. Ideally speaker models should [3]:

- be based on sound theoretical principles (so that model behaviour can easily be understood and so that extensions and improvements can be approached from a mathematical point of view),
- generalise well to unseen data (i.e., the model should not overfit the training data leading to poor performance during testing), and
- be economical as far as memory and computational resources are concerned.

The models generated by the speaker modelling component are subsequently stored in a model database for use during testing. During testing, the decision-making component compares the features generated by the feature extraction component to the speaker models stored in the model database and a measure of similarity (usually a numerical value) is computed. Depending on the task at hand, the decision-making component uses these values to either assign a speaker identity to the input speech signal or to verify that it belongs to a particular speaker. In order to verify that the speech signal belongs to a particular speaker an identity claim has to be made. This is the reason for the optional identity claim entered in the test mode (see Figure 1-1). Speaker recognition systems can be classified according to the constraints placed on the text of the input speech signals [11]. This is discussed in the following section.

1.1.2 Spoken text constraints

When the spoken text is limited to a specific word, phrase or sentence, the system is said to be *text-dependent*. In such systems users are expected to recite a pre-defined password for example. When there are no constraints placed on the spoken text, the system is said to be *text-independent*. Most commercial systems in operation today are text-dependent, as the additional knowledge of the specific phrase to expect can be used to enhance security. This is done by simultaneously making use of speech recognition to verify the text of the input phrase [12]. However, text-independent speaker recognition systems are more flexible than text-dependent ones as recognition can be performed in the background (i.e., regardless of the spoken utterance and without explicit user co-operation) while users are engaged in other speech interactions [3, 9]. This flexibility also lends to the weakness of text-independent systems as they require more training data to ensure that the full range of a particular individual's speech characteristics are captured [5].

One of the major limitations of text-dependent systems is that impostors can fool such systems by playing a recording of a legitimate speaker saying his (or her) specific pass-phrase. For this reason

challenge-response systems [13] (also known as text-prompted systems) were introduced. These systems randomly prompt users to say phrases from a small pre-defined vocabulary. This makes it harder for impostors to fool such systems as the phrases to be repeated cannot be predicted beforehand. However, challenge-response systems increase the time taken to enrol a new speaker as these speakers have to be recorded saying multiple words or phrases [14]. In addition to being either text-dependent or text-independent, several variations of speaker recognition also exist. These are discussed in the following section.

1.1.3 Types of speaker recognition

Speaker recognition systems either: (1) assign an identity to the voice of an unknown speaker (this is known as speaker identification); (2) verify that a speaker is who he (or she) claims to be (this is known as speaker verification); (3) separate one speaker's voice from another in a multi-speaker environment (this is known as speaker segmentation); or (4) determine if, when and for how long a particular speaker is speaking in a multi-speaker environment (this is known as speaker tracking) [7]. Most research papers however, focus on the tasks of speaker identification and speaker verification. The same approach is taken here. The interested reader is referred to references [15] and [16] for more information concerning speaker segmentation and speaker tracking.

Given an utterance, *speaker identification* is the task of deciding who, among a finite set of enrolled speakers, produced it [4]. The utterance is scored against all possible speaker models, and the model that produces the highest score determines the speaker's identity. Thus, the task involves having to make a 1: N classification, where N is the number of enrolled speakers (i.e., the population size). One of the main limitations of speaker identification is that as N increases, the probability of correctly identifying a speaker decreases [17]. In addition to the decrease in accuracy, the size of N also adversely affects the execution time of such systems – the larger the population size, the longer the execution time. The speaker identification task, as described here, is referred to as *closed set*, since the actual speaker is one of the finite set of enrolled speakers. When the system has the option of declaring that the actual speaker is not part of this set, the task is referred to as *open set* speaker identification [3].

While speaker identification allows one to determine the degree to which human voices are unique, the large population problem limits its immediate commercial potential. For this reason, the task of speaker verification has received considerable research interest in recent years. Given an utterance and an identity claim, *speaker verification* (also known as *speaker detection* or *speaker authentication*) is the task of determining whether the utterance can be attributed to the enrolled speaker associated with the identity claim or not [3, 4]. This is done by testing the model

of the targeted (or hypothesized) speaker with the utterance, comparing the score obtained to a threshold, and deciding on the basis of this comparison whether or not to accept the claimant. Thus, a speaker verification system needs to make a binary (2-class) decision as the identity claim is either accepted or rejected. When accepted, the claimant is referred to as either a *true*, *legitimate* or *target speaker*. Upon rejection, the claimant is referred to as an *impostor* or *non-target speaker* [4]. Speaker verification can also be viewed as a special case of open set speaker identification with N equal to 1. However, it differs from speaker identification in that its performance is not dependent on the number of potential impostors (i.e., the population size). Even so, the composition of the impostor set will naturally affect performance if impostors with speech characteristics similar to the targeted speaker(s) are selected. The following section provides an overview of areas in which speaker recognition technology has been employed.

1.1.4 Applications of speaker recognition technology

According to Gish and Schmidt [18], "*the potential for application of speaker recognition systems exists any time speakers are unknown and their identities are important*". As a result, speaker recognition technology has been applied in many application areas over the years. These include preventing toll fraud in telephone networks [19]; controlling access to restricted sites and resources (both on-site and remotely) [9]; securing financial transactions over the telephone [20]; monitoring criminals placed under house-arrest or on parole [13]; ensuring that only authorised inmates make outbound calls [21]; and monitoring the time and attendance of employees. Another application of speaker recognition is that of *audio mining* (also known as *speech data management* [3] and *information structuring* [7]). "*Audio mining entails indexing and searching of audio and audio-visual sources such as movies, TV and radio broadcasts, call centre recordings and videotaped meetings*" [13]. In audio mining speaker recognition is used to tag each utterance with the identity of the speaker that said it. Speaker recognition has also been explored for use in forensic casework [7, 22].

The *Home Shopping Network*[2] currently deploys one of the largest applications of speaker recognition technology [12, 23]. This service allows users to order products over the telephone by calling a toll-free number. The company has approximately five million customers. Every day it handles about 160,000 calls and ships more than 100,000 packages. Most orders are handled by human operators. However, there are times when all the lines are occupied. In cases like these, the user has the option of using an automated ordering service. After a playing welcome message, the user is asked to speak his (or her) telephone number including the area code. Speaker recognition

[2] http://www.hsn.com

technology is then used to verify the identity of the user. If the user is successfully verified, he (or she) can start ordering products; otherwise the user is redirected to a human operator. In so doing, speaker recognition is used to enlarge the capacity of the telephone service. In addition, the risk of falsely rejecting a legitimate user is diminished since a human operator double checks all rejections. The technology however is not perfect, and in cases where impostors are falsely accepted, the ordered goods are still sent to the owner of the account and not to the address of an unauthorised third party – this detracts potential impostors. More extensive reviews of speaker recognition applications can be found in references [13] and [23].

Speaker recognition systems have been shown to perform extremely well in ideal conditions: e.g., a quiet recording environment with high-quality microphones and consecutive recordings of training and test speech [17]. This is also true when the acoustic conditions encountered during training and testing are similar (a scenario known as *matched conditions*). However, when exposed to real-world conditions and different environments (e.g., speech degraded by excessive ambient noise and telephone transmission), speaker recognition systems exhibit a considerable degradation in performance [17]. In the following section several impairments that degrade the performance of speaker recognition systems are discussed.

1.2 Factors affecting speaker recognition performance

In general, the factors affecting speaker recognition systems can to a large extent be attributed to variations in the voices or actions of the speakers themselves, or to the circumstances under which input speech signals are acquired [2, 4, 7].

1.2.1 Speaker variability

The main reason that the variability of a speaker's voice affects a speaker recognition system is because speech is a behavioural characteristic. As such, it is subject to the physical and emotional state of the speaker such as when the speaker is tired, ill or under stress. At times, speakers also unconsciously change their level of speech effort and speaking rate when exposed to high levels of ambient noise. This is known as the *Lombard effect* [4]. In addition, human voices also change over time due to the natural effects of ageing. This is one of the reasons that when training and testing is conducted in a single session, good speaker recognition performance is observed [17]. Some speakers also speak different languages interchangeably, with the linguistic content of the speech dependent on the context of the dialogue, the choice of the words used and the way in which the words are pronounced [23]. All these factors are collectively referred to as *intra-speaker variation* [23]. As mentioned previously, the features extracted from speech signals used in speaker recognition systems should ideally be robust to these types of variation.

Speaker recognition systems are also influenced by the actions of speakers. This includes the co-operativeness of speakers. If speakers do not interact with the system as directed, by speaking naturally and for the length of time required, system performance will be compromised. In addition, misread or misspoken prompts can also affect the performance of speaker recognition systems [2]. It should also be taken into consideration that speakers who are not familiar with speaker recognition systems will tend to make more mistakes than those who have prior experience of interacting with such systems. Of course, speaker recognition systems also have to be robust to impostor attacks made by malicious individuals who modify their voices so as to impersonate legitimate speakers.

1.2.2 Variability in recording conditions

When speech is acquired in real-world scenarios, it is affected by the environment in which it was collected and by the equipment used to collect it (i.e., it is affected by the recording conditions). This has a major impact on the quality of the speech signal. A typical example is that of speech collected over telephone networks. Here, the speech signal is subject to distortions caused by various telephone handset microphones; unpredictable levels of noise in the background or on the line; as well as different transmission channels and network types (mobile versus fixed-line networks for example) [4, 14, 18, 24, 25]. The bandwidth of telephone channels, as well as the various compression techniques in use, also distorts speech signals during telephone transmission. In addition, users also have various ways of holding telephone handsets and often change the orientation of these handsets while talking [4]. The growth in the use of mobile devices also means that speaker recognition can be expected to be performed in several uncontrolled environments (e.g., in crowded shopping malls, in cars, or in rooms with inconsistent or poor acoustics).

The subtle physiological changes in a speaker's voice and variability in the recording equipment and environment which occur between recording sessions is referred to as *intersession variability* [26]. Intersession variability results in speech data being acquired in what is known as *mismatched conditions*. Mismatched conditions generally lead to an acoustic mismatch between the speech data acquired during training and testing. This mismatch can severely degrade the performance of a speaker recognition system. As a result, many researchers cite mismatched conditions as one of the biggest challenges facing contemporary speaker recognition systems [3, 4, 18, 27].

1.3 Problem statement

For the technology to prosper, speaker recognition needs to perform reliably regardless of the conditions under which training and testing is done. The speaker recognition applications mentioned in Section 1.1.4 suggest that the ability to reliably recognise individuals over the telephone has great commercial potential. However, from the discussion presented in Section 1.2.2, speech transmitted over telephone networks is susceptible to numerous distortions. Furthermore, many of these distortions vary between recording sessions. As such, training and test speech is often obtained in mismatched conditions which can degrade the performance of speaker recognition systems operating in telephone environments.

There have been many studies aimed at mitigating mismatched conditions so as to improve the robustness of speaker recognition systems. See for example references [28] and [29]. Mismatched conditions can be viewed as introducing a disparity between the underlying distribution of feature vectors extracted from a particular speaker's speech during training and testing. This thesis proposes and evaluates a technique that is aimed at explicitly minimising this disparity. The technique is known as *Histogram Equalization* and is applied here to improve the robustness of a speaker verification system evaluated on speech degraded by telephone transmission. Histogram Equalization is applied directly to the features extracted from a particular speaker's training and test speech. In so doing, it modifies the underlying feature distributions such that they become less environment-dependent and more consistent across different recording conditions.

1.4 Research objectives

This thesis has three main objectives. The first objective is to provide a comprehensive review of contemporary speaker verification literature with particular emphasis on techniques that are generally regarded standard as practice in the field and techniques that have been used to improve the robustness of speaker verification systems. Sufficient references to other concepts and methodologies used in speaker verification will also be provided for the interested reader.

The second objective of this thesis is to design and implement a baseline text-independent speaker verification system using concepts and methodologies detailed in contemporary literature. This will provide an experimental framework for evaluating the Histogram Equalization technique. Following successful implementation, the system's performance will be compared to that of other systems in literature that are based on similar techniques and that have been evaluated under similar conditions. This will be done so as to verify the implementation of the system. The performance of this system will then be used as a benchmark against which all subsequent improvements will be compared. It is also the author's intention to implement software that is well-commented

and modularised so that the software can easily be integrated into future projects in the Speech and Technology Research Group[3] at the University of Cape Town.

After verifying the implementation of the baseline system, the final objective of this thesis will be to implement Histogram Equalization so as to minimise the mismatch between two distributions. The main motivation for the use of this technique, when applied in speaker verification, is that it could be used to reduce the mismatch between speech obtained in different training and test conditions and hence, lead to improved system performance. The technique is to be evaluated on speech that has been degraded by telephone transmission, different telephone handset microphones, background noises and various periods that elapse between recording sessions. The performance of the technique will be compared to that of other commonly used techniques aimed at mitigating the problem of mismatched conditions in speaker verification. The main design criteria for the technique are that it should be simple, computationally feasible and result in improved performance.

Given the difficulties associated with developing a speaker verification system that performs perfectly in adverse environments, the author is aware that no single research effort conducted over such a short period of time will completely solve the robustness issue in speaker verification. The technique developed here is only a small contribution to the vast body of speaker recognition literature. As such, the author's main aim is to approach the robustness issue from a new angle using well-established statistical concepts that have had limited application in the field of speaker recognition. This is done to gain further insight into how to mitigate the problem of mismatched training and test conditions in speaker verification.

1.5 Contribution to knowledge

As mentioned previously, one of the main objectives of this study is to minimise the problem of mismatched training and test conditions in speaker verification. This is done by using a technique known as Histogram Equalization (HEQ). HEQ is commonly employed in digital image processing to enhance the brightness and contrast of digital images. When digital images are too dull and lack contrast, their histogram of pixel values occupies only a small region of the full grey-level scale. In cases like these, HEQ can be used to map the compressed histogram to a more uniform histogram occupying a larger portion of the grey-level scale. In so doing, the quality of the image can be improved. In this work, Histogram Equalization's ability to map one histogram to another is used to map feature distributions obtained during training and testing to a common reference distribution. Thereby, it is shown that mismatches caused by the use of different telephone hand-

[3] http://www.star.za.net

set microphones, telephone channels and recording conditions encountered during training and testing, are minimised.

In Section 4.4 several previous applications of HEQ in speech-related research is reviewed. This review will show that while HEQ has been used to improve the robustness of numerous speech recognition systems evaluated on several tasks involving speech corrupted by adverse recording conditions, it has had limited application in the area of speaker recognition and on speech corrupted by telephone transmission. The motivation for using HEQ, as well as its theoretical development, implementation and evaluation are covered in Chapters 4 and 6. A few variations of the technique are also proposed and analysed. While the use of HEQ to improve the robustness of a speaker verification system in telephone environments is the author's main contribution to knowledge, in 2004, the author also published two peer-reviewed conference papers on the subject (see references [30] and [31]). These papers showed HEQ to be a very promising approach to mitigating the problem of mismatched recording conditions in speaker recognition research.

1.6 Scope and limitations

While a general overview of speaker recognition was presented in Section 1.1, the remainder of this study is limited to speaker verification (i.e., the task of verifying that an individual is who he (or she) claims to be). In particular, the problem of robust speaker verification in mismatched training and test conditions is addressed. However, a number of the techniques explored in the rest of this document are expected to generalise well to other types of speaker recognition.

As mentioned previously, a technique known as Histogram Equalization is proposed to minimise the disparity between training and test feature distributions obtained in different recording conditions. While there are many ways of making speaker verification systems robust to mismatched training and test conditions, this study focuses on a technique that is applied at the feature level (i.e., on the actual features extracted from each speech utterance). For this reason, the technique is only compared to other commonly used techniques that also operate at the feature level. This is done so as to determine whether HEQ is superior to these techniques or not. Other methods of improving speaker verification performance in adverse environments are mentioned, but are not explored in any detail.

The evaluation of HEQ is limited to speech degraded by telephone transmission. In particular, the speech data contained in the NIST 2000 database was used. This database is discussed in detail in Section 5.1.2. The challenges presented by the data in this database include limited bandwidth, channel noise from various sources, the use of different microphones, recordings from different

locations and recordings collected over a period of time. The robustness of HEQ to degradations caused by sources of mismatch other than these was not evaluated.

Many of the design decisions made when implementing the baseline speaker verification system were centred on the trade-off between system performance and computational complexity. In a number of cases, the system parameters which led to the best overall system performance at the lowest computational cost were selected. This was primarily the case when factors other than the extraction of robust features were considered, as the main purpose of developing a baseline system was to provide an experimental framework in which to evaluate HEQ and to provide a benchmark against which to compare potential improvements. The baseline system was constructed using standard techniques employed in contemporary speaker verification systems. Thus, no novel feature extraction, speaker modelling or decision-making techniques were explored.

1.7 Thesis outline

The remainder of this document is organised as follows:

- The key area that this study addresses is that of speaker verification. Chapter 2 will provide an in-depth review of the fundamental techniques and modules required to build a contemporary speaker verification system. Various aspects of feature extraction, decision-making and speaker modelling will be discussed. Emphasis will be placed on those techniques that are commonly employed in contemporary speaker verification systems since they will later be used to create an experimental framework for evaluating HEQ. Finally, several metrics commonly used to gauge the performance of speaker verification systems will also be presented.

- This study is aimed at making speaker verification systems more robust to training and test data collected in different recording conditions. Chapter 3 will provide a review of several compensation techniques that have been proposed to improve the performance and robustness of speaker recognition systems operating in mismatched training and test conditions. As this thesis proposes a technique that operates at the feature level, particular emphasis will be placed on feature-based compensation techniques. This chapter will also provide an analysis of the effects of additive noise and linear time-invariant filters on a speech signal. This is important as it will provide some insight into the degradations at which the compensation provided by HEQ is aimed.

- As mentioned previously, this study proposes a feature-based compensation technique, known as Histogram Equalization. Chapter 4 introduces Histogram Equalization as a technique aimed at making feature distributions more consistent across different recording environments. In particular, its mathematical formulation as well as its image and speech processing background will be covered. A simple algorithm for implementing the technique will also be provided.

- In order to evaluate HEQ, an experimental framework is required. Chapter 5 will describe the design and implementation of such a framework. The framework will take the form of a baseline text-independent speaker verification system built using techniques discussed in contemporary literature. The performance of this system will be used as a benchmark against which all subsequent improvements will be compared. The characteristics of the speech database on which HEQ will be evaluated, as well as the procedure for using this database, will also be covered.

- In Chapter 6, the HEQ technique proposed in this study will be evaluated. In particular, HEQ will be compared to other feature-based compensation techniques and several variations of the technique will be explored. An analysis of the results obtained will also be provided.

- Finally, Chapter 7 will summarise the achievements of the work done in this study and highlight key conclusions based on the research and experimental work conducted. Directions for future work involving many of the concepts and methodologies explored in this study will also be provided.

1.8 Summary

The main aim of this chapter was to provide a very general introduction to the area of speaker recognition so as to allow the reader to become familiar with the various terms and concepts used throughout the remainder of this document. The mismatch between the recording conditions observed during the training and testing of a speaker verification system was highlighted as the particular problem that this thesis addresses. Histogram Equalization was proposed as a potential technique for attempting to solve this problem. Other important aspects such as the objectives of this thesis, the author's contribution to knowledge and the scope and limitations of this work were also provided. The following chapter presents an in-depth discussion of many of the fundamental techniques used to construct contemporary speaker verification systems.

Chapter 2

Speaker Verification Fundamentals

The purpose of this chapter is to review a number of the techniques used to implement the various components depicted in Figure 1-1. Emphasis is placed on those techniques that are commonly employed in contemporary speaker verification systems since they will be used to create an experimental framework for evaluating Histogram Equalization. Section 2.1 provides an overview of the pre-processing techniques commonly employed before feature extraction takes place as well as several feature extraction techniques. Section 2.2 covers decision-making based on likelihood ratio testing. This strategy is frequently employed in contemporary speaker verification systems to determine whether an individual is who he (or she) claims to be. Various techniques for modelling speakers are covered in Section 2.3. Finally, Section 2.4 discusses several metrics commonly used to gauge the performance of speaker verification systems.

2.1 Feature extraction

Feature extraction is the first component encountered in both the training and test modes of a typical speaker recognition system (see Figure 1-1). As mentioned previously, the purpose of feature extraction is to extract speaker-dependent information from a raw speech signal and, in the process, convert the speech signal into a more compact and efficient representation. As the performance of the other components in a speaker recognition system (i.e., speaker modelling and decision-making) are highly dependent on the quality of the extracted features, it is imperative that these features [9, 10]:

- differentiate between speakers while being tolerant of intra-speaker variabilities (like the speaker's health or mood),
- occur naturally and frequently in the speech signal,
- are easily measurable from the speech signal,

- are minimally affected by ambient noise and transmission over communication channels,
- are stable over time, and
- are not predisposed to mimicry by impostors.

To date, no single feature extraction technique has been discovered that is able to generate features that possess all of the above-mentioned characteristics [32]. However, over the years, many feature extraction techniques have been proposed and successfully applied in speaker recognition research. These include, amongst others, *Perceptual linear predictive analysis* of speech [33, 34]; the *Ensemble interval histogram* (EIH) [35-37]; *Parameterized feature sets* [38, 39]; *Reflection coefficients* [2, 5] and; *Spectral subband centroids* [40, 41]. These feature extraction techniques generally employ some form of processing based on the human auditory system or attempt to extract features related to the speech production system. For example, the EIH feature set is based on a model that mimics the human auditory system. The model consists of a bank of cochlear filters and an array of level crossing detectors coupled with interval histograms. The cochlear filters model the frequency selectivity along various points of the basilar membrane and the level crossing detectors simulate the conversion of filterbank outputs to neural firings patterns along the auditory nerve. The cochlear filters are based on actual neural tuning curves observed for cats [35-37]. The reflection coefficients, on the other hand, model the human vocal tract as a series of cylindrical tubes with different cross-sectional areas. This results in an impedance mismatch between adjacent tubes. Thus, at each boundary a portion of the air wave is transmitted while the rest is reflected (assuming a lossless tube). The reflection coefficients are the percentage of reflection at these discontinuities [2, 5].

Many feature sets have also attempted to extract information related to the fundamental frequency (i.e., the pitch) of a particular speaker's voice [42]. This information is speaker-dependent as it depends on the length, tension and mass of a particular speaker's vocal cords (or glottis) [2]. However, such information "*can be difficult to extract reliably, especially from noise corrupted speech, and is more susceptible to non-physiological factors such as the speaker's emotional state and level of speech effort*" [11].

Although the techniques mentioned in this section have been shown to extract speaker-dependent information from speech signals; from the literature reviewed, cepstral analysis of short-time windowed segments of speech is one of the more prevalent forms of feature extraction used in contemporary speaker recognition systems [4, 5, 7, 18, 27, 29, 34, 40, 43]. This type of processing is primarily based on linear predictive coding or on filterbank analysis of speech. Cepstral analysis is discussed in detail in Section 2.1.2. The following section discusses many of the signal processing techniques commonly employed before features are extracted from a raw speech signal.

2.1.1 Speech acquisition and pre-processing

A speech signal leaves a speaker's lips and propagates across an air interface as an acoustic pressure wave [2]. This pressure wave can be captured by a microphone which converts the continuous air pressure changes into continuous voltage changes [44]. However, speaker recognition systems are generally implemented on computers or on digital signal processors. As such, the analogue speech signal needs to be converted into a digital representation before it can be processed. This process is termed *analogue-to-digital conversion* and involves band-limiting the speech signal with an anti-aliasing filter, sampling it at a fixed rate and finally representing it with finite precision by using a fixed number of bits [2, 23]. The number of bits used controls the precision with which the signal is quantised and usually varies between 8 and 16 bits. The sampling rate controls the quality with which the signal is captured and ranges from 8 kHz for telephone quality speech to 48 kHz for high quality digital audio. Subsequently, the maximum spectral frequency that can be represented is half that of the sampling rate – this is known as the Nyquist frequency [45]. For example, speech transmitted over a telephone network is sampled at 8 kHz with the result that the bandwidth of the signal is only 4 kHz.

Acquiring large quantities of speech from numerous speakers, for research purposes, is a daunting task. Fortunately, numerous speech databases have been created over the years. These databases allow researchers to compare and evaluate their speaker recognition architectures and algorithms under various conditions. Moreover, they allow for the standardisation of procedures and protocols for evaluating and comparing the performances of different systems. Many of these databases can be obtained from the Linguistic Data Consortium[4], the European Language Resources Association[5] and the Oregon Graduate Institute[6] and are reviewed in [46-48]. These databases differ mainly in [47]:

- the number and diversity of the speakers used
- the type of speech used (e.g., digits, words, sentences and spontaneous or conversational speech)
- the language(s) used
- the channel, microphone and recording environment variability
- the number of sessions recorded per speaker, as well as
- the period between session recordings.

[4] http://www.ldc.upenn.edu

[5] http://www.elra.info/

[6] http://cslu.cse.ogi.edu

The database employed in this work was used as part of the NIST 2000 speaker recognition evaluation and is discussed in detail in Section 5.1.

Once speech signals are in a digital form, they are usually passed through a series of pre-processing modules prior to feature extraction. This is aimed at making speech signals more suitable for subsequent spectral analysis. These modules are depicted in Figure 2-1 and generally consist of pre-emphasis filtering, frame-blocking, windowing and speech activity detection.

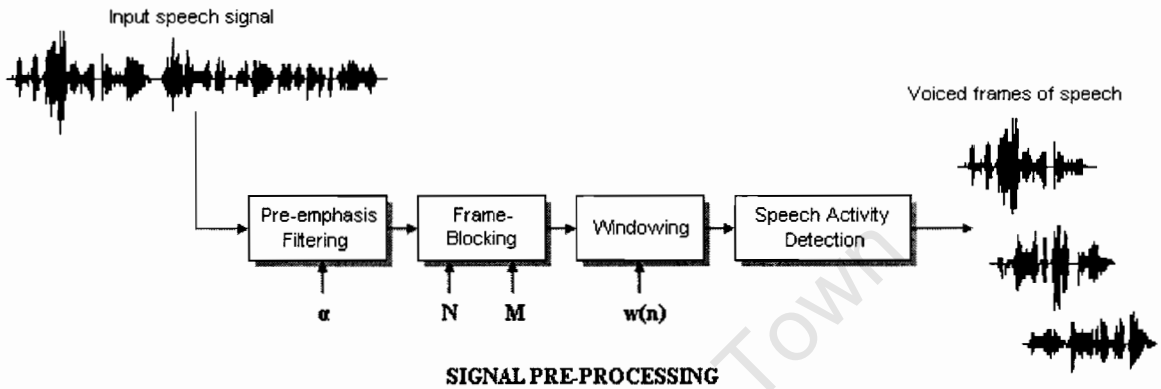

Figure 2-1: Common pre-processing techniques employed before feature extraction

The first step in signal pre-processing is that of pre-emphasizing the input speech signal. The purpose of pre-emphasis filtering is to compensate for the spectral tilt that occurs as a result of voiced[7] sounds having a steep roll-off in the high frequency region of the speech spectrum. In so doing, the distribution of energy across the frequency range of the speech signal becomes more balanced and the harmonics present in the speech signal also become more distinct [7, 36, 44, 49]. This is clearly illustrated in Figure 2-2 which shows a windowed speech signal with and without pre-emphasis filtering. The most commonly used pre-emphasis filter is given by a transfer function of the form [36, 44]:

$$H(z) = 1 - \alpha z^{-1}, \tag{2.1}$$

where α, the pre-emphasis coefficient, controls the slope of the filter and is usually a value in the interval [0.95, 0.98] [7]. In the time domain, the output of the pre-emphasis filter, $y(n)$, is related to the input speech signal, $x(n)$, as follows:

$$y(n) = x(n) - \alpha x(n-1). \tag{2.2}$$

[7] Voiced speech is produced when the vocal cords vibrate periodically as air is expelled from the lungs. The resulting speech waveform is quasi-stationary [36].

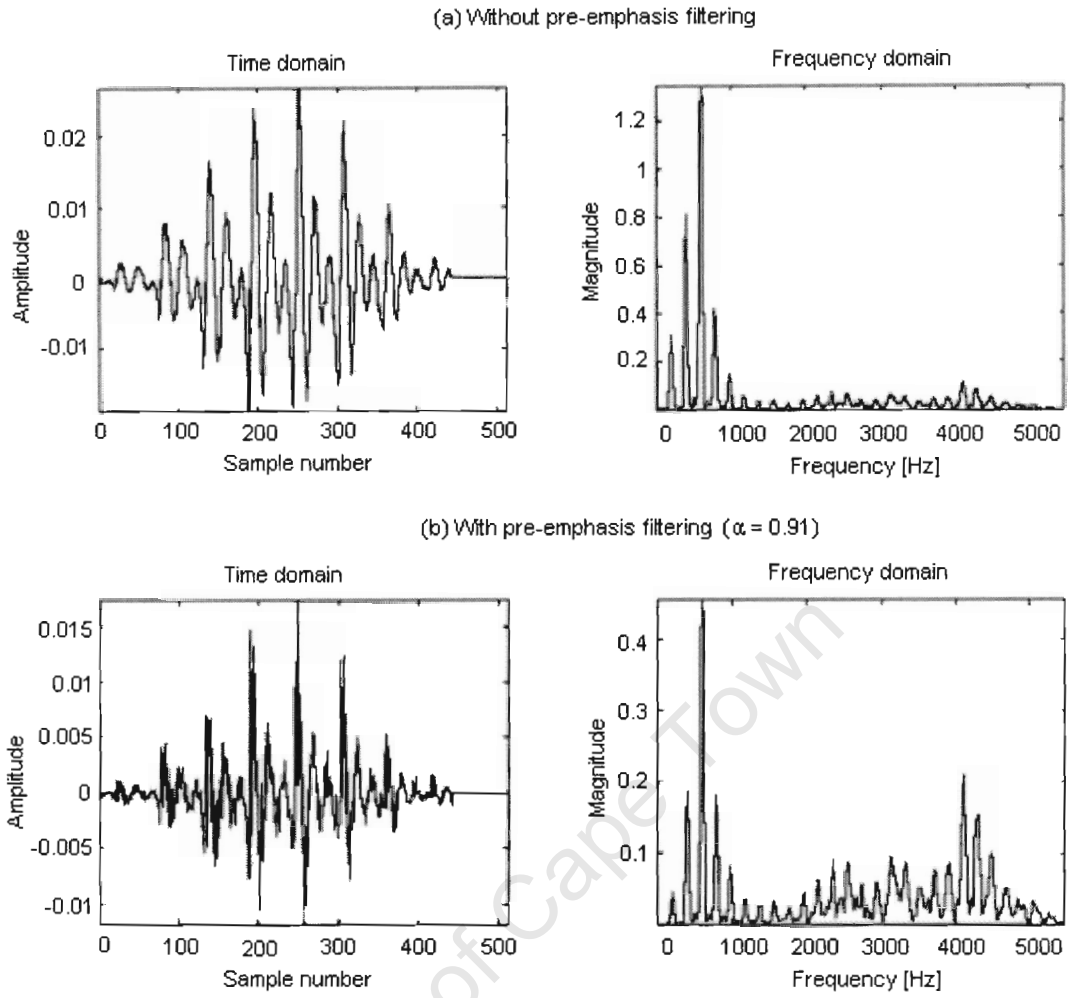

Figure 2-2: A windowed speech signal with and without pre-emphasis filtering [44]

The human speech production organs cannot move from one position to another in less than 5 milliseconds [36]. Thus, one can analyse speech signals over longer time intervals called *frames*, in which speech signals are assumed to be stationary. The process of partitioning the speech signal into frames is known as *frame-blocking* [36] and follows pre-emphasis filtering. Frame-blocking is illustrated in Figure 2-3. In this figure the speech signal is segmented into frames of N samples with adjacent frames being separated from each other by M samples. M is usually chosen to be less than N with the result that adjacent frames overlap. This process continues until the entire speech signal is accounted for in one or more frames. N is usually chosen to be 10-30 milliseconds in length while M is usually chosen to be 30-75% of the value of N [44].

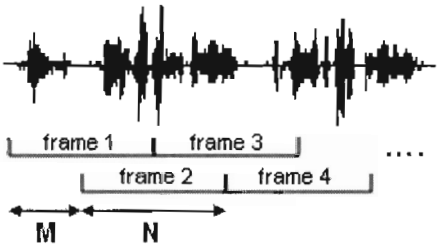

Figure 2-3: Frame-blocking - the process of segmenting a speech signal into frames

Once the speech signal has been segmented into frames, the next step in pre-processing is to window each individual frame. This process is known as *windowing* and minimises signal discontinuities at the start and at the end of each frame. These discontinuities cause spectral distortion in the frequency domain. This is primarily due to the implicit rectangular window used in frame-blocking. In the frequency domain, a rectangular window has a curved pass-band and a large amount of ripple in the stop-band [45]. For this reason, many researchers apply a *Hamming window* to each frame after frame-blocking [7, 36, 44]. This window minimises signal discontinuities by tapering the start and end of each frame to almost zero. The Hamming window has a wider pass-band and significantly less stop-band ripple than a rectangular window, and is defined as follows [44]:

$$\begin{aligned} w(n) &= 0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right), && 0 \le n \le N-1 \\ &= 0, && \text{otherwise.} \end{aligned} \tag{2.3}$$

The following diagram illustrates both the time and frequency domain representations of the rectangular and Hamming windows.

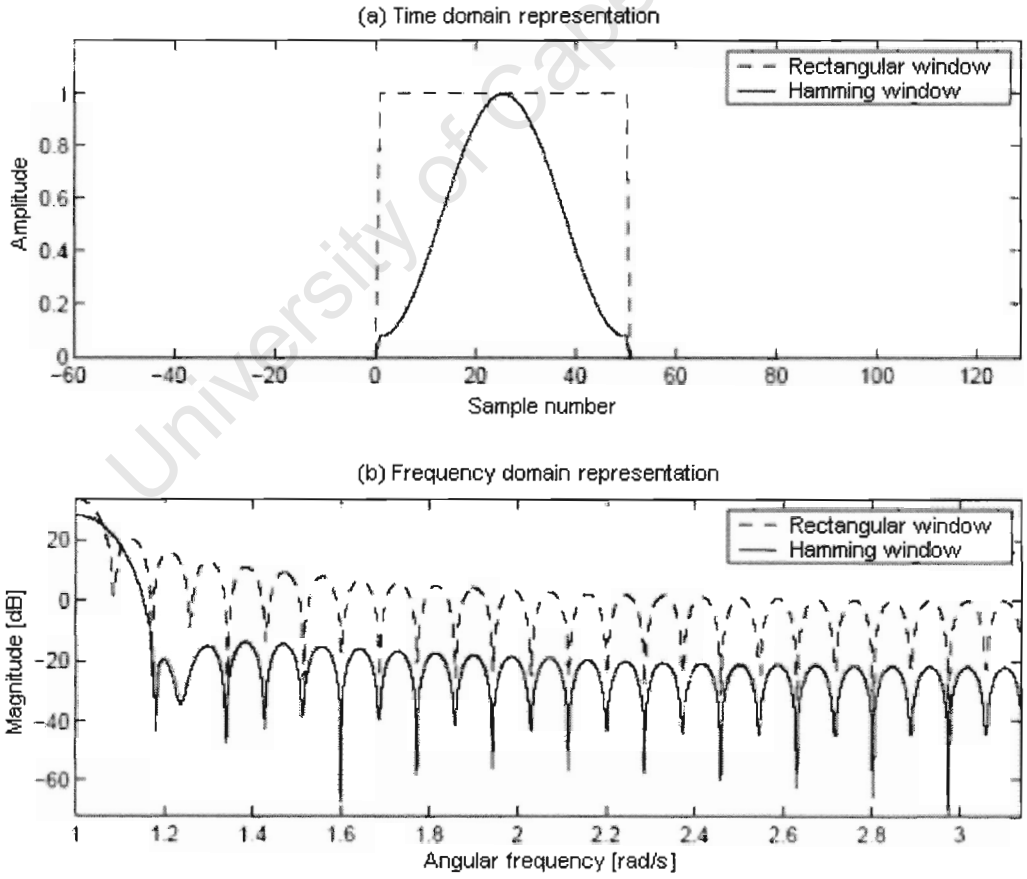

Figure 2-4: The Rectangular and Hamming windows in the time and frequency domains [44]

After windowing it is common to employ some form of *speech* (or voice) *activity detection*. The purpose of this component is to discard frames primarily containing silence, noise or unvoiced[8] speech. This is done so as to "*avoid modelling and detecting the environment rather than the speaker*" [11]. Ideally the outputs of this component are frames that contain speech only. This concludes the section on speech acquisition and pre-processing. The following section is dedicated to cepstral analysis which parameterises the voiced speech frames produced by the pre-processing module.

2.1.2 Cepstral analysis

The use of features extracted from the short-time speech spectrum has been shown to be very effective in many speaker recognition tasks [34, 38, 40]. This is because the peaks in the spectrum (which are also known as *formants*) occur as a result of the resonances of an individual's vocal tract [11, 29]. Due to the fact that the characteristics of the vocal tract (e.g., its length, width and height, its tissue density, the dimensions of the oral and nasal cavities and the size and shape of the lips teeth and tongue) differ among individuals, it is one of the main physiological factors responsible for speaker-dependent information in the speech signal [9, 11, 27, 50].

The magnitude spectrum of a speech signal can be very detailed and can contain many fluctuations. Since we are primarily interested in the locations and dimensions of the peaks in the spectrum, a smooth spectral envelope representation is more appropriate. Two techniques that are commonly used to extract such a representation are linear prediction analysis and filterbank analysis. However, these techniques result in features (linear prediction coefficients and filterbank energies respectively) that are highly correlated [23]. For this reason, these features are transformed into the cepstral domain in which the correlation between the features of the respective representations is reduced. The features are subsequently known as *cepstral coefficients* and represent the *cepstrum* of the speech signal. Properties of the cepstrum include the following:

- In the cepstral domain, attributes of the vocal tract are separated from other less informative parts of the speech signal (such as the pitch of the signal) by retaining the lower ordered cepstral coefficients [18, 51].
- Linear time-invariant filtering effects (due to telephone channels for example) appear as a constant additive bias and hence can be subtracted from the composite cepstrum, resulting in the cepstrum of the original speech signal only [2, 18, 23].
- Cepstral coefficients are well modelled by multivariate Gaussian distributions [18, 29].

[8] When speech is unvoiced, the vocal cords do not vibrate with the result that the speech signal is non-periodic and random in nature [36].

- Cepstral analysis allows for the application of simple measures to compute the distance between cepstral vectors since the cepstral coefficients from an orthogonal set [5, 51].

In the following two sections the extraction of cepstral coefficients based on linear predictive coding and filterbank analysis is discussed in more detail.

2.1.2.1 Linear Predictive Coding based cepstral coefficients [2, 5, 7, 36, 50-52][9]

Linear predictive coding (LPC) is based on the assumption that a given speech sample at time n, $s(n)$, can be approximated as a linear combination of the previous speech samples, such that:

$$s(n) \approx a_1 s(n-1) + a_2 s(n-2) + ... + a_p s(n-p), \tag{2.4}$$

where the coefficients a_1, a_2, ..., a_p are called the *LPC coefficients*, and are assumed to be constant over the speech frame under consideration. Equation (2.4) can be converted into an equality by the addition of an excitation term, $Gu(n)$, giving:

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + Gu(n), \tag{2.5}$$

where $u(n)$ is a normalized excitation and G is the gain of the excitation. The excitation can be viewed as representing the actual source of the speech signal. If the z-transform of Equation (2.5) is taken, the relation $S(z) = \sum_{k=1}^{p} a_k z^{-k} S(z) + GU(z)$ is obtained which leads to the transfer function:

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 + \sum_{k=1}^{p} a_k z^{-k}}. \tag{2.6}$$

This equation represents the linear prediction model of speech production and its interpretation is given in Figure 2-5.

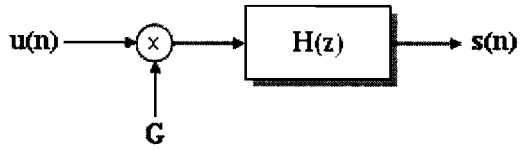

Figure 2-5: The linear prediction model of speech production [36]

In Figure 2-5 the normalized excitation source, $u(n)$, is scaled by a gain factor, G, and is fed into an all-pole system, $H(z)$, to produce the speech signal $s(n)$. $Gu(n)$ can be viewed as the glottal

[9] Much of the information in this section was taken from these references.

source (which is a pulsed air stream for voiced sounds and a random noise generator for unvoiced sounds) and $H(z)$ can be viewed as the composite transfer function representing the vocal tract, nasal tract and the lips [7, 50]. The magnitude spectrum of this transfer function represents the spectral envelope of the speech signal. In reality, $u(n)$ is generally unknown and is thus ignored [2], with the result that Equation (2.5) reduces to:

$$\hat{s}(n) = \sum_{k=1}^{p} a_k s(n-k). \tag{2.7}$$

In Equation (2.7) only speech samples *n-1* to *n-p* are used to predict the n^{th} speech sample $\hat{s}_n$. p is known as the *prediction order*. The prediction error, $e(n)$, between the actual and predicted value of the n^{th} speech sample is given by:

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k). \tag{2.8}$$

In LPC analysis, the minimisation of the mean-squared prediction error, over a short segment of speech, produces the LPC coefficients. The minimisation process results in a set of equations that can be solved using either the autocorrelation method or the covariance method. These are discussed in more detail in references [2, 36] and [50].

Many different types of features can be derived from LPC analysis. These include the LPC coefficients themselves, cepstral coefficients, reflection coefficients, log area ratios and line spectral pairs [2, 5]. According to Furui [27], "*a spectral envelope reconstructed from a truncated set of cepstral coefficients is much smoother than one reconstructed from LPC coefficients, and hence provides a stabler representation from one repetition to another of a particular speaker's utterances*". Given the p LPC coefficients, the cepstral coefficients, c_m, can efficiently be derived using the following recursive formulas [7, 36]:

$$\begin{aligned} c_0 &= \ln \sigma^2, \\ c_m &= a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, \qquad 1 \le m \le p \\ c_m &= \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, \qquad p < m \end{aligned} \tag{2.9}$$

where σ^2 is the gain term in the LPC model, a_m are the LPC coefficients, and p is the number of LPC coefficients computed. The cepstral coefficients produced by LPC analysis are known as *linear prediction cepstral coefficients* (LPCCs). In the following section, the generation of filter-bank-based cepstral coefficients is discussed.

2.1.2.2 Filterbank-based cepstral coefficients [7, 52, 53][10]

In filterbank analysis, the cepstral coefficients are directly computed using the signal processing techniques depicted in Figure 2-6.

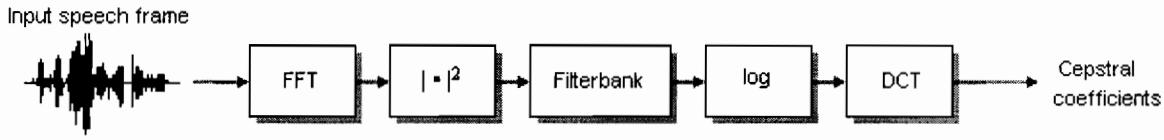

Figure 2-6: Signal processing techniques required to generate filterbank-based cepstral coefficients

In the figure above, a pre-processed frame of speech is first Fourier transformed into the frequency domain where the magnitude-squared representation of its complex Fourier spectrum is obtained. However, this representation of the frame of speech is still very detailed and contains many fluctuations. As mentioned previously, the aim of cepstral analysis is to extract the smooth spectral envelope of the spectrum such that the peaks in the spectrum become more evident. To obtain such a representation, the spectrum is filtered by a bank of band-pass filters so as to obtain the average value of the energy in a particular frequency band.

The filters are usually chosen to be triangular in shape with the start and end frequencies of each filter coinciding with the centre frequencies of the adjacent filters. In addition, the filters are often spaced on scales that relate to the human auditory system which is not equally sensitive across a linear scale [18, 36]. One such scale is the *mel scale* [36, 51, 52]. This scale is approximately linear below 1000 Hz and logarithmic above 1000 Hz, with the result that high frequencies are de-emphasised [2]. A typical mel-scaled triangular filterbank is depicted in Figure 2-7.

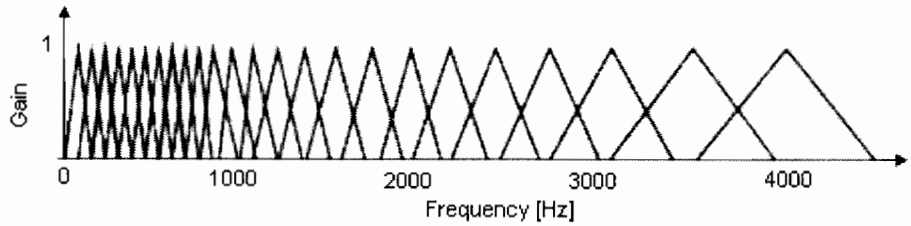

Figure 2-7: A mel-scaled triangular filterbank

[10] Much of the information in this section was taken from these references.

On the mel scale, the location of the centre frequencies of each of the filters in the filterbank is given by [54]:

$$f_{mel} = 2595.\log(1+\frac{f_{Hz}}{700}). \tag{2.10}$$

After the application of the filterbank to the spectrum, the logarithm of the filterbank energies is taken, resulting in the log-filterbank energies, $\log(E_k)$. Finally the discrete cosine transform (DCT) is applied to these values to yield the cepstral coefficients c_m [7, 36]:

$$c_m = \sum_{k=1}^{K} \log(E_k)\cos\left[m(k-1/2)\frac{\pi}{K}\right], \qquad m = 1,2,...,L \tag{2.11}$$

where K is the number of log-filterbank energies produced by the filterbank and L is number of cepstral coefficients to be computed (usually $L \leq K$). The cepstral coefficients produced by mel-scaled filterbank analysis are known as *mel-frequency cepstral coefficients* (MFCCs). Due to the DCT, the MFCC feature vectors form an orthogonal set. Furthermore, for both the filterbank- and LPC-based cepstral coefficients, the zero-th cepstral coefficient, c_0, which represents the energy in the signal is usually discarded as a form of energy normalization [34]. The following section discusses features derived from cepstral coefficients.

2.1.2.3 Cepstral derivatives

Often researchers append the first and second time derivatives of cepstral coefficients to the overall cepstral feature vector so as to capture the dynamic properties of a speech signal [4, 23, 55]. These features are often referred to as *delta* (Δc_m) and *delta-delta features* ($\Delta\Delta c_m$) respectively. While the cepstral coefficients are meant to represent the stationary properties of a speech signal (and are thus often referred to as *instantaneous* or *static features*), cepstral derivatives are meant to show how these properties vary over time (and are thus often referred to as *transitional* or *dynamic features*). The delta features are robust to linear channel distortions as they effectively remove the additive channel bias from the cepstrum of the speech signal [26, 56]. However, these features do not perform well by themselves and are usually appended to their stationary counterparts. This combination has been shown to lead to improved performance as these two feature sets contain complementary information [56]. Delta and delta-delta features can be obtained from the following formulas [7, 57]:

$$\begin{aligned} \Delta c_m &= \frac{\sum_{k=-l}^{l} k \cdot c_{m+k}}{\sum_{k=-l}^{l} |k|}, \\ \Delta\Delta c_m &= \frac{\sum_{k=-l}^{l} k^2 \cdot c_{m+k}}{\sum_{k=-l}^{l} k^2}. \end{aligned} \tag{2.12}$$

This concludes the section on feature extraction. In the experimental work done in this study, filterbank-based cepstral coefficients (i.e., MFCCs) are used as they have been shown to perform reasonably well on numerous speaker recognition tasks [4, 11, 29] and are commonly employed in contemporary speaker verification systems. In addition, they were shown to be relatively more robust than LPC-based cepstral coefficients in the empirical studies done in [34] and [58]. As a point of interest, many of the feature extraction techniques discussed in this chapter have successfully been applied in speech recognition as well [58] which suggests that the feature extraction process generally retains both linguistic and speaker-dependent information. Although cepstral coefficients have been shown to perform extremely well on clean speech, they are not robust to background noise [58] and the effects of telephone channels [17]. The work done in this study is to some extent aimed at making these features more robust to such degradations. The following section deals with decision-making in speaker verification.

2.2 Decision-making

Decision-making in speaker verification refers to the task of deciding whether a particular speaker is who he (or she) claims to be. Thus, given a segment of speech, Y, and a targeted speaker (i.e., the speaker associated with the input identity claim), a speaker verification system must determine whether Y was indeed spoken by the targeted speaker. In order to make such a decision, the speaker verification task is often restated as a basic hypothesis test [7, 11, 59, 60][11] where the system needs to decide between two hypotheses:

H_0: Y is from the targeted speaker or

H_1: Y is not from the targeted speaker (i.e., Y is from an impostor).

According to references [23] and [60], the optimal test to decide between these two hypotheses is a *likelihood ratio test* given by:

$$\left.\begin{array}{l} \dfrac{p(Y \mid H_0)}{p(Y \mid H_1)} \end{array}\right\} \begin{array}{ll} \geq \theta & \text{accept } H_0 \\ < \theta & \text{reject } H_0 \end{array}$$

where $p(Y \mid H_i)$, i = 0,1, is referred to as the *likelihood* of the hypothesis H_i given the speech segment Y and, θ is referred to as the decision threshold which determines whether H_0 is accepted or not. If the value of the likelihood ratio is greater than or equal to θ, H_0 is accepted, otherwise H_0 is rejected. This decision-making process is commonly employed in contemporary speaker verification systems.

[11] Much of the information in this section was taken from these references.

From Section 2.1, we know that the speech segment Y can be represented in the feature space by a sequence of feature vectors $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$. Furthermore, Section 2.3 will show that, from a mathematical point of view, H_0 can be represented by a speaker model, λ_{hyp}, that characterises the targeted speaker in the feature space. The alternative hypothesis, H_1, can likewise be represented by a *background* (or *anti-*) *speaker model*, $\lambda_{\overline{hyp}}$, that characterises speakers other than the targeted speaker in the feature space. The likelihood ratio test is then given by $p(X \mid \lambda_{hyp}) / p(X \mid \lambda_{\overline{hyp}})$ which, after taking logarithms, becomes:

$$\Lambda(X) = \log p(X \mid \lambda_{hyp}) - \log p(X \mid \lambda_{\overline{hyp}}). \tag{2.13}$$

$\Lambda(X)$ is termed the *log-likelihood ratio*. From the analysis presented here, the decision-making process based on a likelihood ratio test can be depicted as follows:

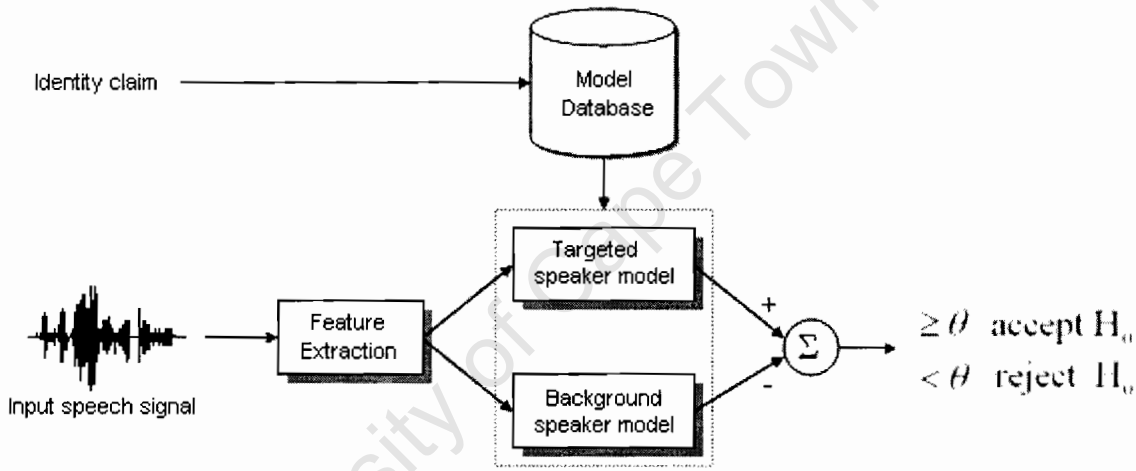

Figure 2-8: Decision-making process based on a likelihood ratio test

"*The absolute likelihood score of an utterance from a speaker model is influenced by many utterance-dependent factors including the speaker's vocal characteristics, the linguistic content and the speech quality*" [59]. These factors will affect both the score obtained for the targeted speaker model as well as the score obtained for the background speaker model. However, the difference between these two scores produces a relative score that is more a function of the speaker of the utterance and less susceptible to non-speaker related variations [59]. This allows for the application of more stable decision thresholds. For this reason, the log-likelihood ratio given by Equation (2.13) can be viewed as a form of score normalization since it helps to minimise non-speaker-related variations in the scores obtained during testing.

An important step in the implementation of the likelihood ratio speaker verification system is the selection of which likelihood function, $p(X \mid \lambda)$, to use. For text-independent speaker recognition

"*the most successful likelihood function has been Gaussian mixture models*" [60]. For this reason speaker modelling, with particular emphasis on Gaussian mixture models, is discussed next.

2.3 Speaker modelling

Section 2.1 described several ways in which a speech signal can be converted into a sequence of feature vectors. Once this is done, a model of the speech characteristics of each speaker must be created. Effectively modelling the speech characteristics of an individual is a crucial step in obtaining good speaker recognition performance. As mentioned previously, it is desirable that speaker models [3]:

- be based on sound theoretical principles (so that model behaviour can easily be understood and so that extensions and improvements can be approached from a mathematical point of view),
- generalise well to unseen data (i.e., the model should not overfit the training data leading to poor performance during testing), and
- be economical as far as memory and computational resources are concerned.

Over the years many modelling techniques that exhibit some or all of these properties have been developed and applied in speaker recognition research. These techniques include, amongst others, *Template matching by dynamic time warping* [3, 9, 50]; *Hidden Markov models* (HMM) [5, 9, 50]; *Artificial Neural networks* [3, 5, 29]; *Vector Quantization* [2, 5, 50] and; *Support Vector machines* (SVM) [7, 61, 62]. These techniques differ mainly in their storage and computational requirements; the type of modelling strategy employed: discriminative or generative[12]; the architecture of the speaker models: parametric or non-parametric[13]; the manner in which the speaker models are trained: supervised or unsupervised[14]; and in the ability of the technique to handle the temporal nature of speech signals.

[12] Generative modelling attempts to capture all the underlying fluctuations and variations of the data for a particular class whereas, discriminative modelling tries to model the decision boundary between classes and ignores the fluctuations within each class [62].

[13] Parametric models assume a structure that is characterised by a collection of parameters. On the other hand, non-parametric models make minimal assumptions regarding the distribution of the data [18].

[14] Supervised training requires all the training data to be labelled with their true class identity. Alternatively, unsupervised training refers to situations in which decision boundaries are based on (or models are trained using) unlabeled data [8, 50].

For example, Hidden Markov models [5, 9, 50] allow one to build probabilistic models that describe both the stationary and time-varying properties of the training speech for each speaker. Thus, it can be considered to be a generative modelling strategy. Each HMM consists of an underlying stochastic process that is not directly observable (hence the term hidden). It can however be observed through another stochastic process that produces a sequence of observations (or output symbols). The basic structure of a HMM is a set of states with transitions between states. At discrete time intervals, the system passes from one state to another with each state producing an output. The transitions between states as well as the outputs associated with each state are probabilistic. In so doing, the model accommodates the temporal variations present in a particular speaker's training speech.

A support vector machine [7, 61, 62], on the other hand, is a discriminative binary classifier that is aimed at separating complex regions between two classes of data. It does this by projecting the data into a higher dimensional feature space (by means of a kernel function) where a separating hyperplane is found by maximizing the margin between the two classes. When the data is projected back to the original feature space, the hyperplane forms a non-linear decision boundary. According to reference [7], the main problems encountered when using SVMs are the search for the appropriate kernel function for a particular application and their "*inappropriateness to handle the temporal structure of speech signals*".

All the speaker modelling techniques mentioned previously have successfully been applied in speaker recognition research. However, from the literature surveyed, modelling speakers using *Gaussian mixture models* (GMMs) [26, 29] is one of the more frequently used techniques in speaker recognition research [4, 7, 28, 35, 38, 40, 55, 59, 60, 63]. This is primarily due to the fact that it exhibits the desirable attributes mentioned at the start of this section, and has "*modest computational requirements and consistently high performance*" [4]. This speaker modelling technique is used in the experimental work done in this thesis and is discussed in more detail in the following sections.

2.3.1 Speaker modelling using GMMs [11, 26, 29, 59][15]

"*The advantages of using a GMM as the likelihood function are that it is computationally inexpensive, is based on a well-understood statistical model, and, for text-independent tasks, is insensitive to the temporal aspects of the speech, modelling only the underlying distribution of acoustic observations from a speaker*" [60]. When using GMMs, the distribution of feature vectors ex-

[15] Much of the information in this section was taken from these references.

tracted from the training speech for a particular speaker is modelled as a Gaussian mixture density. A Gaussian mixture density is weighted sum of M component unimodal Gaussian densities (also known as mixtures) and is given by:

$$p(\mathbf{x}\,|\,\lambda) = \sum_{i=1}^{M} w_i p_i(\mathbf{x}), \tag{2.14}$$

where $\mathbf{x}$ is a D-dimensional feature vector, $p_i(\mathbf{x})$, $i = 1,\ldots, M$ are the component densities and w_i, $i = 1,\ldots, M$ are the mixture weights. Each component density is a D-dimensional Gaussian function of the form:

$$p_i(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}\,|\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\,\Sigma_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right\}, \tag{2.15}$$

with mean vector $\boldsymbol{\mu}_i$ and covariance matrix Σ_i. $(\cdot)^{\mathrm{T}}$ is the transpose operator. The mixture weights satisfy the constraint $\sum_{i=1}^{M} w_i = 1$, which ensures that the mixture is a true probability density function. A complete GMM is parameterised by the mean vectors, covariance matrices and mixture weights from all its component densities and is collectively represented by the notation:

$$\lambda = \{w_i, \boldsymbol{\mu}_i, \Sigma_i\} \quad i = 1,\ldots, M.$$

These parameters are estimated from a speaker's training data with maximum likelihood estimates of the model parameters being obtained using the *Expectation-Maximization* (EM) algorithm [26, 64]. The general form of the GMM supports full covariance matrices. However, according to Bimbot et al. [7], GMMs with diagonal covariance matrices are usually used as the density modelling of an M-th order full covariance GMM can equally well be achieved using a larger order diagonal covariance GMM; and GMMs with diagonal covariance matrices are more computationally efficient than GMMs with full covariance matrices as the need to repeatedly invert DxD matrices is eliminated and the number of parameters to compute is reduced.

In speaker recognition, each speaker is represented by a GMM and is referred to by his (or her) model λ. The average log-likelihood of a sequence of feature vectors, $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$, given a speaker model, λ_s, is given by [59, 60]:

$$\log p(X\,|\,\lambda_s) = \frac{1}{T}\sum_{t=1}^{T} \log p(\mathbf{x}_t\,|\,\lambda_s), \tag{2.16}$$

where $p(\mathbf{x}_t \mid \lambda_s)$ is computed using Equation (2.14) and the sequence of feature vectors are assumed to be independent. According to Reynolds [29], one of the principle motivations for using a mixture of Gaussian densities to model a speaker is that such a model has the ability to form smooth approximations to arbitrarily shaped densities. Gaussian mixture modelling can thus be viewed as a generative speaker modelling technique.

Due to the discussion presented in Section 2.2, speaker verification systems do not only require a model describing the targeted speaker but, a model describing all other speakers (i.e., the background speakers) as well. Various ways of obtaining such a model are discussed in the following section.

2.3.2 Background speaker modelling

With reference to Section 2.2, the model for the hypothesis H_0, λ_{hyp}, is well-defined and can be estimated from the training data of the targeted speaker. However, the model for the alternative hypothesis H_1, $\lambda_{\overline{hyp}}$, is not as well-defined as it must potentially represent the complete set of speakers other than the targeted speaker. Evaluating this set of speakers is infeasible due the large amount of computational and storage resources that would be required. For this reason, two main approaches have been taken to approximate the model for the alternative hypothesis [7, 23, 60]:

- In the first approach, a set of individual speaker models are trained and are then collectively used to represent $\lambda_{\overline{hyp}}$. This set of speakers is often referred to in various contexts as *cohorts* [65] and *background speaker sets* [11]. More formally, given a set of N speaker models $\{\lambda_1,\ldots,\lambda_N\}$, the likelihood of the alternative hypothesis is given by

$$p(X \mid \lambda_{\overline{hyp}}) = f\left[p(X \mid \lambda_1),\ldots, p(X \mid \lambda_N)\right], \tag{2.17}$$

  where $f(\cdot)$ is some function, such as the maximum or average [63], of the likelihood values for the set of speaker models. This set of speakers should ideally be selected to represent the population of expected impostors for a particular verification task and should be as large as possible to better model the impostor population. However, practical considerations of computation time and storage requirements dictate a small set of background speakers [59]. According to references [7] and [60], it has been found that when using this approach, best performance is obtained when using speaker-dependent background speaker sets. However, this can become impractical in applications where there are numerous enrolled speakers since each speaker will require his (or her) own background speaker set.

- In the second approach, a collection of speech from several speakers is used to train a single model to represent the alternative hypothesis. The main advantage of this approach, over the first approach, is that here a single speaker-independent model need only be trained once for a particular verification task and can then be used for all targeted speakers in that task. This single model has been referred to as a *world model* [23] and a *universal background model* (UBM) [63]. The aim is to represent the speaker independent

distribution of features in the feature space. As such, a collection of speech that is "*reflective of the expected alternative speech to be encountered during recognition*" [60] should be used. This applies to both the type and quality of the speech, as well as the composition of the speakers. For example, in [66] and [67] gender- and handset-dependent models were employed.

In this work, the second approach is used to model speakers other than the targeted speakers since the amount of memory required to store the parameters of the single model is much less than that required to store the models of multiple background speakers. Moreover, recognition is more computationally efficient, since only one background speaker log-likelihood need be computed for each targeted speaker. The single model is hereafter referred to as a universal background model (UBM).

In [63], Reynolds compared two approaches of background speaker modelling for a text-independent speaker verification task using Gaussian mixture models. In particular, he compared the use of a speaker-dependent background speaker set to that of a speaker-independent universal background model. For the UBM, it was described how Bayesian adaptation could be used to derive models for the targeted speakers, thereby providing a structure which led to significant computational savings. Experiments conducted on the NIST 1996 speaker recognition evaluation database showed that a system using a UBM and Bayesian adaptation of speaker models produced superior performance compared to one employing speaker-dependent background sets and another employing a UBM with independent models for targeted speakers. Moreover, in [4], Doddington and his colleagues provided and an overview of the annual NIST speaker recognition evaluations and remarked that "*Gaussian mixture models, especially adapted GMMs, were the models most often used primarily due to their modest computational requirements and consistently high performance*". For these reasons, the same approach is adopted in this study and is explained in the following section.

2.3.3 Speaker modelling using Adapted GMMs

Instead of directly using the Gaussian mixture modelling strategy discussed in Section 2.3.1, the model of a targeted speaker can also be obtained by adapting the parameters of a UBM by using the speaker's training data and a form of Bayesian adaptation [7, 60, 63][16]. Here, as opposed to using the standard approach of obtaining a speaker model by maximum likelihood training independent of whether a UBM exists or not, the adaptation approach seeks to derive a speaker model

[16] Much of the information in this section was taken from these references.

by "updating" the well-trained parameters of a UBM via adaptation. This leads to a tighter coupling between the speaker model and the UBM which (1) produces better performance than decoupled models [63] and (2) leads to a fast procedure for calculating the log-likelihood ratio given by Equation (2.13).

Adaptation is a two step process [7, 60, 63]:

- In the first step, estimates of the sufficient statistics[17] of a particular speaker's training data are computed for each mixture in the UBM[18]. This is done as follows. Given a UBM with M mixtures and T training feature vectors, $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$, from a targeted speaker, the probabilistic alignment of the training feature vectors into the UBM mixture components is obtained. That is, for the i^{th} mixture in the UBM we compute:

$$\Pr(i \mid \mathbf{x}_t) = \frac{w_i p_i(\mathbf{x}_t)}{\sum_{j=1}^{M} w_j p_j(\mathbf{x}_t)}, \tag{2.18}$$

  where w_i is the mixture weight for the i^{th} mixture and $p_i(\mathbf{x}_t)$ is the multi-dimensional probability density function representing the i^{th} mixture of the UBM (see Equation (2.15)). $\Pr(i \mid \mathbf{x}_t)$ and $\mathbf{x}_t$ are then used to compute the sufficient statistics for the weight, mean and variance parameters[19]:

$$\begin{aligned} n_i &= \sum_{t=1}^{T} \Pr(i \mid \mathbf{x}_t), \\ E_i(\mathbf{x}) &= \frac{1}{n_i} \sum_{t=1}^{T} \Pr(i \mid \mathbf{x}_t) \mathbf{x}_t, \\ E_i(\mathbf{x}^2) &= \frac{1}{n_i} \sum_{t=1}^{T} \Pr(i \mid \mathbf{x}_t) \mathbf{x}_t^2. \end{aligned} \tag{2.19}$$

[17] These are the basic statistics that are required to compute the adapted GMM parameters. For each mixture, these are the count and first and second moments required to compute the mixture weight, mean and variance [60].

[18] For the work done in this thesis, a UBM can be viewed as a Gaussian mixture model with a large number of mixtures that has been trained (on a development set that is not part of either training or test sets) to represent the speaker independent distribution of feature vectors in the feature space.

[19] $\mathbf{x}^2$ is shorthand for the diagonal of the matrix given by $\text{diag}(\mathbf{x}.\mathbf{x}^T)$

- In the second step, the estimates of the sufficient statistics are combined with the old sufficient statistics from the UBM parameters. The combination is controlled by a data-dependent mixing coefficient α_i. This parameter is designed so that mixtures with high probabilistic counts of data for the targeted speaker rely on more of the new sufficient statistics for final estimation of the adapted GMM parameters. Alternatively, mixtures with low counts of data from the speaker rely more on the old sufficient statistics for final parameter estimation. Given the new sufficient statistics estimated from the training data of a targeted speaker, the old UBM sufficient statistics for the i^{th} mixture are updated as follows:

$$\begin{aligned} \hat{w}_i &= \left[\alpha_i n_i / T + (1-\alpha_i) w_i\right]\gamma, \\ \hat{\boldsymbol{\mu}}_i &= \alpha_i E_i(\mathbf{x}) + (1-\alpha_i)\boldsymbol{\mu}_i, \\ \hat{\sigma}_i^2 &= \alpha_i E_i(\mathbf{x}^2) + (1-\alpha_i)(\sigma_i^2 + \boldsymbol{\mu}_i) - \hat{\boldsymbol{\mu}}_i. \end{aligned} \tag{2.20}$$

The parameters $\hat{w}_i$, $\hat{\boldsymbol{\mu}}_i$ and $\hat{\sigma}_i^2$ are the parameters of the i^{th} mixture of the adapted GMM for the targeted speaker. The scale factor, γ, is computed over all adapted mixture weights to ensure that they sum to unity and the data-dependent mixing coefficient α_i in the above equations is given by:

$$\alpha_i = \frac{n_i}{n_i + r}, \tag{2.21}$$

where is r a fixed relevance factor that is usually chosen to be in the range 8 to 20. Note that if a certain mixture has a low probabilistic count, n_i, for a targeted speaker's training data, then $\alpha_i \rightarrow 0$ causing the "*de-emphasis of the new (potentially under-trained) parameters and the emphasis of the old (better trained) parameters*" [60]. The opposite occurs for mixtures with high probabilistic counts. For this reason references [7], [60] and [63] state that this approach should also be robust to limited amounts of training data.

As mentioned previously, the use of speaker models adapted from a UBM leads to a fast procedure for computing the log-likelihood ratio given by Equation (2.13). This equation requires one to compute both the log-likelihood for the targeted speaker model as well as the log-likelihood for the background speaker model in order to obtain the log-likelihood ratio for a particular verification trial. However, due to two observations made by the references [7], [60] and [63], one can take advantage of the fact that a targeted speaker model was obtained by adapting the parameters of a UBM. The first observation is that when a large GMM (i.e., one with many mixtures) is evaluated for a feature vector, only a few of the mixtures contribute significantly to the overall likelihood value. This is due to the fact the mixtures of a GMM represent a distribution over a

large space. Thus, a single feature vector will only be close to a few of the mixture components. For this reason, the overall likelihood value can be well-approximated using only the C highest scoring mixtures. Secondly, the components of an adapted GMM retain a correspondence with the mixtures of the UBM such that, if a feature vector is close to a particular mixture in the UBM, it is also likely to be close to the corresponding mixture in the adapted speaker model. Using these two observations, a fast scoring procedure can be implemented as follows [7, 60, 63]:

1. For each feature vector, find the C highest scoring mixtures in the UBM (with M mixtures and $C \ll M$) and estimate the UBM log-likelihood using only the scores obtained for these mixtures.
2. Next, score the same feature vector against only the corresponding C mixtures in the adapted speaker model to obtain an estimate of the log-likelihood for the adapted speaker model.

Thus, each feature vector is only evaluated against C of the M mixtures of a targeted speaker model. When large model orders are used and multiple targeted speaker models need to be evaluated for each test segment, the computational savings become significant. Typically a value of $C = 5$ is used. In the following section several metrics commonly used to gauge the performance of speaker verification systems will be presented.

2.4 Performance measures

"*Performance measures serve a number of purposes. These include, most importantly, a means for evaluating research ideas and making consistent long-term technical progress. Other reasons include comparing different systems, evaluating the effectiveness of technology for specific applications, marketing research to sponsors and selling products to customers*" [4]. Thus, this section provides an overview of the most prevalent performance measures used in speaker verification research.

2.4.1 The FAR, FRR, EER, ROC, DET and DCF

Recall that a speaker verification system needs to make a binary decision: i.e., it needs to either accept or reject the current identity claim. As such, it can make two types of errors: i.e., it can either falsely accept impostors or falsely reject legitimate speakers [23][20]. "*Both of these types of error depend on the decision threshold used in the decision making process*" [7]. If the threshold

[20] Much of the information in this section was taken from this reference.

is too low, the system will accept the majority of the identity claims, thus making few false rejections but many false acceptances. Alternatively, if the threshold too high, the system will reject the majority of the identity claims, thus making few false acceptances but many false rejections. The probability of accepting a speaker given that he (or she) is an impostor is termed the *false accept rate* (FAR, or the *false alarm probability*) and is given by:

$$\text{FAR}\,[\%] = 100 \cdot \frac{N_{\text{FA}}}{N_{\text{I}}}, \tag{2.22}$$

where N_{I} is the number of impostor trials (or access attempts) and N_{FA} is the number of those where the impostor was falsely accepted. Similarly, the probability of rejecting a speaker given that he (or she) is indeed a legitimate speaker is termed the *false reject rate* (FRR, or the *miss probability*) and is given by:

$$\text{FRR}\,[\%] = 100 \cdot \frac{N_{\text{FR}}}{N_{\text{L}}}, \tag{2.23}$$

where N_{L} is the number of legitimate speaker trials and N_{FR} is the number of those where a legitimate speaker was falsely rejected. Figure 2-9 shows a typical plot of the false accept rate and the false reject rate as the decision threshold is varied. This diagram clearly illustrates that the FAR can only be decreased at the expense of an increase in the FRR and vice versa. Depending on the application, more emphasis may be placed on one error over the other. For example, in a high security environment, it may be desired to have the FAR as low as possible, even at the expense of a high FRR. On the other hand, in forensic applications it may be acceptable to have a high FAR to prevent excluding probable suspects; that is, to achieve a low FRR. In addition, the point at which the two curves in Figure 2-9 intersect (i.e., where FAR = FRR) is known as the *equal error rate* (EER) and is often used as a single performance indicator for these two types of error. The FAR and the corresponding FRR are collectively referred to as the *operating point* of a speaker verification system [7].

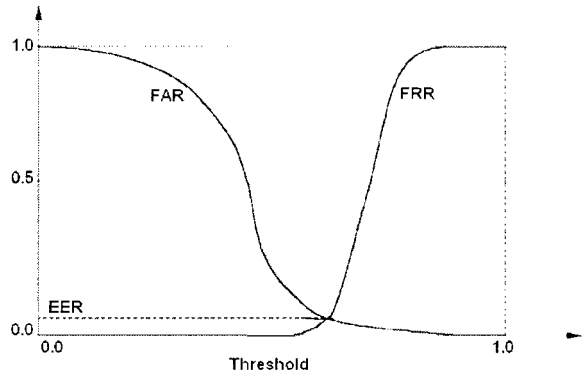

Figure 2-9: FAR and FRR as the decision threshold is varied

To compare the performance of two or more speaker verification systems, *Receiver Operating Characteristic* (ROC) or *Detection Error Trade-off* (DET) curves [4, 7, 68] are often used. Both of these curves plot the FAR versus the FRR but, on different scales – the ROC curve uses a linear scale, whereas the DET curve uses a normal deviate scale. The better the system, the closer these two curves will be to the origin. DET curves, however, are more commonly used than ROC curves since marginal differences in the performance of competing systems are visibly more evident (see Figure 2-10). Furthermore, the DET curve exhibits linear behaviour when the impostor and target score distributions are Gaussian. Figure 2-10 shows an example of two DET curves from competing systems with the corresponding ROC curves adjacent to it. The EER points are depicted by squares on both these plots.

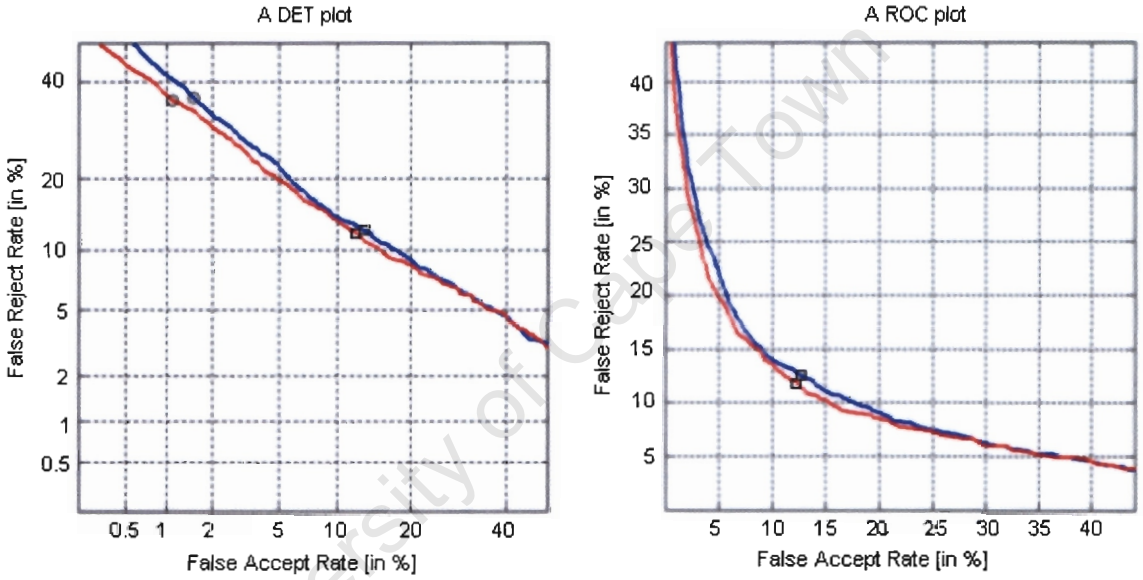

Figure 2-10: A DET plot and the corresponding ROC plot

According to reference [23], the EER performance measure is unsuitable for field trials and realistic applications of speaker verification as these systems do not necessarily operate at the EER point. Furthermore, the "*equal error rate is not an operational criterion, because it does not involve a priori threshold setting; the equal error rate threshold can only be determined after all access attempts have been processed*" [4, 23]. An operational criterion that can however be used to gauge the performance of a speaker verification system, is the total *cost* of the errors made by the system.

One such criterion is the *detection cost function* (DCF) [4] which is a linear combination of the false reject and false accept rates and is given by:

$$DCF = C_{FR} \cdot \text{FRR} \cdot P_L + C_{FA} \cdot \text{FAR} \cdot P_I \,, \tag{2.24}$$

where C_{FR} is the cost of a false reject, C_{FA} is the cost of a false accept and, P_L is the prior probability that a legitimate speaker will use the system while P_I is the prior probability that an impostor will use the system. Doddington et al. [69] states that "*this measure has the advantage that it models the application and produces a number, which is directly meaningful to the application*". Thus, the setting of the parameters in Equation (2.24) depends on the application. For example, when controlling access to a financial application that allows money to be transferred to third party accounts, the cost of a false accept will be high and P_I will be much less than 1. For the detection cost function, the minimum value over all operating points is usually computed and reported on. In Figure 2-10, the minimum DCF points are denoted by circles in the upper left portion of the DET plot. In this study, both the equal error rate and minimum value of the detection cost function are used as performance measures. It should be noted, that when measuring performance, computational time and memory requirements should also be taken into consideration.

2.4.2 Statistical significance

When enhancing a speaker verification system, it is important to know whether any apparent improvement in the performance of the system is statistically discernible. When using the equal error rate discussed in the previous section to gauge system performance, it is not sufficient to say that system A is "better" than system B if $EER_A < EER_B$, especially when the difference between the two measures is extremely small. For this reason, *McNemar's test* [70-73] was used to determine whether the apparent difference in the performance of two algorithms is indeed statistically significant. McNemar's test follows from the fact that the joint performance of two algorithms, A_1 and A_2, can be summarised by a 2x2 contingency table as follows:

		A_2	
		Correct	Incorrect
A_1	Correct	N_{00}	N_{01}
	Incorrect	N_{10}	N_{11}

where: N_{00} is the number of speakers that both A_1 and A_2 classify correctly,

N_{01} is the number of speakers correctly classified by A_1 but, incorrectly classified by A_2,

N_{11} is the number of speakers that both A_1 and A_2 classify incorrectly and,

N_{10} is the number of speakers incorrectly classified by A_1 but, correctly classified by A_2.

The total number of speakers, N, in the test set is equal to $N_{00} + N_{01} + N_{10} + N_{11}$. The null hypothesis is that the two algorithms A_1 and A_2 have the same error rate, i.e.:

$$H_0 : N_{01} = N_{10}. \tag{2.25}$$

"*McNemar's test is based on a χ^2-test for goodness-of-fit that compares the distribution of counts expected under the null hypothesis to the observed counts*" [71]. Under the null hypothesis, the expected counts are:

N_{00}	$(N_{01} + N_{10})/2$
$(N_{01} + N_{10})/2$	N_{11}

McNemar's value, M, is then calculated as follows:

$$M = \frac{(|N_{01} - N_{10}| - 1)^2}{N_{01} + N_{10}}, \tag{2.26}$$

and is distributed as χ^2 with 1 degree of freedom. Equation (2.26) incorporates a *continuity correction* term (of -1 in the numerator) to account for the fact that the statistic is discrete while the χ^2 distribution is continuous [71]. The χ^2 critical value with a 5% level of significance, is written as $\chi^2_{(1,0.95)}$ and is equal to 3.841459. In general, if M is greater than 3.841459, the null hypothesis is rejected and the difference in the performance of the two algorithms can be considered to be statistically significant. In other words, the observed difference would arise by chance on less than 5% of occasions. McNemar's test was previously applied in speaker recognition research in [74] and [75].

2.5 Summary

This chapter provided an in-depth review of the fundamental techniques and modules required to build a contemporary speaker verification system. Feature extraction with particular emphasis on cepstral analysis and speaker modelling with particular emphasis on Gaussian mixture models was covered. Furthermore, decision-making based on a likelihood ratio test was presented. This test requires a model of the targeted speaker as well as a model of all speakers other than the targeted speaker to make a decision. The methodology for obtaining speaker models from a universal background model by means of Bayesian adaptation was also provided. Many of the techniques and concepts discussed in this chapter are used in Chapter 5 to develop an experimental framework for evaluating HEQ and thus, will become more pertinent as this document progresses. The following chapter is aimed at providing some insight into the reasons for speaker verification

systems performing poorly in adverse environments, as well as a review of several techniques that have been used to improve the robustness of such systems.

Chapter 3

Techniques for Robust Speaker Verification

In the vast body of speaker recognition literature, numerous compensation techniques have been proposed to improve the performance and robustness of speaker recognition systems in adverse environments. These techniques can broadly be categorised as either being signal-based, feature-based, model-based, score-based or fusion-based. This thesis proposes a feature-based compensation technique that is aimed at normalizing feature distributions so as to minimise the mismatch between training and test conditions. This technique is also compared to other feature-based compensation techniques in Chapter 6. Feature-based compensation techniques are discussed in more detail in Section 3.2. Score normalization is used in the experimental work done in this study. For this reason, score-based compensation techniques are also discussed in more detail in Section 3.3. These techniques are primarily aimed at normalizing score distributions so as to allow for the setting of stable speaker-independent decision thresholds and to make speaker verification systems more robust to mismatched conditions. For completeness, signal-based, model-based and fusion-based compensation techniques are briefly discussed next.

Signal-based compensation techniques[21] are aimed at suppressing the effects of additive noise sources (e.g., computer hum, car engine noise, door slams, keyboard clicks, traffic noise, music and background babble) on a raw speech signal [76]. In so doing, these techniques enhance the speech content and improve the quality of the speech signal at various signal-to-noise ratios. Both single-channel (e.g., Spectral Subtraction [77]) and multi-channel speech enhancement techniques (e.g., microphone arrays [78]) exist, and are usually applied as a pre-processing stage before feature extraction.

[21] These techniques are also commonly referred to as speech enhancement techniques.

Model-based compensation techniques generally transform or adapt model parameters so as to make speaker models more robust to mismatched conditions. For example, in [28], Murthy et al. proposed a model-based channel compensation technique that was aimed at rendering Gaussian mixture speaker models more robust to channel mismatches. This was done by artificially increasing the variances of the component densities while leaving their means unchanged. In so doing, each speaker model occupied a larger portion of the feature space. The model transformation approach was meant to account for the unknown modification in the means and variances of GMMs which occurred as a result of extracting features from speech obtained from different telephone lines. It was shown to improve performance on a speaker identification task involving speech obtained from different telephone lines and handsets. In [26], Reynolds proposed a model-based noise compensation technique for representing speakers in adverse environments. Here, robustness was achieved by integrating into the model of each speaker a model describing the noise contaminating the speech signal. The composite model was applied to a speaker identification task using noise corrupted speech and was shown to be more robust to mismatched noisy environments than independent speaker models.

Fusion-based compensation techniques are aimed at combining the scores obtained from evaluating different speaker models trained for the same speaker. These models usually incorporate some form of diversity as they are either trained with different utterances from the same speaker, different features extracted from the same speech signal or different modelling strategies. Ultimately, it is desired that the models exhibit uncorrelated behaviour (i.e., they misclassify different speakers). In so doing, performance can be improved as the errors made for one model can be rectified by correct decisions made for the other models and vice versa. Various techniques for combining the scores produced by different models are discussed in [50]. However, when working with fusion-based systems, one must accept a trade-off in memory and computational resources as additional models need to be created, stored and evaluated. An example of where a fusion-based system has been used for speaker recognition can be found in [79]. Here, separate speaker models were trained on features related to the physical characteristics of an individual's vocal tract and features related to an individual's learned manner of speaking (i.e., his (or her) word usage and idiolect). A fusion of the scores obtained for these models led to large improvements in the performance and robustness of the speaker verification system investigated.

While MFCCs are frequently employed in contemporary speaker recognition systems, their performance deteriorates drastically in telephone environments [17]. In [24], Moreno and Stern analysed various impairments in telephone networks that degrade the quality of speech signals. Impairments caused by additive noise and linear filtering were found to be amongst the most problematic. As such, emphasis should be placed on understanding these impairments and compensat-

ing for their adverse effects. The following section provides some insight into how additive noise and linear time-invariant filters affect a speech signal (and hence MFCCs). This is done so as to provide an explanation for why the feature-based compensation technique proposed in this study (and those discussed in Section 3.2) can be expected to improve speaker verification performance in telephone environments.

3.1 Additive noise and linear filtering effects

The speech signal received by the human ear (or a speaker recognition system) is not the same as the signal that was transmitted from the speaker's lips and nostrils. Instead, the speech signal has undergone several transformations that degrade its quality. A model that is often used to describe the effect of additive noise from the environment and linear filtering effects from communication channels (e.g., a telephone channel) on a recorded speech signal, $r(t)$, is depicted in Figure 3-1 [23]. Here, additive noise, $n(t)$, is added to the original speech signal, $s(t)$, and the composite signal is passed through a linear time-invariant filter with characteristic $h(t)$.

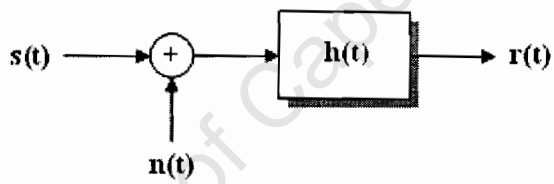

Figure 3-1: A model describing the effects of additive noise and a linear time-invariant filter on a recorded speech signal

Reynolds [26] refers to the model depicted in Figure 3-1 as the *degradation model* and adds that intersession variability can also be modelled as a linear time-invariant filter and thus can be grouped with $h(t)$. Therefore, this model provides a general framework for representing a broad class of distortions. The following sections provide an analysis of the impact of additive noise and linear filtering on MFCCs (which are the features used in this study).

3.1.1 Mathematical analysis

With reference to Figure 3-1, the recorded signal can mathematically be written as:

$$r(t) = [s(t) + n(t)] \otimes h(t), \tag{3.1}$$

with $\otimes$ denoting the convolution operator.

Furthermore, the magnitude-squared Fourier transform of $r(t)$ is given by:

$$|R(w)|^2 = \left|[S(w) + N(w)] \cdot H(w)\right|^2, \tag{3.2}$$

where $R(w)$, $S(w)$, $N(w)$ and $H(w)$ are the Fourier transforms of $r(t)$, $s(t)$, $n(t)$ and $h(t)$ respectively. Using this formulation, for a frame of speech, the log-energy computed for the k^{th} filter in a mel-scaled filterbank, $\log(E_k)$, may be specified by the composition of additive noise, $N(i)$, and linear filtering effects, $H(i)$, at each frequency index i as follows [80][22]:

$$\log(E_k) = \log\left(\sum_{i=a_k}^{b_k} \left|[S(i) + N(i)] \cdot H(i)\right|^2\right), \tag{3.3}$$

where, for simplicity, each filter is assumed to have a rectangular frequency response with discrete frequency indices a_k and b_k indicating the start and end frequency indices of each filter. The effect of the linear filter in Equation (3.3) may be isolated if it is assumed to be constant over the frequency range of the filter (i.e., if the filter is time-invariant: $H_k = H(s_k) \approx H(s_k+1) \approx \ldots \approx H(f_k)$). If it is further assumed that the real and imaginary components of the speech and noise are uncorrelated, the log-energy may be approximated as:

$$\log(E_k) \approx 2\log H_k + \log\left(\sum_{i=a_k}^{b_k} \left(|S(i)|^2 + |N(i)|^2\right)\right). \tag{3.4}$$

If we let the filterbank energies for the speech and noise be represented by S_k and N_k respectively, then for filter k we get:

$$\log(E_k) \approx 2\log H_k + \log\left(S_k + N_k\right). \tag{3.5}$$

From Equation (3.5) it can be observed that a linear time-invariant filter will introduce a global shift of the parameters representing the clean speech signal, while additive noise distorts these parameters non-linearly. Due to the discrete cosine transform, mel-frequency cepstral coefficients are in effect a weighted combination of the filterbank log-energies. As such, the effects of linear filtering and additive noise in the log-energy domain are also present in the cepstral domain. In order to illustrate these effects on the clean log-energy values output by filter k, and on their overall distribution, a Monte Carlo simulation was performed[23].

[22] Much of the information in this section was taken from this reference.

[23] A similar simulation was performed by de la Torre et al. in [105].

3.1.2 Simulation using artificial data

Equation (3.5) can be rewritten as:

$$\log(E_k) \approx 2h_k + \log\{\exp(s_k) + \exp(n_k)\}, \tag{3.6}$$

where $h_k = \log(H_k)$, $s_k = \log(S_k)$ and $n_k = \log(N_k)$. To represent the clean log-energies, s_k, a set of values were randomly generated according to a Gaussian probability distribution with zero mean and unity standard deviation. For the additive noise log-energies, n_k, values were randomly generated according to a Gaussian distribution with a mean of -1.25 and a standard deviation of 0.15. Figure 3-2 shows scatter plots of the clean log-energy values versus their contaminated counterparts obtained by the simulation (i.e., $\log(E_k)$ vs. s_k). In addition, lines representing the average value of the noise and the characteristic of the plot that would be obtained in the absence of additive noise and linear filtering effects (i.e., when $\log(E_k) = s_k$) are also depicted. In Figure 3-2(a), the log-energy of the component due to the linear filter, h_k, was set equal to 0 to obtain the contaminated log-energy values caused by the addition of noise only. To obtain the contaminated log-energy values caused by both additive noise and linear filtering effects, h_k was set equal to 1.5 in Figure 3-2(b).

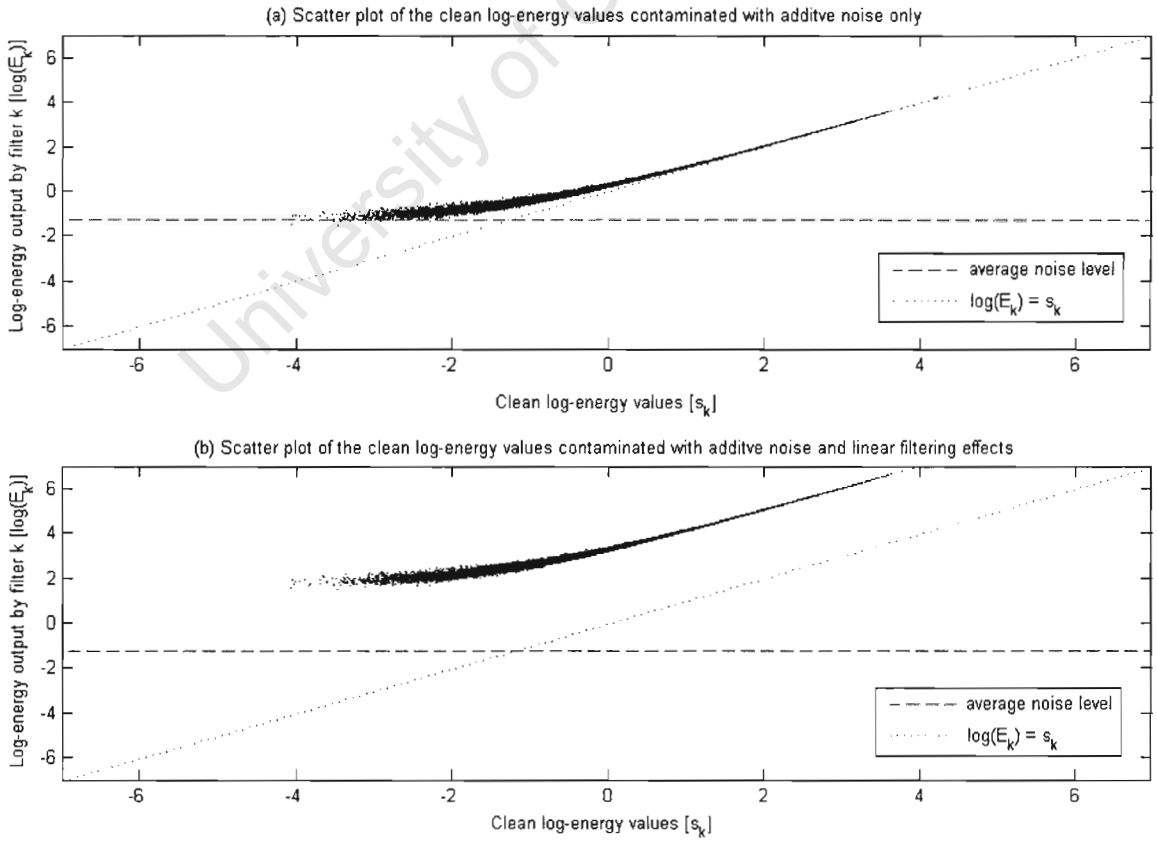

Figure 3-2: The effect of additive noise and a linear time-invariant filter on clean log-energy values

Figure 3-2(a) illustrates the situation in which the speech signal is only affected by additive noise (as $h_k = 0$). Here, it is clear that the clean log-energies have been non-linearly transformed as a result of the additive noise. This is because for values of the clean log-energies much greater than those of the noise log-energies, $\log(E_k)$ asymptotically tends to s_k, whereas for values of the clean log-energies in the same range as that of the noise log-energies, $\log(E_k)$ asymptotically tends to n_k. In Figure 3-2(b), the situation in which the speech signal is affected by both additive noise and linear filtering effects is depicted (as $h_k = 1.5$). When Figure 3-2(b) is compared to Figure 3-2(a) it is clear that the linear filter introduced an additive component in the log-energy values output by filter k, which resulted in a global linear shift of the points obtained by plotting $\log(E_k)$ versus s_k.

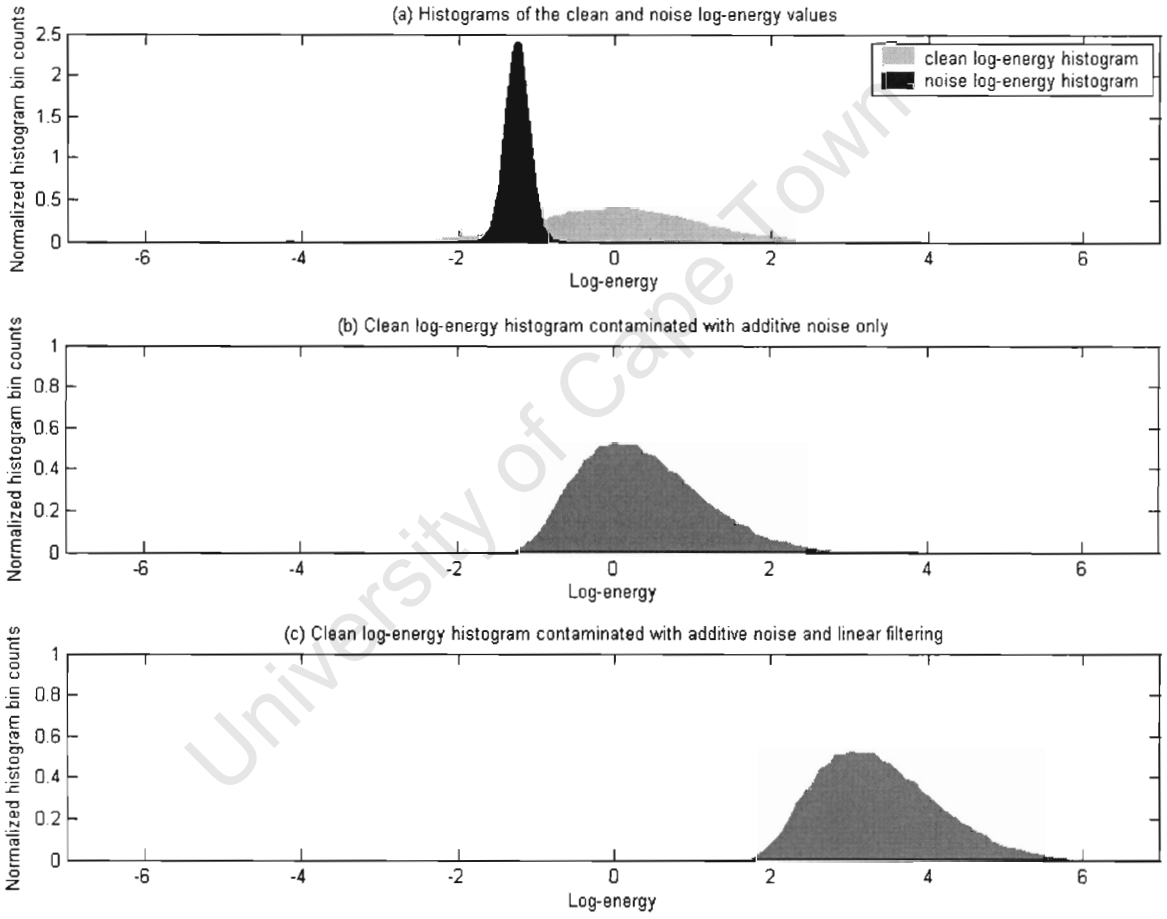

Figure 3-3: The effect of additive noise and a linear time-invariant filter on clean log-energy histograms

From the distortions depicted in Figure 3-2, it can be expected that additive noise and linear filtering effects also modify the distribution of the clean log-energy values. This is confirmed by Figure 3-3. Figure 3-3(a) illustrates the normalized histograms of the clean log-energy values and the additive noise log-energy values obtained by the simulation. Below it, Figure 3-3(b) shows the normalized histogram of the noise-contaminated clean log-energy values. This diagram shows that the non-linear transformation caused by the noise compresses the low energy part of the clean

log-energy histogram, which not only distorts its shape, but causes a reduction in the variance and a shift in the mean of the histogram. Figure 3-3(c) shows the histogram obtained when the clean log-energy values are distorted by components due to both additive noise and a linear time-invariant filter. As illustrated, linear filtering causes a shift in the mean of the histogram in addition to the distortions caused by the additive noise.

3.1.3 Simulation using real data

In order to confirm that these theoretically motivated degradations indeed exist in the real world, and that additive noise and linear filtering effects are indeed present in the cepstral domain, MFCCs were extracted from replicas of the test utterance "she had your dark suit in greasy wash water all year", taken from the TIMIT and NTIMIT databases[24]. These databases contain the same utterances but, recorded under different conditions. For the TIMIT database, all utterances are obtained in noise-free recording conditions, whereas for the NTIMIT database, all utterances are transmitted through a carbon-button telephone handset and recorded over local and long-distance telephone loops [47]. Figure 3-4 shows the histogram of the first component of the MFCC feature vectors (hereafter referred to as $MFCC_1$) extracted from the test utterance for both the TIMIT and NTIMIT databases[25].

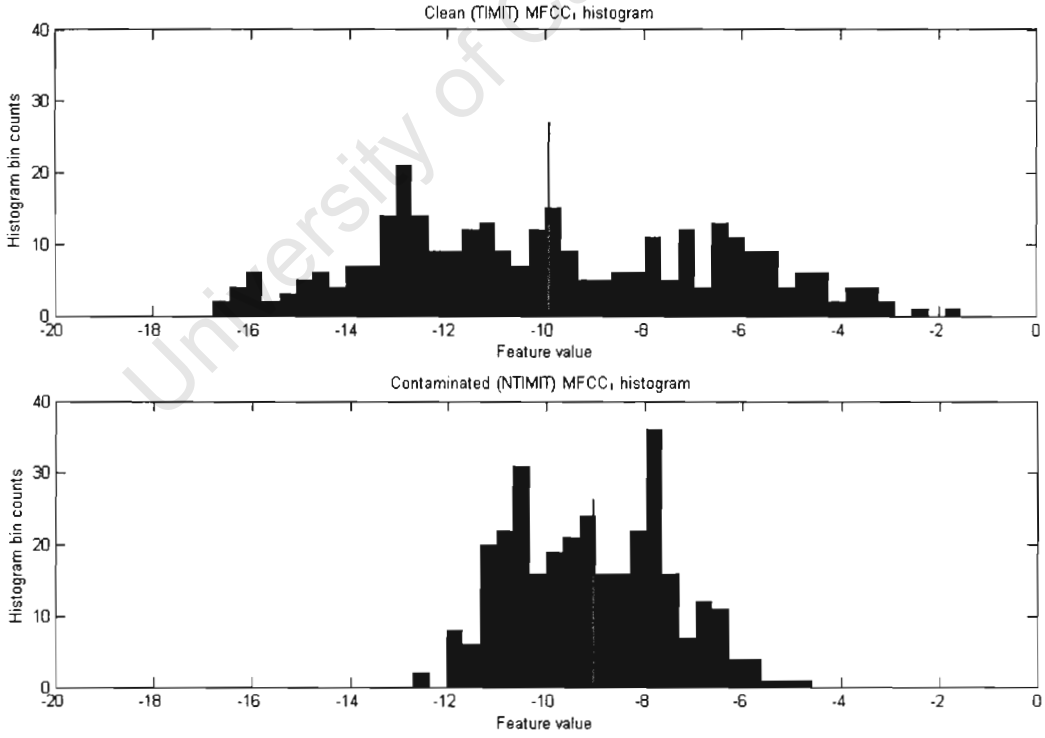

Figure 3-4: $MFCC_1$ histograms extracted from the same utterance in the TIMIT and NTIMIT databases

[24] The utterance was spoken by the same speaker.

[25] Similar plots were obtained for the other components of the extracted MFCC feature vectors.

As illustrated by Figure 3-4, the linear filtering property of telephone channels does indeed have an additive effect on the $MFCC_1$ values extracted from the NTIMIT version of the test utterance. This causes a shift in the mean of its histogram relative to the mean of the histogram of the $MFCC_1$ values extracted from the TIMIT version of the test utterance[26]. From the analysis presented earlier, the shift in the mean can also partially be attributed to additive noise effects. There is also a clear reduction in the variance of the histogram of the $MFCC_1$ values extracted from the NTIMIT version of the test utterance. This can be attributed to additive noises encountered in the telephone network. These noises include low frequency tone-like signals or white noise caused by thermal and other physical phenomena, as well as clicks and other transient artefacts caused by intermittent connections [81]. The non-linearity of carbon-button microphones (see reference [25]) can also account for some of the effects illustrated in Figure 3-4.

This section showed how additive noise and linear filtering effects distort a speech signal and consequently MFCCs as well. In the following section, a review of commonly used feature-based compensation techniques, as well as the distortions at which they are targeted, is provided.

3.2 Feature-based compensation techniques

Feature-based compensation techniques are aimed at making the features generated by the feature extraction component more robust to mismatched conditions. This is done by either normalizing feature distributions or by transforming the features in such a way that the degradations imposed by adverse environments are compensated for. In this section, several commonly used feature-based compensation techniques are reviewed.

3.2.1 Cepstral Mean Normalization

From the analysis presented in Section 3.1, it is clear that when a speech signal is passed through a linear filter, the resulting MFCCs contain an additive component, c_h, attributed to the linear filter. Thus, the cepstrum of the filtered signal, c_m, is equal to the cepstrum of the speech signal, c_s, plus the cepstrum of the filter:

$$c_m = c_s + c_h \tag{3.7}$$

[26] The means of the two distributions are indicated by the vertical lines splitting each histogram.

The average value of c_m for the duration of an utterance, $\{c_m(n)\}_{n=1}^{N}$, can be defined as follows:

$$\begin{aligned}\mu_{c_m} &= \frac{1}{N}\sum_{n=1}^{N} c_m(n) \\ &= \frac{1}{N}\sum_{n=1}^{N}\left(c_s(n) + c_h(n)\right) \\ &= \frac{1}{N}\sum_{n=1}^{N} c_s(n) + \frac{1}{N}\sum_{n=1}^{N} c_h(n).\end{aligned} \tag{3.8}$$

If it is assumed that in addition to the filter being linear, it is also time-invariant, then the second term in Equation (3.8), representing the cepstrum of the filter, becomes a constant:

$$\mu_{c_m} = \frac{1}{N}\sum_{n=1}^{N} c_s(n) + c_h. \tag{3.9}$$

If it is further assumed that the duration of the speech signal is long enough such that energy in the signal is uniformly distributed across the entire range of the spectrum, then the first term in Equation (3.9) tends toward zero [53]. Thus, the average of c_m represents an estimate of the cepstrum of the filter:

$$\mu_{c_m} \approx c_h. \tag{3.10}$$

The cepstrum of the original speech signal can then be recovered by subtracting this value from the cepstrum of the filtered signal [23]:

$$\hat{c}_m(n) = c_m(n) - \mu_{c_m}. \tag{3.11}$$

This technique is generally referred to as *cepstral mean normalization* (CMN) or *cepstral mean subtraction* and was first proposed by Atal in [82]. Over the years it has successfully been applied in numerous speaker recognition tasks [26, 28, 55] so as to compensate for the linear filtering effects induced by transmitting speech over telephone channels (and, to some extent, the effects of additive noise encountered in the telephone network). From Equation (3.11) it is clear that CMN also has the dual effect of causing the mean of the distribution of the compensated variable $\hat{c}_m(n)$ to be equal to zero. In so doing, it normalizes the first moment of the distribution.

The effectiveness of CMN is however limited, as it only provides a linear transformation of the feature space, which means that it is unable to adequately compensate for the non-linear effects of additive noise and telephone transmission. Furthermore, CMN requires the duration of the speech signal to be long enough to assume a flat long-term average spectrum, which is seldom the case in practical applications. For this reason, CMN will also tend to remove some speaker specific information in addition to linear filtering effects [23]. CMN has also been shown to degrade performance when applied to clean training and test speech recorded with the same microphone [28].

However, when applied to speech exposed to mismatched telephone environments, the technique has consistently improved the robustness and performance of speaker recognition systems employing MFCCs [28, 29]. In [26], CMN was also shown to be able to compensate for some of the effects of intersession variability when applied to clean speech data recorded at different times. According to the same reference, this is due to the fact that these effects are also well modelled by a linear filter.

3.2.2 Mean and Variance Normalization

Mean and variance normalization (MVN) can be viewed as an extension of CMN, as it not only provides a transformation that normalizes the mean of the distribution of each MFCC feature vector component but, one that normalizes its variance as well. It does this by transforming each MFCC according to the following equation [83]:

$$\hat{c}_m(n) = \frac{c_m(n) - \mu_{c_m}}{\sigma_{c_m}}, \tag{3.12}$$

where μ_{c_m} is the average value of a particular MFCC over the duration of an utterance (see Equation (3.8)) and σ_{c_m} is its standard deviation given by:

$$\sigma_{c_m} = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}\left(c_m(n) - \mu_{c_m}\right)^2}\ . \tag{3.13}$$

The transformation given by Equation (3.12) causes the mean of the resulting distribution to be equal to zero and its variance to be equal to one regardless of the conditions under which the speech data was collected. The analysis presented in Section 3.1 showed that additive noise non-linearly distorts the variance of the distribution of a particular MFCC feature vector component. Thus, MVN can be used to compensate for the effects of linear filtering as well as sources of additive noise [83]. However, the compensation provided by the technique is limited as it can only account for linear transformations of the mean and variance of MFCC distributions. The technique has however been shown to improve the robustness of MFCCs in various speech and speaker recognition tasks and was shown to be more robust than CMN [55, 80, 83].

3.2.3 RASTA Processing

Another feature-based approach that is often used to compensate for the effects of adverse environments is that of *RelAtive SpecTrAl* (RASTA) processing [84]. This technique takes advantage of the fact that the rate of change of the non-linguistic components in speech (due to additive noise or time-varying transmission channels for example) tends to lie outside the typical rate of

change of speech. It does this by suppressing these components. In essence, the time trajectories of the logarithmic spectral energies derived from short-time analysis (also known as the *modulation spectrum* [74, 85]) are filtered with a band-pass filter of the form [52]:

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_3 z^{-3} + a_4 z^{-4}}{(1 - b_1 z^{-1}) z^{-4}}. \tag{3.14}$$

This filtering operation is aimed at suppressing spectral components that vary more slowly or rapidly than the typical rate of speech. The process of obtaining cepstral derivatives (see Section 2.1.2.3) and CMN (see Section 3.2.1) can also be viewed as techniques that filter the trajectories of feature vector sequences [74, 85]. CMN can be interpreted as a high-pass filter as it effectively removes the DC component of the sequence of feature vectors to which it is applied. In [74, 85] and [86] RASTA processing was shown to improve the robustness of speaker recognition systems in mismatched environments. However, CMN was shown to outperform RASTA processing as it has a lower low cut-off frequency (e.g., 0.025, 0.075 and 0.25 Hz for window lengths of 30, 10 and 3 seconds respectively) when compared to the conventional RASTA filter which has a low cut-off frequency of approximately 1 Hz [74]. An analysis of the relative importance of the components of the modulation spectrum was presented in [74] and [85]. It was shown that spectral components between 0.1 Hz and 10 Hz contain the most useful speaker information. For this reason, van Vuuren and Hermansky [85] stated that RASTA processing could be more useful for speaker verification applications if components below 1 Hz are retained.

3.2.4 Feature Warping

In [80], Pelecanos and Sridharan proposed a novel feature-based compensation technique, termed *feature warping*. The technique employed a form of cumulative distribution mapping which nonlinearly transformed the statistics of feature distributions to that of a reference distribution over a specified time interval. This was done so as to construct a more robust representation of each MFCC feature vector component across different recording environments. The technique processes the distribution of each feature vector component separately. Feature warping is performed over a sliding window of size N and only the central feature, $c_m(n)$, in each window is transformed. Thus, the number of windows of features processed is equal to the number of features extracted from a speech signal. Feature warping is implemented as follows. The features are first sorted into ascending order and the rank, r, of the central feature (which has a value between 1 and N) is found. Its corresponding cumulative distribution function (CDF) value is then approximated as:

$$\Phi = \frac{(r - 1/2)}{N}. \tag{3.15}$$

In order to transform the statistics of $c_m(n)$ to that of a Gaussian distribution with zero mean and unity variance, its transformed value, $\hat{c}_m(n)$, should satisfy the equation:

$$\Phi = \int_{-\infty}^{\hat{c}_m(n)} P(x)dx, \tag{3.16}$$

where $P(x)$ is the probability density function (PDF) of the standard normal distribution, i.e.:

$$P(x) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right). \tag{3.17}$$

$\hat{c}_m(n)$ can therefore quickly be found using a simple lookup in a standard normal CDF table.

Pelecanos and Sridharan evaluated the technique on the database used in NIST 1999 speaker recognition evaluation and showed that it outperformed CMN, MVN and modulation spectrum filtering under various sources mismatch encountered in telephone environments. This is mainly because it "*compensates in part for the linear channel in that the short-term mean is removed, and attempts to conform the distributive shape and spread to limit additive noise effects*" [80]. In [87], feature warping was incorporated into *short-time guassianization* which uses a global linear transformation to decorrelate features before applying feature warping. The combined approach was shown to improve the performance of a speaker verification system evaluated on the NIST 2001 cellular phone corpus.

The technique proposed in this study (namely, Histogram Equalization) is in many ways similar to feature warping in that it also non-linearly transforms the statistics of feature distributions to that of a reference distribution. However, it differs in the manner in which the technique is implemented and applied to an utterance. Furthermore, in Chapter 6 a variation of HEQ is shown to outperform feature warping. HEQ is also less computationally expensive as, for the sliding window approach employed by feature warping, computational complexity and memory requirements are directly proportional to the length of the window used – larger window lengths will increase the time taken to perform feature warping as a larger number of features will need to be processed in order to obtain the normalized version of the central feature in each window. Moreover, in [80], the authors chose to use a sliding window with a 3 second duration without providing any comparative analysis of the effects of different window lengths. It is also unclear why the authors compared feature warping to other compensation techniques by using adapted GMM speaker models where the weights, means and variances were adapted when it is well known (and in fact mentioned in the paper in question) that adaptation of the means alone results in the best performance [60, 63].

From Figure 3-4 it is also clear that MFCC distributions are multimodal in nature. Hence, mapping feature distributions to a multimodal distribution, instead of a unimodal one, may be more appropriate. This conjecture, as well as other variations of the technique proposed in this study, is evaluated in Chapter 6 on the NIST 2000 telephony corpus. This database contains twice the amount of verification trials contained in the database used in the NIST 1999 speaker recognition evaluation. However, since the underlying motivation for using HEQ and feature warping is the same, namely, that matching the statistics of feature distributions obtained in different training and test conditions could improve the robustness of speaker verification systems, the feature warping approach does confirm the feasibility of the proposed technique. The following section provides a review of several score-based compensation techniques.

3.3 Score-based compensation techniques

The final step in speaker verification is that of decision-making, where a score (like a log-likelihood ratio for example) obtained for a targeted speaker model and test utterance pair, is compared to a decision threshold. If the score is above the threshold, the identity claim is accepted else, it is rejected. The setting of stable speaker-independent decision thresholds is however a non-trivial task. This is primarily due to the variability in the scores obtained for different speakers in mismatched training and test conditions. For this reason many score-based compensation techniques[27] have been proposed over the years (see for example references [7, 23, 55, 60, 63] and [88]).

Score normalization techniques are aimed at explicitly addressing the problem of score variability in speaker verification so as to allow for the setting of speaker-independent decision thresholds. This is done by normalizing speaker score distributions so as to minimise the effect of environment-dependent biases and scales. Note that for each speaker, performance is the same with and without score normalization but, when numerous scores are pooled and compared to a single threshold (as is the case in this study), score normalization results in enhanced performance by making the decision process more robust. In this section various score normalization techniques are reviewed.

[27] These techniques are also commonly referred to as score normalization techniques.

3.3.1 Zero Normalization

Zero normalization (Z-norm) [7, 23, 55, 88] is a score normalization technique that rescales the impostor log-likelihood distribution to a standard normal distribution by applying the following transformation:

$$\Lambda(X)_{Z-norm} = \frac{\Lambda(X) - \mu_I^S}{\sigma_I^S}, \tag{3.18}$$

where $\Lambda(X)$ is the log-likelihood ratio score for speaker S given a test segment X (see Equation (2.13)), μ_I^S and σ_I^S are the mean and standard deviation of the scores obtained for speaker S when evaluated against a set of impostor test segments and, $\Lambda(X)_{Z-norm}$ is the distribution normalized score. The procedure for obtaining the normalization parameters, μ_I^S and σ_I^S, is as follows. Given a development set[28], test the speaker model, S, against a set of impostor utterances from the development set and compute the mean and standard deviation of the scores obtained. Subsequently, normalize the log-likelihood ratio score for speaker S using Equation (3.18). In addition to producing scores that are normalized, Z-norm also allows for the estimation of normalization parameters to be performed off-line during training.

3.3.2 Handset Normalization

Handset normalization (H-norm) can be viewed as a variant of Z-norm that was proposed in [63] to alleviate handset-dependent biases and scales in log-likelihood ratio scores produced by handset mismatch between training and test data. This approach estimates handset-dependent normalization parameters by testing each speaker model against a set of handset-dependent impostor utterances [7, 60, 63]. During testing, the handset type of the test utterance X determines the set of normalization parameters to use for score normalization as follows:

$$\Lambda(X)_{H-norm} = \frac{\Lambda(X) - \mu_{H(X)}^S}{\sigma_{H(X)}^S}, \tag{3.19}$$

where $H(X)$ is the handset label of test segment X, $\Lambda(X)$ is the log-likelihood ratio score for speaker S given X, $\mu_{H(X)}^S$ and $\sigma_{H(X)}^S$ are the handset-dependent normalization parameters for speaker S determined by $H(X)$ and, $\Lambda(X)_{H-norm}$ is the distribution normalized score.

[28] The use of a development set is aimed at avoiding the introduction of a bias into the results [7] as this set contains data that is not part of either training or test sets.

3.3.3 Test Normalization

Test normalization (T-norm) was proposed in [88] and is similar to Z-norm in that it also uses mean and standard deviation estimates to normalize log-likelihood ratio scores. However, these estimates are obtained by scoring each test segment in the test set on a number of impostor models trained using data from a development set [7, 55, 88]. During testing, the log-likelihood of the test segment, X, given a targeted speaker in the training set is normalized as follows:

$$\Lambda(X)_{T-norm} = \frac{\Lambda(X) - \mu_I^X}{\sigma_I^X}, \tag{3.20}$$

where $\Lambda(X)$ is the log-likelihood ratio score for the targeted speaker given test segment X, μ_I^X and σ_I^X are the mean and standard deviation estimates obtained by evaluating test segment X on a set of impostor models and, $\Lambda(X)_{T-norm}$ is the distribution normalized score.

A disadvantage of T-norm is that the estimates of the normalization parameters have to be obtained on-line during testing. However, because the estimation of the normalization parameters is computed on the same utterance that is used to test a targeted speaker model, "*an acoustic mismatch, between the test utterance and the normalization utterances, possible in Z-norm, is avoided*" [88]. A handset-dependent variant of T-norm, namely HT-norm, has been shown to improve performance above that exhibited after the application of T-Norm, by compensating for the score variability due to handset mismatch as well [88]. Here, handset-dependent normalization parameters are estimated by testing each test utterance against a set of handset-dependent impostor models.

It should be noted that the score normalization techniques discussed in this section can be viewed as being *impostor-centric* since impostor score distributions are in effect being normalized [88]. This is because in order to have accurate estimates of the normalization parameters, a large amount of data is required. Unfortunately, contemporary speaker recognition databases [47] usually only contain sufficient data to allow for the estimation of impostor or pseudo-impostor distributions. A more thorough review of the normalization techniques discussed in this section as well as other normalization techniques can be found in [7]. According to the discussion presented in [7], HT-norm marginally outperforms all the score normalization techniques reviewed in this section. However, for the experimental work done in this study, T-norm was used to perform score normalization due to an imbalance in the handset-dependent training data available to train the impostor models.

3.4 Summary

This chapter showed through both an analytical argument and a simulation, that additive noise (from the environment for example) and linear time-invariant filtering effects (from communication channels for example) distort MFCCs and their distributions. This important observation is one of the main motivations for the feature-based compensation technique proposed in this study. As mentioned previously, the technique is aimed at normalizing feature distributions so as to minimise the mismatch between training and test conditions. This chapter also presented a review of several feature-based compensation techniques to provide some insight into the limitations of these techniques and their previous applications. Score-based compensation techniques were also reviewed as one of these is used in Chapter 5 when creating an experimental framework for evaluating Histogram Equalization. The following chapter introduces Histogram Equalization and provides a discussion of how it can be used to improve the robustness of speaker verification systems.

Chapter 4

Histogram Equalization (HEQ)

This chapter introduces the Histogram Equalization technique. This technique has been proposed in this study to improve the robustness of speaker verification systems operating in mismatched training and test conditions. Section 4.2 covers the mathematical formulation of HEQ and Section 4.3 provides a general overview of how the technique has been applied in its field of origin (i.e., in digital image processing). Section 4.4 provides a review of how HEQ has been applied in speech-related research and finally, Section 4.5 presents a simple algorithm for implementing the technique. The following section motivates why HEQ could be successful when applied to improve the robustness of a speaker verification system exposed to mismatched training and test conditions.

4.1 Motivation for using HEQ

Statistical speaker modelling strategies, like GMMs and HMMs for example (see Section 2.3), are aimed at modelling the underlying distribution of feature vectors extracted from a particular speaker's speech during training. During testing, speakers are classified according to the statistical similarity between the features extracted from their speech and the speaker models generated during training. This approach is based on the implicit assumption that for the same speaker, the features extracted from his (or her) speech will have similar statistical properties. A mismatch between the statistical properties of the training and test speech however, to some extent violates this assumption and leads to a deterioration in classification performance.

From the discussion presented in Section 1.2, it is clear that in many practical applications of speaker verification, a system will have to operate under non-ideal conditions. That is, the input speech could be corrupted by ambient noise or by distortions caused from transmitting it over a telephone channel. In Section 3.1 it was shown that distortions caused by additive noise and linear filtering effects corrupt spectral-based features and, as a result, modifies their distributions. Furthermore, different environmental conditions and communication channels affect speech signals differently. Thus, a statistical speaker model trained with speech collected in one environment will generally perform poorly when recognizing the same speaker using speech collected under different recording conditions, since the feature distributions will be different.

In this thesis, a feature-based compensation technique, known as *Histogram Equalization* (HEQ), is proposed to minimise the mismatch between feature distributions collected under different recording conditions. It does this by non-linearly transforming the characteristics (i.e., the scale, shape and location) of one probability distribution to that of another such that their statistical properties (i.e., the mean, variance and skew) match. The technique was originally used in digital image processing to alleviate brightness and contrast alterations in digital images. In this work the use of this technique is motivated by the fact that feature-based compensation techniques that normalize the first and second moments of feature distributions, like CMN and MVN for example (see Section 3.2), have been shown to be effective in improving speaker verification performance in adverse environments. However, CMN and MVN are linear techniques which limits their ability to compensate for non-linear distortions of the feature space (such as those caused by additive noise for example).

The non-linear compensation provided by HEQ however, can be used to not only normalize the first two moments of feature distributions, but all the other moments as well. As such, HEQ can be used in speaker verification to map the characteristics of a particular speaker's feature distributions, obtained during training and testing, to that of a reference (or target) distribution regardless of the conditions under which the speech was collected. In so doing, the statistical mismatch between the training and test feature distributions will be reduced, which in turn, can be expected to improve the accuracy a of statistical speaker recognition system (like the experimental framework developed to evaluate HEQ, see Chapter 5). The following section mathematically derives the non-linear transformation provided by HEQ.

4.2 Mathematical formulation

As mentioned in the previous section, HEQ provides a transformation that allows one to convert one probability distribution to another. It does this by matching the CDFs of the reference distribution and that of the variable to be transformed. This is accomplished as follows [89-92]: Let x be a random variable with a probability distribution $p_x(x)$, and let $y = T(x)$ be a single-valued and monotonically increasing transformation that converts the probability distribution $p_x(x)$ into a reference probability distribution $p_{ref}(y)$. $T(x)$ thereby makes the probability of finding x in the differential range dx equal to the probability of finding y in the differential range dy, i.e.:

$$p_{ref}(y)dy = p_x(x)dx. \tag{4.1}$$

Thus, the transformation $y = T(x)$ modifies the original probability distribution $p_x(x)$ according to the expression:

$$p_{ref}(y) = p_x(x)\frac{dx}{dy} = p(G(y))\frac{dG(y)}{dy}, \tag{4.2}$$

where $G(y)$ is the inverse of $T(x)$. Using Equation (4.2), the relationship between the cumulative distribution functions associated with $p_x(x)$ and $p_{ref}(y)$ is as follows:

$$\begin{aligned} C_x(x) &= \int_{-\infty}^{x} p_x(x')dx' \\ &= \int_{-\infty}^{T(x)} p_x(G(y)')\frac{dG(y)}{dy}dy' \\ &= \int_{-\infty}^{y} p_{ref}(y')dy' \\ &= C_{ref}(y) \\ &= C_{ref}(T(x)). \end{aligned} \tag{4.3}$$

Thus, the transformation $T(x)$, that converts $p_x(x)$ into $p_{ref}(y)$, is given by:

$$T(x) = C_{ref}^{-1}(C_x(x)), \tag{4.4}$$

where C_{ref}^{-1} is the inverse of the CDF of the reference probability distribution.

For practical implementations only a finite number of observations are usually available. As a result, cumulative histograms instead of cumulative probabilities are used. This is the reason that the transformation is called Histogram Equalization and not probability distribution equalization. The transformation given by Equation (4.4) cannot however easily be applied to the multi-dimensional feature vectors generated by the feature extraction component of a speaker recognition system. For this reason, it is assumed that all the dimensions of the feature vectors are independent. Under this simplifying assumption, the transformation can be applied to each feature vector component independently. A graphical illustration of the cumulative distribution matching performed by HEQ is depicted in Figure 4-1. It shows how the cumulative histogram of the original variable, x, and the reference cumulative histogram can be used to perform the transformation. Here, each value of x is replaced by the value of y that corresponds to the same point in the cumulative histogram of the original variable and the reference cumulative histogram.

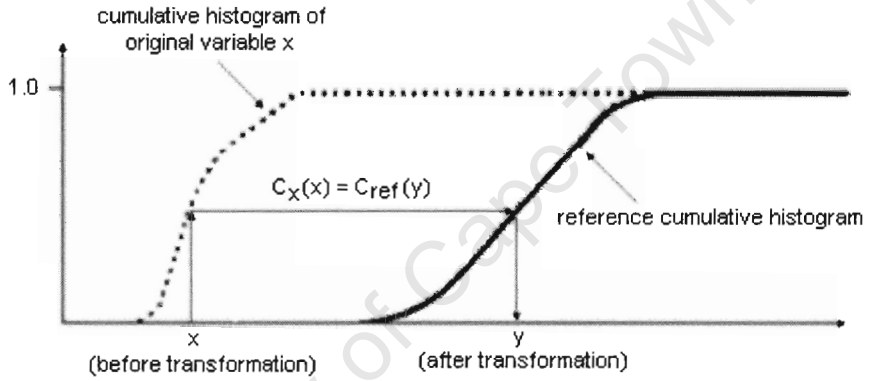

Figure 4-1: The cumulative distribution matching performed by HEQ

In the following section an overview of how HEQ is used in its field of origin is presented. The expected limitations of the technique, when applied to speech processing, are also highlighted.

4.3 Image processing background

Over the years Histogram Equalization (also commonly referred to as Histogram Matching, Histogram Specification and Histogram Normalization) has extensively been used in digital image processing to improve the brightness and contrast of digital images. It does this by optimizing the dynamic range of the grey-level scale [93-95] as depicted in Figure 4-2.

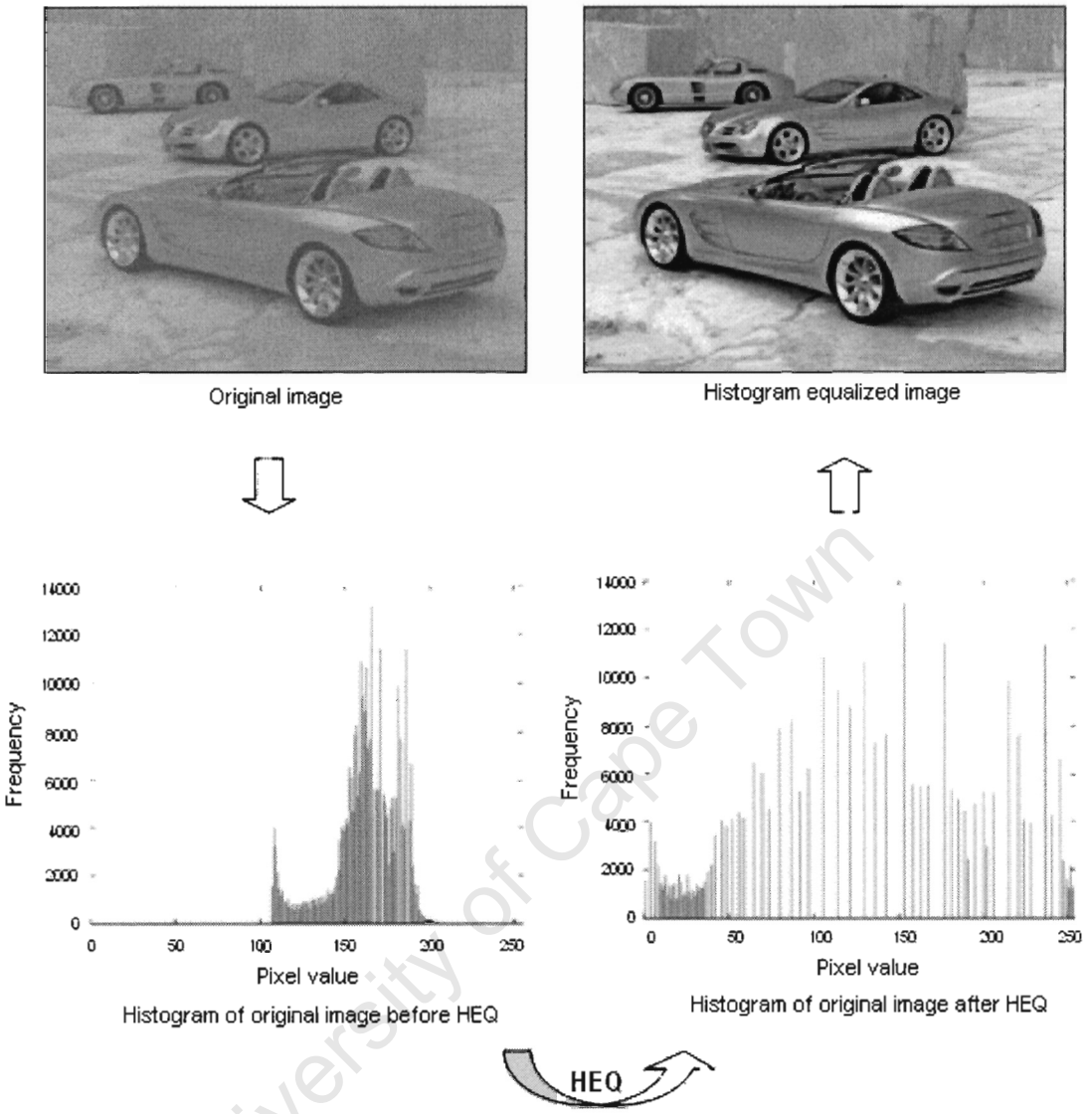

Figure 4-2: The application of HEQ to enhance a digital image [95]

As can be observed from Figure 4-2, the original image is very dull and lacks contrast. As a result, its histogram[29] of pixel values is compressed such that it occupies only a small region of the grey-level scale – the majority of the pixel values fall into the range 105 to 205. Here, Matthews [95] used HEQ to convert the original histogram of pixel values into a more uniformly distributed histogram which enhanced the brightness and contrast of the image.

[29] A histogram plots the frequency at which each pixel value occurs (from 0 for black to 255 for white).

In general, HEQ only provides an accurate compensation for the effects of non-linear transformations of the feature space provided that [89]:

1. "*There are sufficient observations of the signal being compensated*". This condition is adequately met in image processing as an image typically contains a large number of pixels (usually several thousand to several million) which all contribute to accurate estimations of the histograms used in HEQ.

2. "*The transformation is monotonic*" and hence, does not lead to a loss of information. In image processing, incorrect lighting and non-linearities in the receptors are mainly responsible for images that are too bright or too dark or those that lack contrast. These degradations usually correspond to non-linear monotonic transformations of the grey-level scale.

While these two conditions make the application of HEQ in digital image processing very effective, it limits the effectiveness of HEQ in speaker recognition as there is usually much less data available to estimate the histograms accurately – especially when short training and test utterances are available. Also, according to [89], the non-linear transformation caused by additive noise primarily has two effects on a speech signal:

1. It distorts the speech signal such that a mismatch between training and test conditions occurs and,
2. Due to its random nature, can cause a non-monotonic transformation of the feature space which will lead to an irreversible loss of information.

Thus, while HEQ equalization will be able to reduce the mismatch between training and test conditions, it will not (like other compensation techniques) be able to recover lost information. Furthermore, the assumption that all feature vector components are independent means that Histogram Equalization cannot be used to compensate for correlated distortions of the feature space, such as rotations for example. HEQ can however be used to deal with a scaling, shift or any other non-linear transformation of each feature vector component. In the following section an overview of the application of Histogram Equalization in speech-related research will be presented. This section will show that even with the limitations of HEQ highlighted in this section, HEQ has still been shown to improve the performance of a number of speech recognition systems in adverse conditions.

4.4 Speech processing background

Histogram Equalization was first applied to speech-related research in 1998 when Balchandran and Mammone applied the basic technique of matching cumulative distributions to a speaker identification task [96]. The technique was found to be "*robust, computationally efficient and universally applicable*". It was applied directly to raw speech samples and was shown to successfully restore artificially distorted speech, without itself introducing any noticeable distortion. However, due to the fact that the research was conducted on artificially corrupted speech, it has received much criticism [90, 97]. Also, an additional smoothing factor had to be introduced to avoid over-compensation.

Two years later, HEQ was applied for the first time to speech recognition (using MFCCs) in the form of an unsupervised histogram-based mapping technique. Dharanipragada and Padmanabhan used the technique to rapidly adapt a speech recognition system to new acoustic conditions [92]. The technique was based on the idea of mapping the cumulative distribution of the test data to the cumulative distribution of the training data. Under the simplifying assumption of independence between the dimensions of the feature vectors, this resulted in a simple, text-independent, histogram mapping procedure that was non-parametric, non-linear and computationally efficient. The performance of the technique was shown to be comparable to that of another adaptation technique termed maximum likelihood linear regression (MLLR) and, resulted in a relative reduction in the word error rate of the baseline system of over 30%. Additional improvements in performance were also reported when the technique was combined with MLLR. These experiments showed that HEQ could be used to compensate for the mismatch between speech collected with a telephone handset and a speakerphone.

In a series of papers from 2001 to 2003, Molau et al. studied the application of HEQ to speech recognition under adverse acoustic conditions [90, 97-99]. The technique was termed "*Histogram Normalization*". In [98], experiments were conducted to determine: (1) at which stage during feature extraction HEQ should be applied; (2) the effect of normalizing both the training and test data; and (3) the effect of using a smoothed reference distribution. The application of HEQ at different feature extraction stages consistently improved system performance. However, from the results presented, the largest reduction in the word error rate, of about 9% relative to the baseline system, was observed when HEQ was applied at the filterbank stage of feature extraction (i.e., after mel-scale filtering) with a smoothed reference distribution normalizing both the training and test data. The reference distribution was estimated from the overall distribution of the training data and was smoothed by approximating it with a mixture of two Gaussians according to a minimum mean-squared error criterion.

In [99], Molau et al. enhanced the HEQ-based system described in [98] by the application of two other techniques; namely, silence fraction treatment and feature space rotation. HEQ was said to rely on two basic assumptions: (1) "*the global statistics of the speech signal are the same independent of what was said*" (i.e., the frequency of the phonemes[30] occurring in both the training and test speech is similar); and (2) "*the feature space dimensions are oriented such that the variations are independent in each dimension*". The first assumption is often violated due to the fact that different speakers have different periods of silence in their speech. These differences lead to variations in the histograms estimated for HEQ. This can degrade speech recognition performance as more word insertions will occur for speakers with a higher than average silence fraction, since HEQ will erroneously convert a number of feature vectors to speech. On the other hand, for speakers with a lower than average silence fraction, some feature vectors containing speech will be converted to silence and cause more word deletions. The solution devised in [99] was to estimate two independent histograms, one for speech and one for silence. In the normalization step, the silence fraction of each speaker is estimated. This value is then used to form an adapted reference histogram for each speaker by linear interpolation between the speech and silence histograms estimated from all the training data.

The second basic assumption, that variations in the feature space dimensions are uncorrelated, is violated when the feature space is rotated by a small amount. To this end, explicit feature space rotations were explored. Rotation matrices were designed such that the principle axis with the largest data scatter becomes identical for all speakers. In [99], experiments were conducted on the VerbMobil II, EuTrans II and CarNavigation databases to determine the effect of silence fraction treatment and feature space rotation. The VerbMobil II database contains conversational speech recorded with a headset and a room microphone. The EuTrans II database contains conversational speech recorded over a telephone channel that varied significantly between recording sessions; and the CarNavigation database contains isolated words recorded in a quiet office environment, as well as in city and highway traffic. The experimental results showed that HEQ with silence fraction treatment led to improved performance on all the previously mentioned databases. However, a sequential application of HEQ with silence fraction treatment and feature space rotation only improved the results in a few cases and depended on the order in which the two techniques were applied. Also, since silence fraction treatment and feature space rotation are speaker-dependent, they can be viewed as speaker normalization techniques. While normalizing the speaker population is desirable in speech recognition applications, it will degrade the performance of speaker recognition applications as more inter-speaker confusions will occur. In his PhD dissertation [90],

[30] A phoneme is the smallest unit of sound in a spoken language that distinguishes one word from another.

Molau consolidated many of the findings presented in [98] and [99] and also discussed HEQ, silence fraction treatment and feature space rotation in more detail. The journal paper "*Matching Training and Test Data Distributions for Robust Speech Recognition*" [97] further recapitulates the work done by Molau and his colleagues in [90, 98] and [99].

In [100] and [101], Hilger et al. used a parametric form of Histogram Equalization, termed "*quantile equalization*", to improve the robustness of a speech recognition system. For this technique, a small number of quantiles (or bins) of the cumulative distributions used in the HEQ formulation were estimated. Piece-wise linear and power transformation functions were then fitted to these quantiles according to a minimum mean-squared error criterion so as to approximate the actual transformation functions. In the experiments, quantile equalization was applied after mel-scale filtering, and only the test data was normalized. The cumulative distribution of the training data, averaged over all filter channels, was used as the reference distribution. Speech recognition experiments on a number of databases recorded in car environments (e.g., city and highway traffic) showed power function transformations to be more robust than piece-wise linear transformations. Furthermore, it was shown that quantile equalization using only four quantiles is sufficient to obtain significant reductions in the word error rates of these databases – especially under high mismatch conditions. It was also shown that single word utterances were sufficient to reliably estimate the transformation functions.

In 2002, Segura et al. showed that HEQ could be used to improve the performance of a technique known as the Vector Taylor Series (VTS) [102]. VTS is a signal-based compensation technique that is aimed at producing the clean version of a noise contaminated speech signal given statistical models describing the clean speech and the noise. However, the compensated signal is only an approximation of its actual clean version, and as such, still retains a residual noise. In [102], HEQ was applied during both training and testing to remove the non-linear distortion of MFCC distributions caused by this residual noise. The reference distribution for the HEQ technique was a Gaussian distribution with zero mean and unity variance (i.e., the standard normal distribution). The application of HEQ after VTS was shown to result in a relative improvement of about 6% in the word accuracy of a speech recognition system trained with clean speech and tested with speech contaminated with several noise types at different signal-to-noise ratios.

Later in 2002, Segura et al. showed the versatility of HEQ when it was used to compensate for the residual noise caused by another signal-based compensation technique, namely Spectral Subtraction (SS) [103]. SS was used to reduce the effects of additive noise in the spectral domain while HEQ was applied in the cepstral domain to compensate for the effects of the residual noise caused by SS on MFCC distributions. In addition, HEQ was used to reduce the effects of channel mismatch as SS is unable to deal with this type of distortion. Once again, a Gaussian probability dis-

tribution with zero mean and unity variance was used as the reference for HEQ. Speech recognition experiments were conducted on the Aurora II and Aurora III databases. The results of these experiments showed that the application of SS led to relative improvements in the word accuracy of a baseline system employing MFCCs of 23.57% on the Aurora II database and 30.54% on the Aurora III database. The subsequent application of HEQ further increased the relative improvements to over 35% and 45% on the Aurora II and Aurora III databases respectively.

In 2003, Segura et al. examined the use of a segmental version of Histogram Equalization [104]. While the original non-segmental version of HEQ is applied on an utterance-by-utterance basis, segmental HEQ is applied on overlapping buffers of features extracted from a single utterance. Only the central feature in each buffer is normalized, and each successive buffer is formed by removing the first element of its predecessor and appending the next time-ordered feature to the end of this buffer. In order to improve the computational efficiency of the technique, HEQ was applied by exploiting the relationship between the order statistics of a dataset (i.e., a buffer of features) and its cumulative histogram. In so doing, an asymptotically unbiased point estimate of the CDF of the data set was obtained. For buffers of features extracted from 600 milliseconds of speech, segmental HEQ was found to perform slightly worse than its non-segmental counterpart. However, for on-line applications, like financial transactions over the telephone, segmental HEQ can be applied while an individual is speaking. This not only avoids long or variable delays attributed to the time taken to perform HEQ after an individual is done speaking but, allows the algorithm to adapt to changing environmental conditions. The segmental form of HEQ is essentially the same as the feature warping technique discussed in Section 3.2.4.

A year later in [91], Segura et al. explored the segmental version of HEQ in more detail. Two computationally efficient algorithms were presented for its implementation so as to avoid computing an entire cumulative histogram for each buffer of features. As in [102-104], Segura et al. selected the reference distribution to be Gaussian with zero mean and unity variance, and HEQ was applied in the cepstral domain. The size of the buffer of features was varied from 100 to 1400 milliseconds with a buffer size of 600 milliseconds producing the best overall results. When performing speech recognition experiments on a system trained with clean speech and tested with noisy speech, the segmental version of HEQ was found to marginally outperform non-segmental HEQ. However, non-segmental HEQ was not directly implemented. Instead, segmental HEQ with a buffer size of 2500 milliseconds was used to approximate the performance of non-segmental HEQ. Furthermore, when training and testing the system with speech contaminated with different kinds and levels of noise, non-segmental HEQ was found to marginally outperform the segmental version of HEQ. Segmental HEQ was however shown to outperform segmental implementations of CMN and MVN.

In 2002, de la Torre et al. showed both theoretically and by means of a simulation that additive noise non-linearly distorts MFCCs [105]. This distortion causes a mismatch between training and test conditions which significantly degrades the performance of speech recognisers. HEQ was proposed to compensate for the effects of additive noise. It was shown that in addition to non-linearly distorting the feature space, additive noise also causes a shift in the mean and a reduction in the variance of the distribution of the parameters representing the speech signal. This is why the linear transformation provided by cepstral mean normalization (which normalizes the mean of a distribution) and mean and variance normalization (which normalizes the mean and variance of a distribution) can be used to only moderately reduce the effects of additive noise. HEQ, CMN and MVN were applied to a connected-digit recognition task where the speech was contaminated with different noise types at several signal-to-noise ratios so as to simulate various adverse environmental conditions. HEQ was shown to outperform CMN and MVN due to its ability to compensate for linear as well as non-linear distortions of the feature space. In the experiments performed in [105] HEQ was applied on a sentence-by-sentence basis where the required histograms were estimated using the feature vectors extracted from each sentence. Mel-frequency cepstral coefficients were used to parameterise the speech signal and a Gaussian distribution with zero mean and unity variance was once again used as the reference distribution. HEQ does not make any assumptions of how different noises distort the speech parameterisation (i.e., the extracted speech features) and does not depend on the parameterisation used. For these reasons, de la Torre et al. concluded that it could be used to reduce the effects of a wide range of noise processes and could be combined with other noise compensation methods to obtain additional improvements.

In [89], de la Torre et al. stated that the fact that HEQ is absent of any assumptions concerning the contamination process could be considered as one of the limitations of the technique. This is because compensation techniques are generally based on some estimation of the type of noise affecting the speech signal together with a statistical or analytical model describing the noise effects. For this reason, these techniques could be expected to provide a more accurate compensation of specific noise effects. However, it does reinforce the notion that HEQ could be expected to compensate for a wide range of noise processes affecting a wide variety of speech parameterisations, and lead to additional improvements when combined with other noise compensation methods. This notion was experimentally verified when HEQ was combined with the Vector Taylor Series noise compensation technique. The combination resulted in a performance superior to that of each method applied in isolation. This consolidates the results obtained by Segura et al. in [102] where HEQ was also combined with VTS. An experimental set-up similar to that used in [105] was employed.

In [106], Obuchi and Stern observed that although the use of time-derivative parameters such as delta and delta-delta features (see Section 2.1.2.3) improved speech recognition performance, few attempts have been made to normalize these features. To this end various strategies for applying HEQ to MFCCs and their first and second derivatives were developed and evaluated. All the strategies improved speech recognition performance on speech recorded with the built-in microphone of a personal digital assistant. However, due to the extra computations required, the execution time of these strategies was slightly longer than HEQ performed on MFCCs alone.

In [107], HEQ was combined with the time domain noise reduction technique proposed by Noé et al. in [108]. The application of the two techniques in tandem once again showed that since HEQ makes no assumptions about the process distorting the speech signal, it can be combined with other noise compensation techniques to obtain additional improvements.

From the literature reviewed in this section, the following summary concerning the speech-related use of Histogram Equalization can be made:

- HEQ has been shown to improve the robustness of numerous speech recognition systems evaluated on several tasks involving speech corrupted by adverse recording conditions (like in the presence of ambient noise for example).
- It has also been shown to outperform linear feature-based compensation techniques, like CMN and MVN, due to its ability to compensate for both linear and non-linear distortions of the feature space.
- HEQ has had limited application in the area of speaker recognition.
- Gains in speech recognition performance, when HEQ was applied at various signal analysis stages during feature extraction, were reported.
- There have been more applications of HEQ to speech corrupted by additive noise than speech contaminated by telephone transmission.
- Substantial improvements in speech recognition performance were reported when HEQ was applied as either a stand-alone technique or in combination with other noise compensation techniques.
- Various forms of HEQ exist (e.g., quantile equalization and segmental HEQ), with each having its own pros and cons as far as its computational requirements and performance are concerned.
- HEQ is very versatile as it does not make any assumptions of how different sources of noise distort the speech parameterisation, and does not depend on the parameterisation used.

In the following section, a simple algorithm for implementing the Histogram Equalization technique is presented.

4.5 Practical implementation

This section provides a simple algorithm for directly implementing the HEQ technique described in Section 4.2. Similar algorithms can be found in [90, 92]. As mentioned previously, HEQ is applied separately to the distribution of each feature vector component extracted from the speech utterance under consideration. As such, the algorithm below is applied to each feature vector component independently and subscripts are dropped for ease of notation.

The goal of HEQ is to modify the feature distributions obtained during training and testing such that their characteristics are similar to that of a reference distribution. Thus, the first step in performing HEQ involves selecting a reference distribution. Once a suitable reference distribution, $p_{ref}(y)$, has been selected and its cumulative histogram, $C_{ref}(y)$, has been computed, HEQ can be applied to the training and test feature distributions of each speaker as follows:

1. Determine the maximum and minimum values, x_{max} and x_{min}, across the entire set of observations (i.e., across all the observations of a particular feature vector component).
2. Divide the range $[x_{max}, x_{min}]$ into M equally-spaced non-overlapping bins (or intervals), B_i, where $x_{min} = b_1 < b_2 < \ldots\ldots < b_{M+1} = x_{max}$ and $B_i = [b_i, b_{i+1})$.
3. Using these bins, construct a histogram of the observations in the set. This is done by scanning the set and counting the number of observations that fall into each bin.
4. Compute the normalized version of the histogram obtained after step (3) by using the following equation:

$$p_x(x \in B_i) = \frac{n_i}{N_x}, \tag{4.5}$$

where n_i is the number of observations in bin B_i and N_x is the total number of observations in the set. Equation (4.5) in effect approximates the probability of x being in bin B_i.

5. Compute the cumulative histogram of the set using the normalized histogram constructed in step (4) such that:

$$C_x(x : x \in B_i) = \sum_{j=1}^{i} \frac{n_j}{N_x}. \tag{4.6}$$

Equation (4.6) is a piecewise constant function approximation of the true cumulative distribution function.

6. Replace each value of x by the value of y that corresponds to the same point in the reference and computed cumulative histograms such that $C_x(x) = C_{ref}(y)$. This is in direct correspondence to Equation (4.4).

To efficiently implement step (6), one could construct two lookup tables $\{x , C_x\}$ and $\{y , C_{ref}\}$ from $C_x(x)$ and $C_{ref}(y)$ respectively, such that they take on values in the range [0,1] in equal increments. This allows one to combine the two tables such that a new table $\{x , y\}$, which is a piecewise constant approximation of the true transformation function, is formed [92]. Then, for every value of x, the closest value of y can be found by using a binary search. This value can then be used as the normalized value of x.

4.6 Summary

This chapter introduced the Histogram Equalization technique and provided a discussion of how it has been used in digital image processing. Previous applications of the technique in speech-related research were also reviewed. This review showed that while HEQ has been shown to improve the robustness of numerous speech recognition systems evaluated on speech corrupted by adverse recording conditions, it has had limited application in the area of speaker recognition and on speech corrupted by telephone transmission. The following chapter describes the development of an experimental framework (i.e., a baseline speaker verification system) for evaluating HEQ. In Chapter 6, HEQ is applied to improve the robustness of this system when evaluated on a database contaminated by telephone transmission.

Chapter 5

Experimental Framework for Evaluating HEQ

The purpose of this chapter is to describe the design, implementation and evaluation of a baseline text-independent speaker verification system. This system will provide an experimental framework for evaluating the HEQ technique proposed in this study. Furthermore, the performance of this system will be used as a benchmark against which all subsequent improvements will be compared. The main design criteria for the baseline system are as follows:

1. It should be based on techniques detailed in contemporary literature, namely those discussed in Chapters 2 and 3. This is done so as to create a baseline system incorporating techniques that are generally regarded as standard practice when constructing speaker verification systems. Furthermore, there is not much point in developing a system using outdated techniques that have been shown to be inferior to more contemporary techniques.
2. Its performance should be on par with other speaker verification systems evaluated under similar conditions. This would not only verify that the implementation is correct, but would provide a good foundation for further improvement. Furthermore, if the performance of the baseline system is way below that of other systems, then the credibility of any subsequent improvements may be questioned.
3. The software created should be well-commented and modularised so that it can easily be reused and modified. Also, design decisions and parameter selections should be made to make the system less computationally intensive as long enrolment and verification times could detract potential users. For example, it is well known that when a large amount of training data is available, GMMs with larger model orders generally lead to improved performance, but at the expense of extra computational time and memory requirements.

As such, a design decision concerning the number of mixtures in a GMM needs to be made so as to allow for a suitable trade-off between performance and computational complexity.

With these criteria in mind, the layout of this chapter is as follows. Section 5.1 deals with the characteristics of the speech database used to evaluate the baseline system as well as the procedure for using this database. In Section 5.2, the design and implementation of the baseline system is discussed. Finally in Section 5.3, the baseline system is evaluated. This section not only verifies the system implementation but, experimentally confirms many of the observations reported in contemporary literature.

5.1 Experimental database and protocol

Since 1996, the National Institute of Standards and Technology (NIST) have annually coordinated text-independent speaker recognition evaluations where participants from around the world evaluate their speaker recognition architectures on a number of different speaker recognition tasks [109]. The data used in the evaluations is extracted from the Switchboard speech corpora [47, 110] which contain thousands of telephone conversations involving hundreds of speakers. The conversations generally involve two adults (who do not know each other) and are typically five to ten minutes in duration. Since the data is collected in what is referred to as telephone environments, "*the challenges presented by this data include limited bandwidth, channel noise from various sources, the use of different microphones, recordings from different locations, and recordings collected over a period of time*" [111] . All these factors contribute to mismatched training and test conditions.

5.1.1 The one-speaker detection task

All the annual NIST evaluations have included the basic *one-speaker detection* task which consists of a series of verification trials (or access attempts). For each trial, the task is to determine whether a specified speaker is speaking in a given single-channel segment of μ-law encoded telephone speech [110]. Each trial presents a speaker verification system with a targeted speaker model (created from speech obtained from the targeted speaker) and a test segment spoken by a single unknown speaker. The system must then decide whether or not the unknown speaker is the speaker that was targeted. Two types of trials exist [109]: (1) *target trials* where the unknown speaker is the targeted speaker, and (2) *non-target trials* where the unknown speaker is someone else.

Evaluation kits from past NIST speaker recognition evaluations are publicly available from the Linguistic Data Consortium (see footnote 4 on pg 16), and includes training data to generate speaker models, test segment data to test speaker models and index files specifying the individual verification trials (i.e., targeted speaker and test segment pairs). A typical evaluation kit includes hundreds of speakers and thousands of test segments.

5.1.2 The NIST 2000 evaluation kit

For the experimental work done in this study, the database used in the NIST 2000 speaker recognition evaluation [112] was used. This was primarily due to it being readily available and due it containing mismatched training and test data. The complete one-speaker detection task includes data from 1003 (546 female and 457 male) speakers and requires the evaluation of 6096 target trials and 60476 non-target trials. The speech in this database was extracted from the Switchboard-II corpus, phases 1 and 2. The Switchboard II corpus, phase 1, consists of 3702 recorded telephone conversations from 661 speakers mainly from the North-eastern United States [110, 111]. The conversations are typically 5 minutes in duration and involve two adults discussing a suggested topic - the conversations were however not restricted to the proposed topic. The speakers were required to initiate each of their calls from a different telephone and, each speaker was only allowed to receive and initiate one call per day. On average, each speaker participated in 11 calls. The Switchboard II corpus, phase 2, was collected in a similar manner. This corpus however, contains 4575 conversations from 684 speakers mainly from the Mid-western United States, each participating in an average of 13 calls [110, 111].

The speech data collected for each speaker has been digitised at a sample rate of 8 kHz and stored as 8-bit μ-law encoded speech signals in separate NIST SPHERE audio files. The header of each file contains fields such as the sample count, channel count and sample rate as well as the type of microphone employed in the telephone handset used to collect the speech data (either a carbon-button or electret microphone). Since the speech data in the NIST 2000 database is of a conversational nature (i.e., there are no constraints placed on the spoken text) any speaker verification system that makes use of this database is inherently text-independent.

The set of targeted speakers in the NIST 2000 database consists of all speakers who initiated at least one call in which they spoke for at least two minutes. The training data for each targeted speaker was then collected by concatenating two minutes of speech from a single side of the conversation initiated by the speaker. Since speakers were required to initiate calls from different telephones, all other conversation sides involving a particular speaker implied the use of a handset different from the one used to collect the speaker's training data. Each side of all remaining conversations was then subsequently used to generate the test segments – a random interval of speech

was extracted from each conversation side and concatenated to form two test segments (one from each side of the conversation). The test segment durations varied from a few seconds to almost a minute, with the majority ranging between 15 and 45 seconds [111].

The one-speaker detection task for the NIST 2000 speaker recognition evaluation consisted of a set of index files which specified the verification trials to perform by pairing eleven targeted speakers with each test segment. The actual speaker was the targeted speaker in one of these eleven trials. Thus, there was about a ten-to-one ratio of non-target to target trials. A typical entry in an index file is shown in Figure 5-1. Here "gaaa" is the name of the test segment and the other eleven entries are the labels of the targeted speakers.

```
gaaa 9173 3775 3753 3334 3129 1791 1593 1474 1418 1346 1269
```

Figure 5-1: An example of an index file entry

The correct speaker to associate with the current test segment is specified by a file containing the answer keys[31]. A typical entry in this file is deciphered in Figure 5-2.

```
gaaa hgka 960530_1265_1346 b 1346 f et 22 elec tar diff trnhs 60.48 42 215542hpm ce pl 217.9 225
  1                        2   3   4  5  6   7          8          9                    10

Field #1:  1-speaker detection test segment name
Field #2:  channel side
Field #3:  speaker label/ID
Field #4:  gender
Field #5:  training segment handset type (et for ELECTRET, ct for CARBON-BUTTON)
Field #6:  age
Field #7:  test segment handset type (elec for ELECTRET, carb for CARBON-BUTTON)
Field #8:  specifies whether the same or different handsets where used
           (all different for the NIST 2000 evaluations)
Field #9:  duration of the 1-speaker detection test segment
Field #10: Swithboard phase
```

Figure 5-2: Description of the answer key fields

From Figure 5-2, the speaker labelled "1346" would be the correct speaker to associate with test segment "gaaa". To fully implement the one speaker detection task, one has to read in and parse all the index files and the file containing the answer keys.

In this work, the one-speaker detection task was performed in its entirety (i.e., all the 66572 verification trials specified for the NIST 2000 database were performed). As such, the system had to decide for each trial whether the targeted speaker is speaking in the given test segment. The evaluation tests both males and females separately (i.e., there are no cross-gender trials). However, all trials must be performed independently of each other. The likelihood ratio scores should

[31] For the NIST 2000 speaker recognition evaluation, this file is available at:
ftp://jaguar.ncsl.nist.gov/outgoing/sid2000.keys.tar.gz

all employ a common scale with larger values indicating a greater likelihood that a specified trial is in fact a target trial. This allows a speaker-independent decision threshold to be varied, thereby generating the full range of operating points for the system under consideration. Of the 66572 verification trials specified for the NIST 2000 database, more than 45000 involved training and test segments obtained from telephone handsets employing electret-type microphones. For this reason, the overall performance of any system evaluated on the NIST 2000 database is dominated by the performance obtained for these trials. A detailed description of the evaluation settings and rules may be found in [112].

According to Przybocki and Martin [111], the performance range for ten speaker verification systems evaluated on the NIST 2000 database was as follows. Under the conditions that (1) only male verification trials where the test segment duration is in the 15 to 45 second range are used; and (2) the test segment and targeted speaker training data come from conversation sides that employ electret-type microphones in the telephone handset, the EER varies between 7% and 18% and the minimum DCF values[32] vary between 250×10^{-4} and 600×10^{-4}. However, no details concerning the implementation of the speaker verification systems evaluated were provided. According to reference [111], verification trials involving females only resulted in slightly poorer results. Unfortunately, no performance figures where provided.

This section described the experimental database and protocol used to evaluate the baseline speaker verification system developed for this study. The remaining sections of this chapter describe the design, implementation and evaluation of this system.

5.2 System design and implementation

The baseline system developed in this study is based on the GMM-based speaker verification system described by Zilca in [113] and [114]. The reason being that Zilca not only employed many of the standard techniques discussed in Chapters 2 and 3, but also provided detailed information concerning his implementation. Furthermore, Zilca's system has been evaluated on the NIST 2000 database, which is the same database used in this study, and resulted in good overall performance (see Table 5-1).

The feature extraction procedure employed in Zilca's system is as follows. Each input speech signal was first segmented into 25 millisecond frames produced every 12.5 milliseconds which trans-

[32] The minimum DCF values specified in [111] were computed using Equation (2.24) with $C_{FR} = 10$, $P_L = 0.01$, $C_{FA} = 1$ and $P_I = 0.99$. To allow for a fair comparison, the same approach is taken in this work.

lates to a frame rate of 80 Hz. Each frame was then windowed with a Hamming window and passed through a voice activity detector that discarded about 50% of the frames. 18-dimensional MFCC feature vectors were then extracted from the remaining frames. As a final step, CMN was applied.

As far as speaker modelling is concerned, Zilca employed adapted GMMs. Four UBMs were trained depending on the handset type (i.e., carbon-button or electret) and gender of the speakers in the training data. Each UBM consisted of 512 Gaussian mixtures and was trained using about 2 hours of speech taken from the test portion of the database used in the NIST 1999 speaker recognition evaluation. The UBMs were trained using the *Distance-based GMM* (DB-GMM) procedure discussed in [115]. This procedure partitions the feature vectors into clusters using the k-means algorithm (see reference [116] for a good description of how this algorithm works). The parameters of a GMM are then calculated as follows. The mixture weights are given by the number of feature vectors in each cluster divided by the total number of feature vectors, and the means and variances are simply the sample mean and variance of the vectors in each cluster. No iterations of the EM algorithm were performed. In [115], the DB-GMM procedure was shown to be simpler in its implementation than the EM algorithm and produced performance comparable to that of GMMs trained with the EM algorithm. Speaker models were obtained by adapting the parameters of one of the four UBMs using a Bayesian adaptation procedure (see Section 2.3.3). The choice of the UBM to adapt depended on the handset type of the speaker's training data and the speaker's gender. Finally log-likelihood scores were obtained by retaining the scores of 5 highest scoring mixtures in the UBM and the corresponding speaker model (see Section 2.3.3). The performance obtained by Zilca's system when evaluated on the NIST 2000 database is as follows [113, 114]:

Table 5-1: EER obtained by Zilca's system under different training and test conditions

Training and test conditions	**EER**
The training and test data both come from telephone handsets with electret microphones (elec/elec)	14.7%
The training and test data both come from telephone handsets with carbon-button microphones (carb/carb)	19.5%
The training data comes from telephone handsets with electret microphones while the test data comes from handsets with carbon-button microphones (elec/carb)	21.2%
The training data comes from telephone handsets with carbon-button microphones while the test data comes from handsets with electret microphones (carb/elec)	29.1%
Combined performance for all trials	17.3%

As tabulated, for the elec/elec condition, Zilca's system produced an equal error rate of 14.7% for both male and female verification trials which is well within the 7% to 18% EER range typical of

a number of systems evaluated on the NIST 2000 database. Unfortunately, Zilca does not provide any minimum DCF scores for his system. Table 5-1 also shows that, as expected, under matched handset-type conditions (i.e., elec/elec and carb/carb) the system's performance is better than that obtained under mismatched handset-type conditions (i.e., elec/carb and carb/elec). Furthermore, the performance of the system is best for the elec/elec condition and worst for the carb/elec condition. This is as a result of the low quality and non-linearity of carbon-button microphones which degrade the quality of the speaker models trained with speech collected from them.

At this point it should be noted that even when the training and test data both come from telephone handsets employing the same type of microphone, as in the case of the elec/elec and carb/carb conditions, the exact same telephone handsets are not used in the collection of both the training and test data (see Section 5.1.2). In addition, other types of mismatch, including channel noise from various noise sources, recordings from different locations and recordings collected over a period of time are also present in the data.

The baseline system described in this chapter is aimed at emulating the architecture and performance of the system developed by Zilca. As such, many of the parameter choices are kept the same (such as the dimension of the MFCC feature vectors for example). Also, if the baseline system developed for this thesis obtains a performance comparable to that of Zilca's system, it would verify that the implementation of the baseline system is indeed correct.

The architecture of the baseline system developed for this thesis is depicted in Figure 5-3. As illustrated, the system consists of 3 modules. The first module, the feature extraction module, contains all the signal processing steps required to extract features from the input speech signal and is in operation in both the training and test modes. The second module, the speaker modelling module, is responsible for creating speaker models from the input feature vectors during training and uses these models to populate a model database. The third module, the decision-making module, operates in the test mode and is responsible for determining whether the input speech signal indeed emanated from the targeted speaker. The implementation of these modules is discussed in more detail in the following sections.

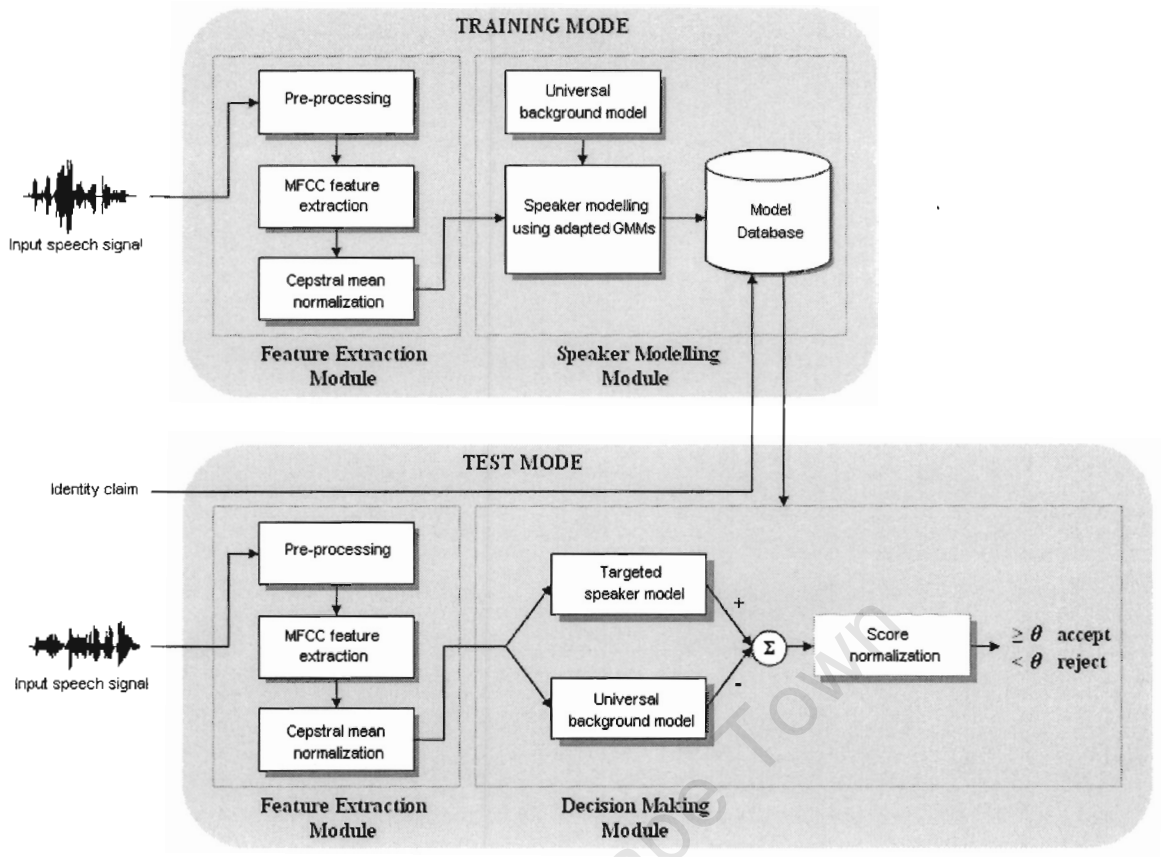

Figure 5-3: The baseline system architecture

5.2.1 Implementation of the feature extraction module

The purpose of this module is to convert the input speech signal into a compact and efficient representation that is more stable and discriminative than that of the original signal. As illustrated in Figure 5-3 this module consists of 3 components namely, pre-processing (see Section 2.1.1), MFCC feature extraction (see Section 2.1.2.2) and cepstral mean normalization (see Section 3.2.1). For the implementation of the pre-processing component, the speech signal was first filtered with a pre-emphasis filter of the form $H(z) = 1\text{-}0.97z^{-1}$ and partitioned into 25 millisecond frames at a frame rate of 80 Hz. These frames were then multiplied with a Hamming window to minimise signal discontinuities at the start and end of each frame, and passed through a voice activity detector (VAD). The purpose of the VAD is to eliminate all frames primarily containing silence, noise or unvoiced speech. The VAD was implemented as a simple energy-based detector that discarded all frames below a specified energy level. About 30% of all frames were discarded.

The remaining frames were then forwarded to the next component where the extraction of mel-frequency cepstral coefficients took place as follows. Each frame was first Fourier transformed into the frequency domain. The squared magnitude spectrum of each frame was then filtered by a bank of 26 mel-scaled triangular filters distributed over the frequency band of 240-3480 Hz

(which is approximately the bandwidth of the telephone channel). The logarithm of the filterbank outputs were then cosine transformed into 18-dimensional MFCC feature vectors. Finally cepstral mean normalization was applied to compensate for the linear filtering effects of telephone channels. The value of the parameters selected for the implementation of the feature extraction module were either determined empirically (such as the value of the pre-emphasis coefficient for example) or selected in accordance to those employed by Zilca in his system (such as the order of the MFCC feature vectors for example).

5.2.2 Implementation of the speaker modelling module

The purpose of the speaker modelling module is to create a model of each speaker's speech characteristics from the features generated by the feature extraction module. For this module, the training procedure for speaker modelling using adapted GMMs, as described in Section 2.3.3, was fully implemented. Four handset- and gender-dependent UBMs (i.e., a male-electret UBM, a male-carbon UBM, a female-electret UBM and a female-carbon UBM) were trained with about 8 hours of speech (2 hours of speech per model) taken from the test portion of the database used in the NIST 1999 speaker recognition evaluation[33]. Each GMM-based UBM consisted of 512 Gaussian mixtures with diagonal covariance matrices (see Section 2.3.1). Similarly to Zilca's system, the DB-GMM procedure was used to train the four UBMs. As mentioned before, this procedure clusters the feature vectors extracted from the UBM training data using the *k*-means algorithm and then calculates the weights, means and variances of each of these clusters. These values are then stored as the final GMM parameters. Twenty-five iterations of the *k*-means algorithm were performed. No iterations of the EM algorithm were performed. Subsequently, speaker models were obtained by adapting the parameters (i.e., the weights, means and variances) of a particular UBM by using the speaker's training data and the Bayesian adaptation procedure described in Section 2.3.3. The UBM selected for adaptation depended on the gender of the speaker in the training data as well as the type of microphone employed in the telephone handset used to obtain the training data.

5.2.3 Implementation of the decision-making module

As depicted in Figure 5-3, the decision-making module is based upon the likelihood ratio test discussed in Section 2.2. During testing, this module compares the features generated by the feature extraction module to the targeted speaker model (i.e., the speaker model associated with the input identity claim) and the corresponding UBM. The difference between the log-likelihood scores

[33] This was done to avoid introducing a bias into the results (see reference [7])

obtained is then compared to a decision threshold, θ, to determine whether to accept or reject the identity claim. This module also includes an extra component termed "score normalization" to normalize the log-likelihood scores prior to making the final decision. This is merely to illustrate where score normalization fits into the overall system but, is not active at this stage of the implementation. The effect of score normalization is examined in Section 5.3.3.

The testing procedure for the targeted speaker models (i.e., the adapted speaker models) and UBM pairs is as discussed at the end of Section 2.3.3. That is, for each feature vector, the 5 highest scoring mixtures in a gender- and handset-dependent UBM were found and an estimate of the UBM log-likelihood was computed using only the scores obtained for these mixtures. The same feature vectors were then scored against the 5 corresponding mixtures in the targeted speaker model to obtain an estimate of the log-likelihood for the targeted speaker model. The difference between these two scores was then obtained and formed the log-likelihood ratio score for the speaker model and UBM pair. This same testing procedure was employed for each verification trial specified for the NIST 2000 database. At completion, all the log-likelihood ratio scores were pooled and compared to a varying decision threshold so as to obtain the full range of operating points (including the EER) for the baseline speaker verification system[34].

Section 5.2 has covered the design and implementation of the baseline speaker verification system. In the following section the performance of this system is evaluated. However, before proceeding to the next section, it is important to clarify which software components were actually implemented by the author. General signal processing utilities such as the FFT, DCT, Hamming and triangular windows were provided by Dr. Jialong He in his speaker verification library[35]. He also provides functions that implement various feature sets and classifiers, but these were not used due to their lack of modifiability. The code to implement the *k*-means algorithm was provided by the author's supervisor, Dr. Daniel Mashao. All the other software components, such as the code to extract MFCCs, the code to implement adapted GMMs and their testing procedure, the code to implement cepstral mean normalization, the code to perform likelihood ratio testing, the code to perform score normalization and the code to combine all the modules depicted in Figure 5-3, were implemented by the author. The majority of the code was implemented in C++. MATLAB was used to analyse the results obtained. In the following section the baseline speaker verification system is evaluated.

[34] The decision threshold was varied over a range that marginally exceeded the minimum and maximum values of the pooled log-likelihood ratio scores.

[35] This speaker verification library is available at: http://tiger.la.asu.edu/download/svlib_pc.zip

5.3 System evaluation

In this section, the implementation of the baseline speaker verification system, as described in the previous section, is evaluated. In particular, experiments were done to verify that the implementation of the baseline system is indeed correct. This was done by determining whether its performance is comparable to that of Zilca's system or not. In addition, the effect of adapting different UBM parameters, score normalization, different GMM model orders and cepstral mean normalization was examined. This was done in order to consolidate the observations reported by other researchers in contemporary literature and, to establish a suitable trade-off between computational complexity and system performance. For all the results tabulated in this section, each experiment was performed at least three times so as to obtain a more accurate estimate of the average performance of the baseline system under certain conditions. In each experiment the initial GMM means were chosen randomly.

5.3.1 System verification

In this section, experiments were done in order to verify the correct implementation of the baseline system. Since the architecture and implementation of the baseline system resembles that of Zilca's system, its performance should be comparable to that of Zilca's system. As such, the performance obtained by Zilca's system, as tabulated in Table 5-1, was used as a benchmark for verifying the correct implementation of the baseline system. As for Zilca's system, all 66572 verification trials (6096 target trials and 60476 non-target trials), specified for the NIST 2000 database, were performed. The result of this experiment is depicted in Figure 5-4.

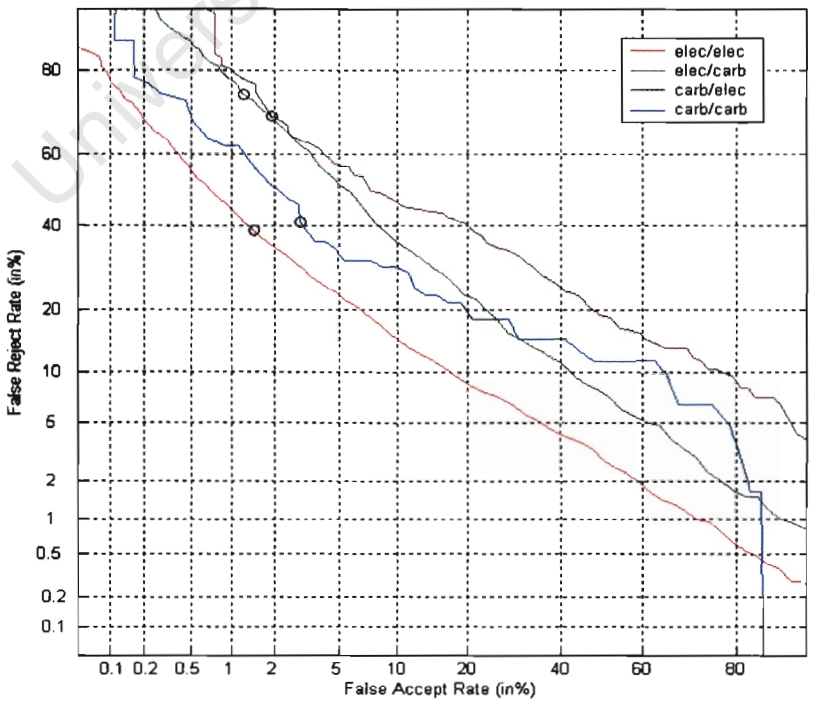

Figure 5-4: DET curves for the baseline system under different training and test conditions

As illustrated, the trend in the results is similar to that reported for Zilca's system, as for matched handset-type conditions (i.e., elec/elec and carb/carb) the overall system performance is above that obtained under mismatched handset-type conditions (i.e., elec/carb and carb/elec). In addition, the performance for the elec/elec condition is significantly better than that obtained under all other training and test conditions, with the carb/elec condition producing the worst performance. As mentioned before, this degradation in performance can primarily be attributed to the low quality and non-linearity of carbon-button microphones. The experiment that resulted in the performance depicted in Figure 5-4 was repeated three times and the average performance of the baseline system was as follows:

Table 5-2: Baseline system performance under different training and test conditions (average ± standard deviation)

Training and test conditions	EER	Minimum DCF ($x10^{-4}$)
elec/elec	12.59 ± 0.11%	535 ± 3.61
carb/carb	20.48 ± 0.53%	693 ± 7.57
elec/carb	21.42 ± 0.38%	875 ± 1.00
carb/elec	30.96 ± 0.50%	884 ± 5.00
Combined performance for all trials	15.36 ± 0.12%	614 ± 2.52

From Table 5-2 it is clear that the baseline system developed for this thesis outperforms Zilca's system (see Table 5-1). The combined EER across all trials of 15.36% is 11.21% lower than that obtained by Zilca's system. Furthermore, the minimum DCF value of $535x10^{-4}$ for the elec/elec condition falls well within the $250x10^{-4}$ to $600x10^{-4}$ minimum DCF range obtained by other systems evaluated on the NIST 2000 database under similar training and test conditions. The discrepancy between the results obtained for the baseline system and those obtained by Zilca's system can be attributed to the fact that Zilca employed a VAD that discarded about 50% of all the speech frames whereas for the baseline system developed here, only 30% of all frames were discarded. In the following section the effect of adapting different UBM parameters when training speaker models is examined.

5.3.2 The effect of adapting different UBM parameters

In [60], Reynolds et al. empirically showed that adapting different combinations of UBM parameters, when training adapted speaker models (see Section 2.3.3), leads to variations in speaker verification performance. Adaptation of the means alone produced the best overall results. Unfortunately, no theoretical reasons were provided for these observations. In Zilca's system, all the parameters (i.e., the weights, means and variances) of the UBMs were adapted. Initially, this same approach was adopted for the baseline system developed in this study. In this section, the results

of experiments conducted to determine whether any improvement can be obtained by only adapting the means of the UBMs used in the baseline system when training speaker models are reported on. The combined performance of the baseline system when all the parameters are adapted and, when only the means are adapted, is illustrated in Figure 5-5. From this figure it is clear that adaptation of the means alone results in a significant improvement in the performance of the baseline system.

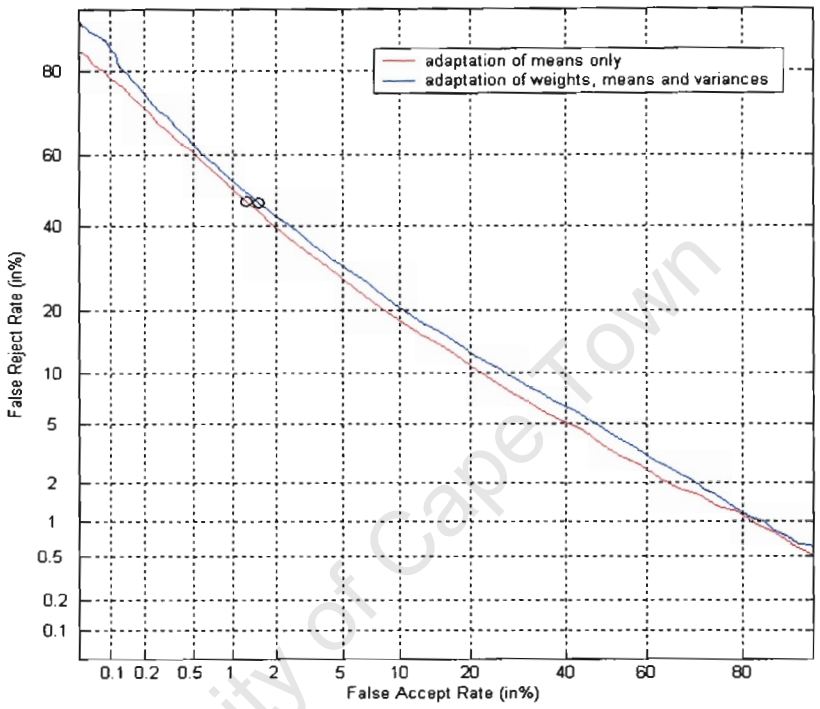

Figure 5-5: The effect of adapting different UBM parameters

The experiment that resulted in the performance depicted in Figure 5-5 was repeated three times and the average performance of the baseline system was as follows:

Table 5-3: Combined performance for all trials obtained by adapting different UBM parameters (average ± standard deviation)

Parameters adapted	EER	Minimum DCF ($x10^{-4}$)
Weights, means and variances	15.36 ± 0.12%	614 ± 2.52
Means only	14.30 ± 0.11%	592 ± 2.00

The results tabulated in Table 5-3 show that for the baseline system, there is a relative reduction in the EER of 6.90% and a relative reduction in the minimum DCF value of 3.58% when only the means of the UBMs used were adapted. This consolidates the observations of Reynolds et al. in [60] that when training adapted speaker models, adaptation of the UBM means alone produces

performance superior to that of adaptation of all the UBM parameters. For rest of the experimental results reported on in this document, only the means of the UBMs used when training adapted speaker models were adapted. In the following section, the impact of score normalization on the baseline system's performance is examined.

5.3.3 The effect of score normalization

As mentioned in Section 3.3, the setting of stable speaker-independent decision thresholds is a very challenging task due to the score variability caused by mismatched training and test conditions. In this section, experiments are conducted to determine whether score normalization, in the form of T-norm (see Section 3.3.3), can be used to improve the performance of the baseline system. In order to implement T-norm, each test segment in the NIST 2000 database was scored against a number of gender-dependent impostor models trained with data taken from the NIST 1999 database. From the scores obtained, two mean and standard deviation estimates were computed and used to normalize the log-likelihood ratio scores obtained for each of the 66572 verification trials specified for the NIST 2000 database. The mean and standard deviation estimates to use for score normalization were selected according to the gender of the speaker in a specific verification trial. Figure 5-6 shows the performance obtained for the baseline system as the number of impostors used in the T-norm formulation was increased:

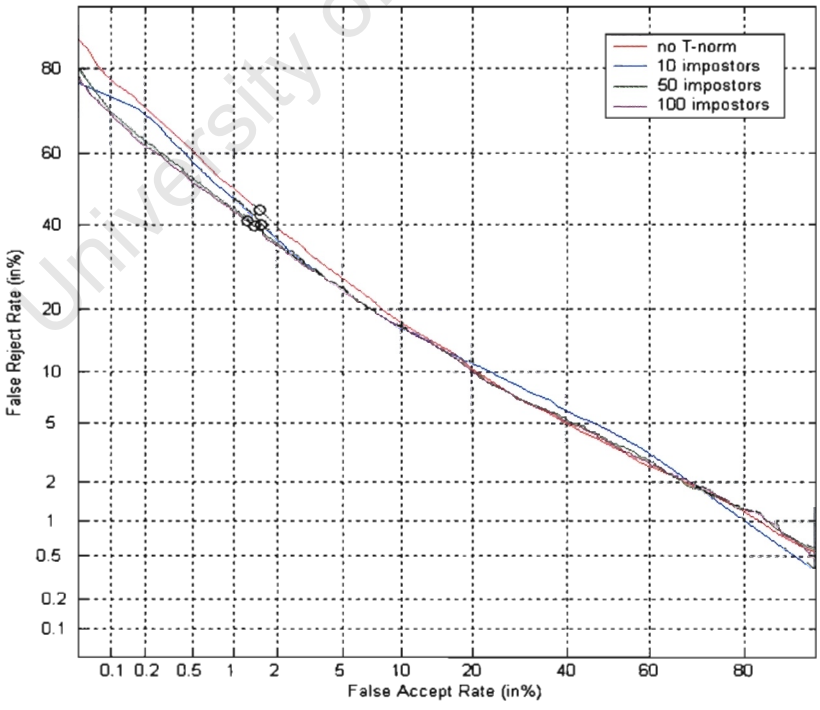

Figure 5-6: DET curves for the baseline system with and without T-norm

The experiment that resulted in the performance depicted in Figure 5-6 was repeated three times and the average performance of the baseline system was as follows:

Table 5-4: Combined performance for all trials as the number of impostors used for T-norm is increased (average ± standard deviation)

Number of Impostors	EER	Minimum DCF ($x10^{-4}$)
10	13.99 ± 0.20%	555 ± 9.85
50	13.89 ± 0.58%	543 ± 4.58
100	13.90 ± 0.09%	533 ± 7.77
No T-norm	14.30 ± 0.11%	592 ± 2.00

From Table 5-4 it is clear that the application T-norm with as few as 10 impostors leads to an improvement in the performance of the baseline system. Figure 5-6 shows that this improvement is largest at low false accept rates. However, at low false reject rates, a degradation in the performance of the baseline system is observed. This can primarily be attributed to inaccurate estimates of the mean and standard deviation parameters required for T-norm. However, increasing the number of impostors to 50 improved the overall system performance across most operating points as this led to more accurate estimates of the mean and standard deviation parameters. Increasing the number of impostors to 100 however, did not result in any discernible improvement in the EER of the system obtained when 50 impostors were used. However, a minor reduction in the minimum DCF value was observed. Thus, for rest of the experimental results reported on in this document, T-norm with only 50 impostors is used to normalize all the log-likelihood scores obtained. At this point, it should be noted that although T-norm does provide a gain in system performance, it is at the expense of increased memory and computational resources. In the following section the variation in the performance of the baseline system, when using GMMs with different model orders, is evaluated.

5.3.4 The effect of different model orders

According to Reynolds and Rose [29], determining the correct number of mixtures to use in a GMM (i.e., the model order) is "*an important but difficult problem*" for the following two reasons: (1) The selection of too few mixtures could produce a speaker model that does not accurately capture and model the distinguishing characteristics of the underlying distribution of feature vectors extracted from a particular speaker's speech. (2) The selection of too many mixtures would require more memory and computational resources, and could reduce performance when there are a large number of model parameters relative to the available training data. For these reasons, the selection of the correct number of mixtures to use in the baseline system was examined. In particular, the trade-off between computation time and system performance is reported on.

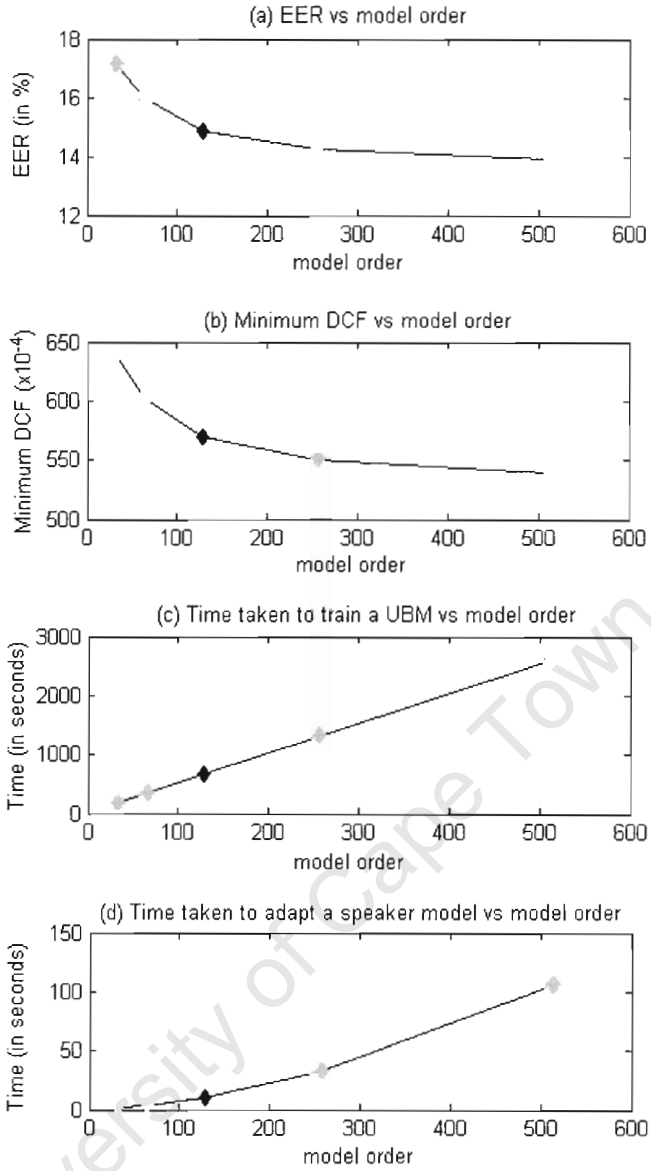

Figure 5-7: System performance and computation time versus the model order of the UBMs and adapted speaker models

Figures 5-7(a) and 5-7(b) depict the combined performance of the baseline system across all training and test conditions as the model order of the UBMs and adapted speaker models used is increased from 32 to 512 mixtures. As illustrated, increasing the model order from 32 to 512 mixtures leads to a definite increase in the performance of the baseline system as the EER is reduced from 17.13% to 13.89% and the minimum DCF value is reduced from 639×10^{-4} to 543×10^{-4}. However, Figures 5-7(c) and 5-7(d) show that this improvement in performance is at the expense of increased computation time. Furthermore, there is a larger increase in the performance of the baseline system, and a smaller increase in the computation time, from 32 to 128 mixtures than from 128 to 512 mixtures. As illustrated by Figures 5-7(c) and 5-7(d), there is a linear increase in the time taken to train a UBM and a super-linear increase in the time taken to adapt a speaker

model when the model order is increased. These results were averaged over three runs and were obtained on a computer running the Windows XP operating system on a 2.8 GHz Pentium 4 processor with 500 MB of RAM. The UBM training data consisted of about 2 hours of speech taken from the NIST 1999 database and, the speaker training data consisted of about 2 minutes of speech taken from the speaker labelled 1018 in the NIST 2000 database.

The combined performance of the baseline system when using only 128 mixtures (i.e., an EER = 14.87% and minimum DCF = 569×10^{-4}) is not only better than the performance obtained by Zilca's system but reduces the time required to train the 512 mixture UBMs by 74.21% and the time required to adapt speaker models by 90.48%. This computational saving becomes even more significant when one considers the number of UBMs (4) and the speaker models (1003) that need to be trained (this excludes the speaker models required for T-norm). As such, for rest of the experimental results reported on in this document, GMMs with 128 mixtures were used. In the next section the effect of cepstral mean normalization on the performance of the baseline system is evaluated.

5.3.5 The effect of cepstral mean normalization

The purpose of this section is to determine the effect that the absence of CMN has on the performance of the baseline system. Recall from Section 3.2.1 that CMN is generally aimed at compensating for the linear filtering effects of telephone channels, and to some extent, the effects of

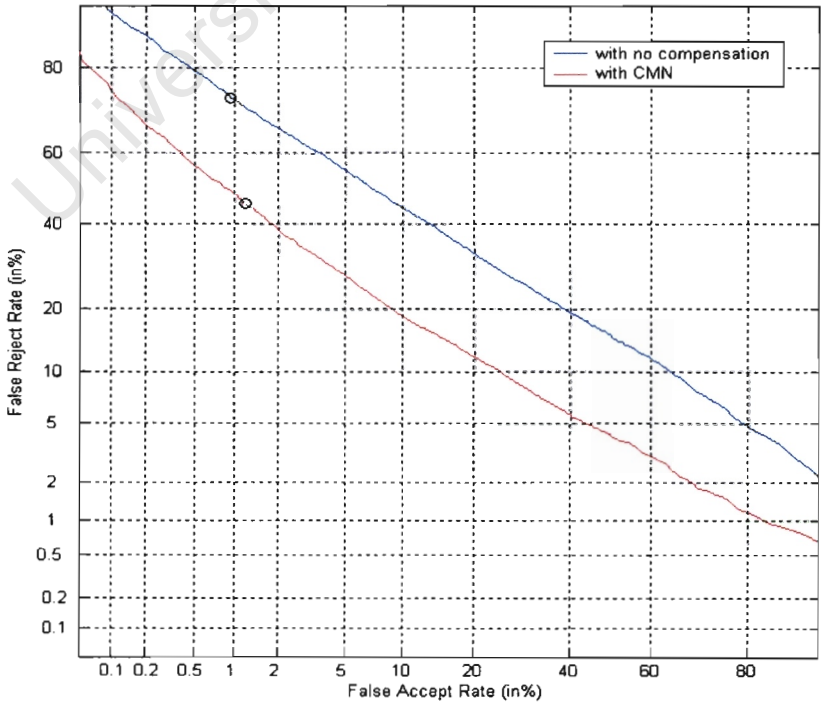

Figure 5-8: DET curves for the baseline system with and without CMN

additive noise and intersession variability. The combined performance for the baseline system with and without CMN is depicted in Figure 5-8. As illustrated, the application of CMN leads to a significant improvement in the performance of the baseline system. This confirms the knowledge that CMN improves the performance of speaker verification systems operating in telephone environments. The experiment that resulted in the performance depicted in Figure 5-8 was repeated three times and the average performance of the baseline system was as follows:

Table 5-5: Baseline system performance with and without CMN
(average ± standard deviation)

	With CMN		Without CMN	
Training and test conditions	**EER**	**Minimum DCF ($x10^{-4}$)**	**EER**	**Minimum DCF ($x10^{-4}$)**
elec/elec	12.79 ± 0.13%	491 ± 4.73	25.56 ± 0.12%	790 ± 2.52
carb/carb	21.31 ± 2.82%	585 ± 22.7	40.38 ± 1.30%	811 ± 14.0
elec/carb	20.45 ± 0.34%	792 ± 7.81	30.50 ± 0.35%	926 ± 2.52
carb/elec	27.43 ± 0.26%	811 ± 6.51	35.44 ± 0.83%	942 ± 7.67
Combined performance for all trials	14.87 ± 0.11%	569 ± 5.03	26.84 ± 0.01%	834 ± 3.61

As tabulated in Table 5-5, CMN improves the performance of the baseline system across all training and test conditions. For the combined performance of the baseline system across all verification trials, a relative improvement of 44.60% in the EER and 31.77% in the minimum DCF value was observed.

This concludes Section 5.3 in which various aspects of the baseline system, including its implementation and techniques for improving its performance, were evaluated. The final results for the baseline system, as well as how it compares with the original system developed by Zilca, is given in Table 5-6.

5.4 Summary

This chapter described the design, implementation and evaluation of a baseline text-independent speaker verification system. This system will be used in the following chapter as an experimental framework for evaluating the HEQ technique. Furthermore, the performance of this system will also be used as a benchmark against which all subsequent improvements will be compared. The system is based on the speaker verification system developed by Zilca in [113] and [114]. The reason being that Zilca not only employed many of the contemporary techniques discussed in Chapters 2 and 3 but, also provided detailed information concerning his implementation. Furthermore, Zilca's system has been evaluated on the NIST 2000 database, which is the same data-

base used in this study and, resulted in good overall performance. From the experiments conducted in this chapter, the performance of the baseline system was shown to be comparable to that of Zilca's system and other systems evaluated under similar conditions.

The characteristics of the NIST 2000 database, as well as the procedure for using this database were also covered. Furthermore, the experiments conducted in this chapter also consolidate the knowledge that: (1) when training adapted GMM speaker models by adapting the parameters of a UBM, adaptation of the means alone results in performance superior to that of adaptation of all the UBM parameters; (2) score normalization minimises score variability which leads to improved performance; and (3) CMN can be used to improve the performance of speaker verification systems operating in telephone environments. The following chapter is aimed at evaluating HEQ using the experimental framework (i.e., the baseline text-independent speaker verification system) described in this chapter.

Table 5-6: The final parameters and performance of the baseline system

	Parameter	**Zilca's system**	**The baseline system**	
Mel-frequency cepstral coefficients (MFCC)	Feature order	18	18	
	Bandwidth	Not specified	240-3480 Hz	
	No. of filters	Not specified	26	
	% frames discarded	50%	30%	
	Frame size	25 milliseconds	25 milliseconds	
	Frame rate	80 Hz	80 Hz	
	Pre-emphasis filter	Not specified	$H(z) = 1 - 0.97z^{-1}$	
	Feature compensation	CMN	CMN	
	Score normalization	None	T-norm	
Adapted GMM speaker models	Model order	512	512	
	Parameters adapted	Weights, means and variances	Means only	
	UBM type	Handset and gender-dependent	Handset and gender-dependent	
System performance (average ± standard deviation)	**Training and test conditions**	**EER**	**EER**	**Minimum DCF ($x10^{-4}$)**
	elec/elec	14.7%	12.79 ± 0.13%	491 ± 4.73
	carb/carb	19.5%	21.31 ± 2.82%	585 ± 22.7
	elec/carb	21.2%	20.45 ± 0.34%	792 ± 7.81
	carb/elec	29.1%	27.43 ± 0.26%	811 ± 6.51
	Combined performance for all trials	17.3%	14.87 ± 0.11%	569 ± 5.03

Chapter 6

Experimental Results and Analysis

This chapter is aimed at experimentally evaluating the Histogram Equalization technique described in Chapter 4. The results of the experiments are analysed and possible explanations for any gains (or degradations) observed, are provided. In Section 6.2, experiments which determine the optimal parameter values to use for the HEQ technique, when evaluated on the NIST 2000 database, are conducted. Section 6.3 compares HEQ to other feature-based compensation techniques and, in Section 6.4 various ways of applying HEQ to an utterance are evaluated. Section 6.5 is aimed at determining whether the use of multimodal reference histograms, instead of unimodal histograms, makes any difference to the performance of HEQ. Finally in Section 6.6, HEQ is applied to a combined feature set, namely, mel-frequency cepstral coefficients concatenated with a pitch-based feature set known as the Maximum Autocorrelation Values. This is done so as to determine whether HEQ indeed has the ability to compensate for distortions regardless of the speech parameterisation used. For all the results tabulated in this chapter, each experiment was performed at least three times so as to obtain a more accurate estimate of the average performance of the baseline system under certain conditions.

When mapping the feature distributions of different speakers to a common reference distribution, one would expect speaker verification performance to degrade due to feature distributions of different speakers occupying the same region in the feature space. However, after the application of HEQ, the set of features extracted from the speech of different speakers remains dissimilar and will still result in different cluster patterns in the feature space. This is the reason that more inter-speaker confusions do not occur when HEQ is used to map the feature distributions of different speakers to a common reference distribution. In fact, as this chapter will show, HEQ improves the robustness of a speaker verification system operating in telephone environments (i.e., one evaluated on speech contaminated by telephone transmission). The following section is aimed at de-

termining whether the algorithm described in Section 4.5 indeed allows one to map one histogram to another.

6.1 Algorithm verification

In order to verify that the HEQ algorithm provided in Section 4.5 indeed allows for the mapping of one histogram to another, HEQ was used to equalize the histograms of the clean and contaminated log-energies obtained by the Monte Carlo simulation performed in Section 3.1.2. Figure 6-1(a) shows the clean log-energy histogram and its contaminated version caused by the application of additive noise and a linear filtering effect. When compared to the clean log-energy histogram, the contaminated log-energy histogram exhibits a shift in its mean, a reduction in variance and a positive skew. Figure 6-1(b) shows the non-linear transformation that was obtained by matching the cumulative histograms of the two log-energy histograms – the histogram of the clean log-energy values was used as the reference for the HEQ technique. When transforming all the contaminated log-energy values using the transformation depicted in Figure 6-1(b), the compensated log-energy histogram depicted in Figure 6-1(c) was obtained. For comparative purposes, the clean log-energy histogram is also depicted. From this figure it is clear that HEQ was able to compensate for both linear and non-linear distortions of the feature space, which resulted in a closer match between the clean and contaminated log-energy histograms.

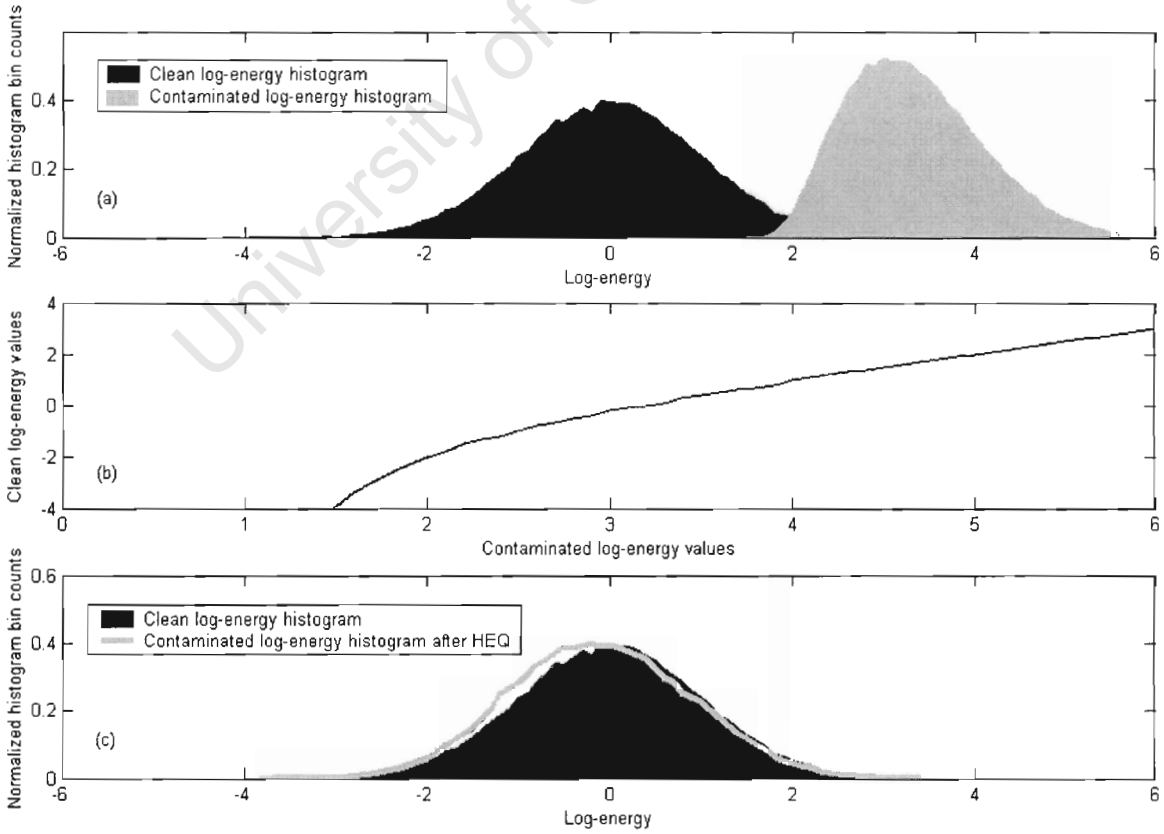

Figure 6-1: Application of HEQ to restore a corrupted log-energy histogram

To determine whether these observations also pertain to data degraded by real-world distortions, the HEQ algorithm was applied to the MFCCs obtained from the simulation performed in Section 3.1.3. Recall that in Section 3.1.3 MFCCs were extracted from both clean (TIMIT) and contaminated (NTIMIT) versions of the same utterance, so as to give some insight into how degradations attributed to telephone transmission distort feature distributions. The histograms of the first MFCC feature vector component, $MFCC_1$, extracted from both the clean (TIMIT) utterance and the contaminated (NTIMIT) utterance are displayed in Figures 6-2(a) and 6-2(b) respectively. As illustrated, the two histograms differ in their shape, scale, spread and location. Here, HEQ was used to equalize the two histograms so as to compensate for the degradations caused by telephone transmission. The reference distribution was chosen to be Gaussian with zero mean and unity variance. The cumulative histograms of the clean and contaminated $MFCC_1$ distributions were estimated according to steps (1) to (5) of the HEQ algorithm described in Section 4.5. All histograms were estimated using 100 uniformly spaced intervals between the minimum and maximum values of the respective $MFCC_1$ values. Figures 6-2(c) and 6-2(d) show the clean and contaminated histograms after the application of HEQ respectively. Not only do the clean and contaminated $MFCC_1$ histograms appear to be more alike in terms of their overall shape, scale, spread and location but, the two histograms are also very similar to a Gaussian distribution with zero mean and unity variance.

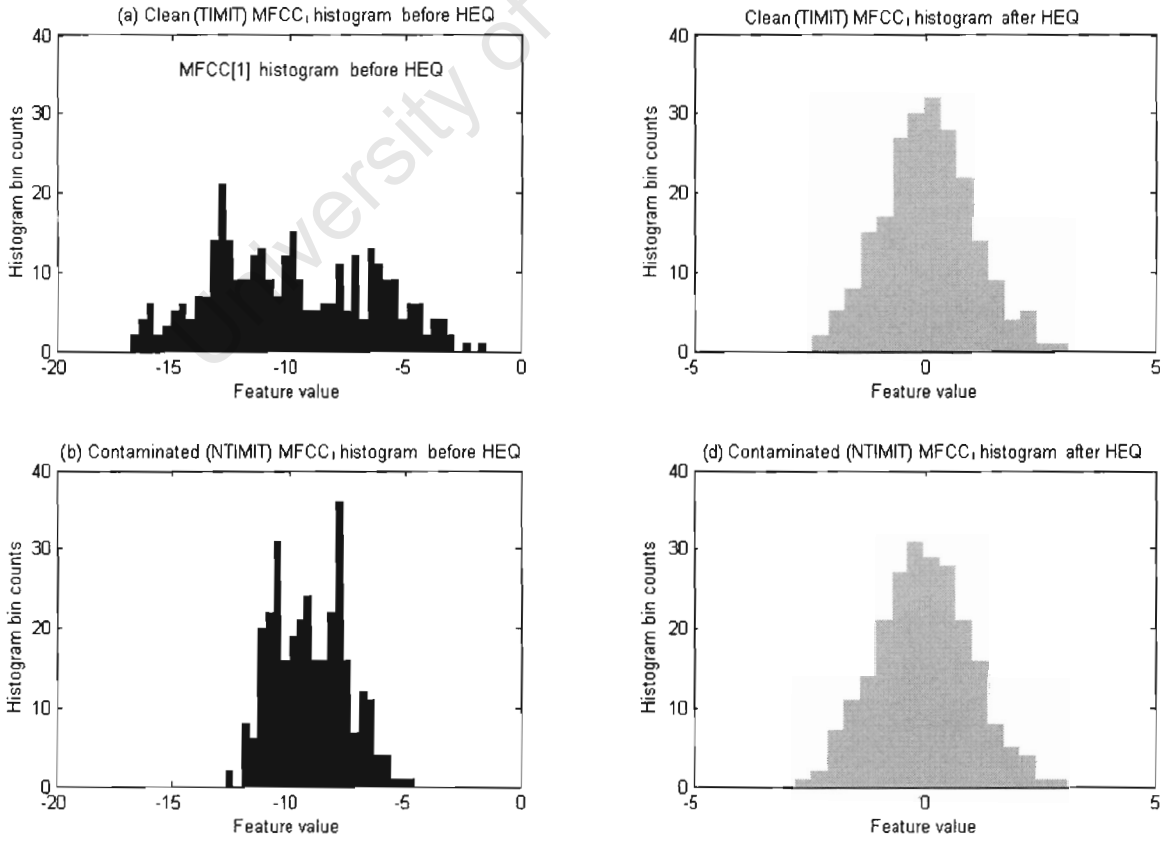

Figure 6-2: $MFCC_1$ histograms before and after the application of HEQ

A plot of the trajectory (i.e., the time sequence) of the first 250 $MFCC_1$ values extracted from the TIMIT and NTIMIT test utterance is shown in Figure 6-3. From this figure, it is clear that the histogram equalized feature trajectories (Figures 6-3 (c) and (d)) are more similar than their unequalized counterparts (Figures 6-3 (a) and (b)). Figures 6-1 to 6-3 reinforces HEQ's ability to make MFCC distributions more consistent across different recording conditions, and verifies that the HEQ algorithm provided in Section 4.5 maps one histogram to another (and was implemented correctly).

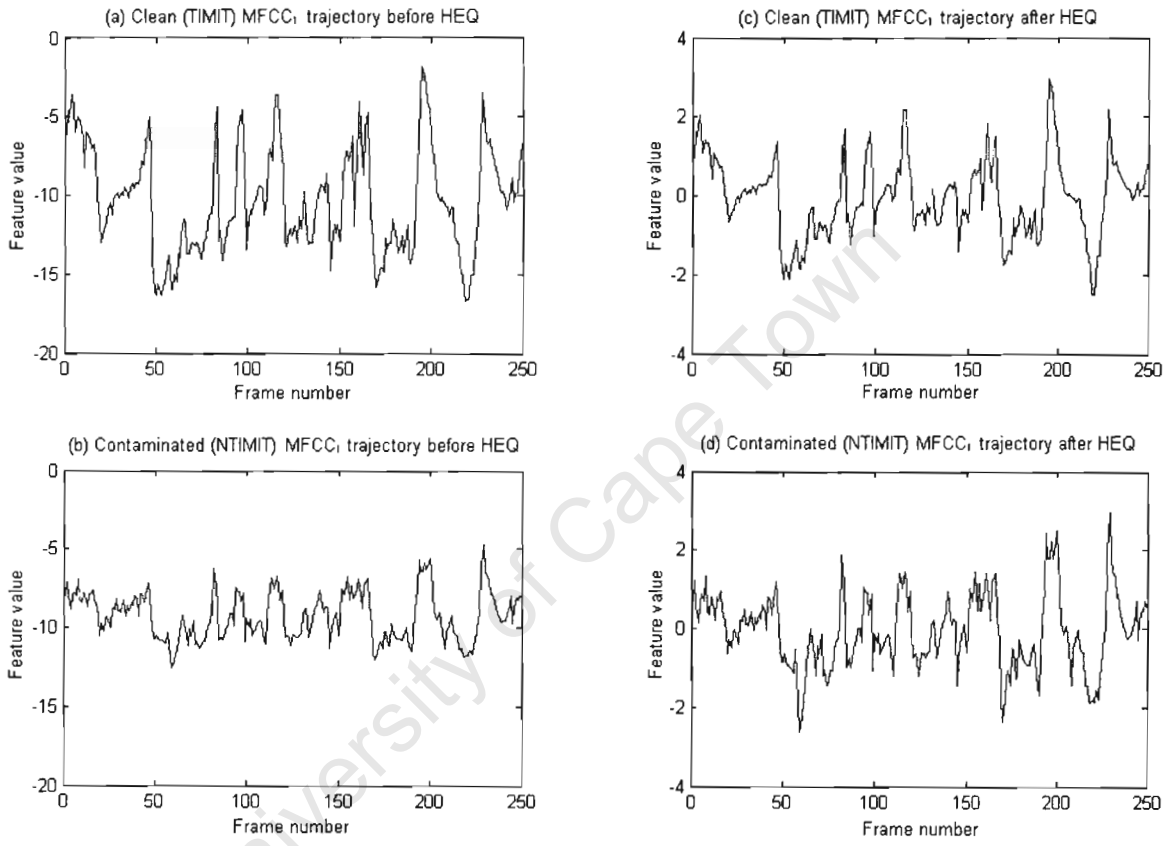

Figure 6-3: $MFCC_1$ trajectories before and after the application of HEQ

6.2 Parameter optimization

In this section the optimal parameter values to use for the Histogram Equalization technique, when applied to MFCCs extracted from speech data in the NIST 2000 database, were determined. The experimental setup of the speaker verification system is exactly the same as the baseline system described in Chapter 5. The only difference is that the cepstral mean normalization component (see Figure 5-3) was replaced by Histogram Equalization. Initially, the reference distribution was chosen to be Gaussian with zero mean and unity variance. HEQ was applied utterance-wise (i.e., it was applied over the entire duration of each speaker's training and test utterances). The distribution of each MFCC feature vector component was processed separately.

The first set of experiments was conducted in order to determine the optimal number of bins to use in estimating the required histograms. All 66572 verification trials specified for the NIST 2000 database were performed. According to Segura et al. [91], the "*number of bins used in the estimation of the cumulative histograms must be selected taking into account the trade-off between smoothness and resolution of the cumulative histograms*". In other words, the more bins that are used, the more accurate the histogram estimates but, the less smooth the resulting cumulative histograms will become. The number of bins was increased from 100 and 2000. Each of these experiments was repeated three times and the average performance is depicted in Figure 6-4.

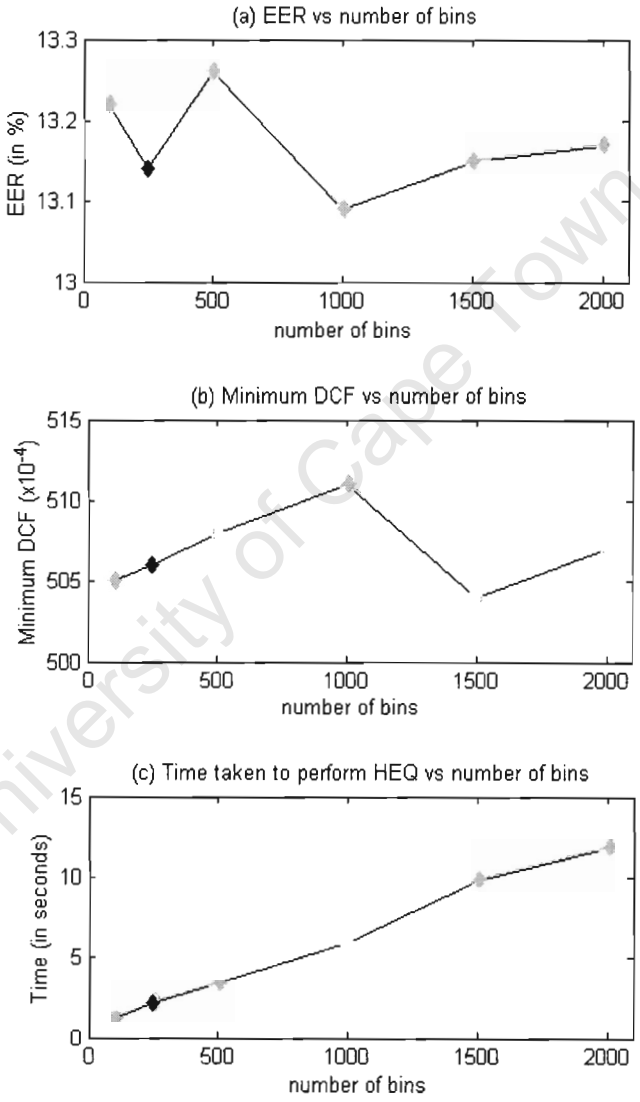

Figure 6-4: System performance versus the number of bins used for HEQ

From Figures 6-4(a) and 6-4(b), it appears as though the use of 1000 bins leads to the lowest EER (13.09%) but at the expense of an increased minimum DCF value (511×10^{-4}). On the other hand, the use of 1500 bins leads to the lowest minimum DCF value of 504×10^{-4}. Overall however, there is not any substantial variation in the performance obtained as the number of bins is increased.

From Figure 6-4(c), it is clear that the larger the number of bins used, the longer the time taken to perform HEQ. This observation is intuitive as more bins result in more (time consuming) computations being performed. When taking the trade-off between system performance and execution time into account, the smallest number of bins that produces the best overall performance should be selected. For this study, 250 bins were selected as it not only led to an EER (13.14%) and minimum DCF value (506×10^{-4}) comparable to the best obtained, but also resulted in the second lowest computation time. Furthermore, according to McNemar's test[36], the difference between the EERs, when 250 bins are used instead of 1000 bins, was found not to be statistically significant.

After determining the number of bins to use for HEQ, a set of experiments were conducted in which the variance of the reference Gaussian distribution was varied. This was done so as to determine whether any gains are obtained when altering the reference distribution for the HEQ technique. The mean of the distribution was fixed at zero so as compensate for linear filtering effects (and some of the effects of additive noise). In a number of the papers reviewed in Section 4.4, the authors used a Gaussian distribution with zero mean and unity variance. While the choice of the value for the mean of the distribution makes sense, the value selected for the variance of the distribution should not make much difference to the overall performance obtained. This is because after the application of HEQ, all the histograms will have the same variance anyway. A set of experiments in which the variance of the reference distribution was varied from 0.75 to 2.0 was conducted. Figure 6-5 displays the average performance obtained after three runs of each experiment. Once again all the verification trials specified for the NIST 2000 database were performed.

As expected, there is a marginal difference in the overall performance of the system as the variance of the reference distribution is varied from 0.75 to 2.0. In fact, the difference in the EER of the baseline system, when the variance is varied from 0.75 to 2.0, was found not to be statistically significant. A variance of 1.0 was selected for the reference Gaussian distribution as HEQ, using a reference distribution with zero mean and unity variance, can be considered as an extension of the

[36] In order to compare the performance of two different algorithms on the baseline system, each algorithm was applied to the system independently and all the trials specified for the NIST 2000 database were performed. For each algorithm, the decision threshold was then varied until the EER point was reached. At this operating point, the system's performance for each algorithm was represented as a list of ones and zeros. A zero indicated a correct decision for a particular verification trial whereas a one indicated an incorrect decision (i.e., a speaker was either incorrectly accepted or rejected). A 2x2 contingency table was then constructed as discussed in Section 2.4.2, and McNemar's value was computed according to Equation (2.26). Only if McNemar's value is found to be greater than 3.841459, is the difference between the EERs obtained by applying the two algorithms to the same system said to be statistically significant.

mean and variance normalization technique (see Section 3.2.2) to all the moments of a probability distribution. Furthermore, a variance of 1.0 resulted in the best overall performance (see Figure 6-5). In all the remaining experiments reported on in this study, HEQ with 250 bins and, a Gaussian reference distribution, with zero mean and unity variance, was used (unless specified otherwise). In the following section, HEQ is compared to other feature-based compensation techniques, namely, cepstral mean normalization and mean and variance normalization.

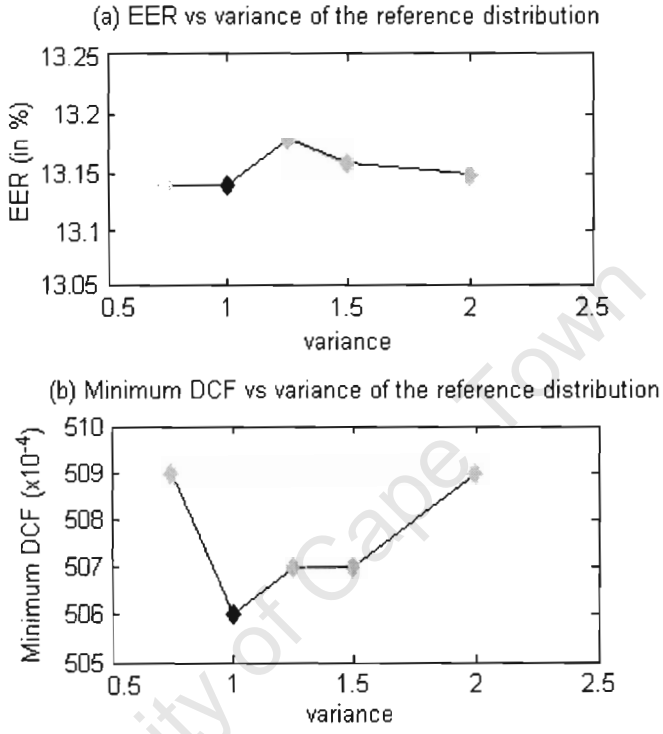

Figure 6-5: System performance versus the variance used for the reference Gaussian distribution

6.3 HEQ versus other feature-based compensation techniques

This section compares the performance of HEQ to cepstral mean normalization (see Section 3.2.1) and mean and variance normalization (see Section 3.2.2). In the speech recognition experiments performed in [91] and [105], HEQ was shown to outperform CMN and MVN and, MVN was shown to outperform CMN. The purpose of this section is to determine whether these observations apply to speaker verification as well. CMN, MVN and HEQ were each applied separately to the baseline system. These feature-based compensation techniques were applied utterance-wise with the distribution of each MFCC feature vector component being processed separately. The combined performance for all the 66572 verification trials specified for the NIST 2000 database is depicted in Figure 6-6. As illustrated by the DET curves in Figure 6-6, HEQ outperforms the other two feature-based compensation techniques across all operating points.

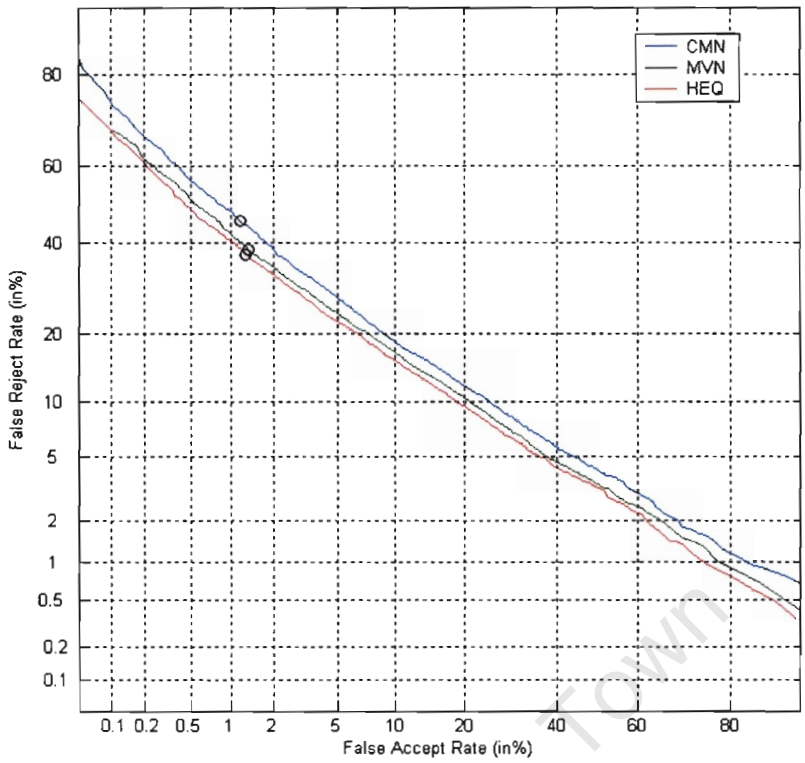

Figure 6-6: DET curves for the baseline system with different feature-based compensation techniques

The experiment that resulted in the performance depicted in Figure 6-6 was repeated three times and the average performance of the baseline system is given in Table 6-1.

Table 6-1: Combined performance for all trials with different feature-based compensation techniques (average ± standard deviation)

Feature-based compensation technique	EER	Minimum DCF ($x10^{-4}$)
CMN	14.87 ± 0.11%	569 ± 5.03
MVN	13.56 ± 0.06%	520 ± 1.15
HEQ	13.14 ± 0.04%	506 ± 4.93

From the results tabulated in Table 6-1, it is clear that HEQ reduces the EER by 11.63% relative to the EER obtained when CMN was used and by 3.10% when MVN was used. The reduction in the minimum DCF value was 11.07% relative to that obtained for CMN and 2.69% relative to that obtained for MVN. Thus, the progressive compensation of higher order moments of the feature distributions results in better speaker recognition performance. Recall that the baseline system used CMN to normalize feature distributions. Thus, these reductions in the EER and minimum DCF value can also be interpreted as improvements above the baseline system performance reported on in Chapter 5. As mentioned previously, the improvement in performance is primarily due to HEQ's ability to compensate for non-linear as well as linear distortions of the feature

space. In so doing, it normalizes the shape, scale, spread and location of feature distributions. This consolidates the knowledge that HEQ outperforms linear techniques such as CMN and MVN. Furthermore, MVN was shown to outperform CMN. Thus, the trend in the performance observed for speech recognition applications, applies to speaker verification as well. All the improvements are statistically significant as there was a greater then 95% chance that HEQ was better than MVN and that MVN was better than CMN.

Recall from Table 5-5, that with no compensation, the baseline system obtained an EER of 26.84% and a minimum DCF value of 834×10^{-4}. When these results are compared to those shown in Table 6-1 it is clear that:

1. Feature-based compensation is a crucial step in obtaining good performance in adverse environments. This is primarily due to the vulnerability of MFCCs when exposed to additive noise and linear filtering effects (see Section 3.1.3).
2. For the NIST 2000 database, the largest improvement in performance, when using feature-based compensation, is due normalization of the mean of the feature distributions. Normalization of other moments of the feature distributions leads to marginal, albeit statistically significant, improvements in performance. This result makes sense, as for the NIST 2000 database, the speech data is primarily degraded by linear filtering effects due to transmission by telephone.

Up to this point HEQ was applied on an utterance-by-utterance basis (i.e., non-segmental HEQ was applied to all the MFCCs extracted from a particular utterance). In the next section, the application of HEQ to MFCCs extracted from short speech segments (or intervals) within each utterance is examined.

6.4 Segmental versus non-segmental HEQ

In the previous section Histogram Equalization was found to outperform other feature-based compensation techniques. In this section, segmental HEQ (see Section 4.4: page 65) is compared to non-segmental HEQ. Both algorithms process the distribution of each MFCC feature vector component separately. The motivation for using segmental HEQ is that since it is applied to a buffer of features extracted from short overlapping speech segments within an utterance, it has the ability to adapt to changing environmental and recording conditions. Thus, it could potentially provide a more accurate compensation for non-stationary noise processes encountered within long utterances. A typical example is that of an individual speaking in a car. Here, the speech signal is subject to distortions caused by engine noise (which changes depending on the speed at which the car is travelling) and traffic noise (which changes depending on the time of the day or the part of

town in which the individual is travelling). Segmental HEQ is essentially the same as the feature warping technique discussed in Section 3.2.4. This technique applies a form of cumulative distribution mapping to a sliding window of features by using the relationship between the order statistics and the CDF of a dataset.

This section also provides an analysis of the effects of applying the original HEQ algorithm to a buffer of features extracted from short adjacent speech segments within an utterance. This version of HEQ is hereafter referred to as *modified segmental Histogram Equalization*. The motivation for the proposed approach is that it could make the original non-segmental version of HEQ more suitable for real-time applications as normalization is now performed over shorter time intervals, and could thus be done while an individual is speaking. Furthermore, it could be used to make HEQ more robust to changing environmental and recording conditions. It would also be much simpler to implement than segmental HEQ and should also be more computationally efficient. Figure 6-7 shows how segmental HEQ, non-segmental HEQ and modified segmental HEQ are applied to the feature vectors extracted from a particular utterance (assuming an utterance length of 9 features and a segment length of 3 features for segmental and modified segmental HEQ). As can be seen from this figure, non-segmental HEQ is applied utterance-wise, segmental HEQ is applied over a sliding window of features and modified segmental HEQ is applied over adjacent segments of the utterance.

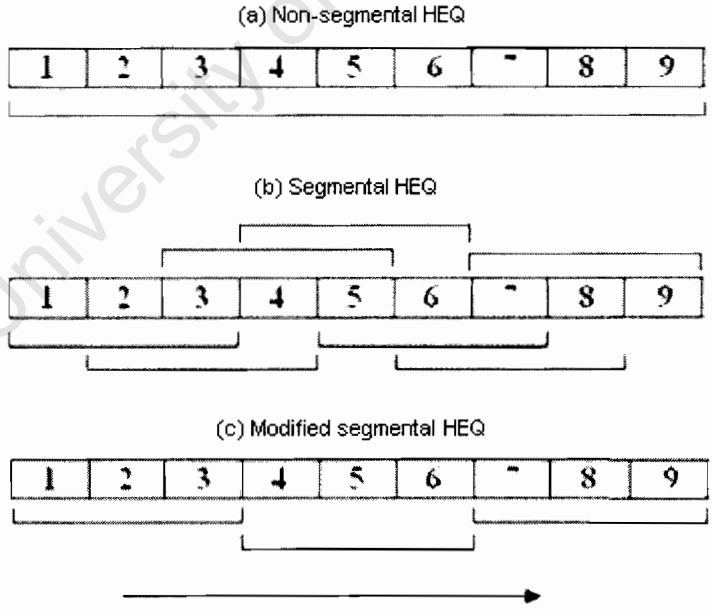

Figure 6-7: The application of non-segmental, segmental and modified segmental HEQ to the features extracted from an utterance.

Figure 6-7 shows that while non-segmental HEQ and modified segmental HEQ are applied differently, both techniques process the same number of features. On the other hand, segmental HEQ

processes more features due to its sliding window approach. Thus it could be expected to be more computationally expensive than non-segmental and modified segmental HEQ.

Table 6-2 shows the performance obtained when non-segmental HEQ and segmental HEQ replaced the CMN component in the baseline system[37]. The results were averaged over three runs. Non-segmental HEQ used the parameters discussed in Section 6.2, whereas segmental HEQ was implemented according to the feature warping approach discussed in Section 3.2.4. A three second sliding window was used according to the work done by Pelecanos and Sridharan in [80].

Table 6-2: Combined performance for all trials with non-segmental and segmental HEQ (average ± standard deviation)

HEQ version	EER	Minimum DCF ($x10^{-4}$)
Non-segmental HEQ	13.14 ± 0.04%	506 ± 4.93
Segmental HEQ	13.04 ± 0.03%	516 ± 3.46

Table 6-2 shows that there is a marginal improvement in the EER obtained for segmental HEQ over that obtained for non-segmental HEQ. However, the minimum DCF value obtained for segmental HEQ is poorer than that obtained for non-segmental HEQ. McNemar's test showed that the difference between the baseline system performance with segmental and non-segmental HEQ is not statistically significant. A possible reason for there not being a larger difference in the performance between segmental and non-segmental HEQ could be due to the way in which the CDFs were estimated. According to Blanco-Archilla et al. [117], when using order statistics "*around 500 ordered samples are enough to estimate very robustly and easily any CDF*". However, when using a 3 second sliding window and a frame rate of 80 Hz, only 3x80 = 240 feature vectors are present in each window. This lack of data could have led to inaccurate point estimates of the CDFs used, which produced relatively poor results for segmental HEQ.

After establishing the performance of segmental and non-segmental HEQ, modified segmental HEQ was applied to the baseline system. Figure 6-8 shows the combined performance for all 66572 verification trials specified for the NIST 2000 database as the segment length over which modified segmental HEQ is applied, is varied from 1 to 120 seconds. The performance previously obtained for segmental and non-segmental HEQ is also shown. The experiment was repeated three times and only the average results are shown. Recall from Section 5.1.2, that in the NIST 2000 database, the training data for each targeted speaker consists of two minutes of speech collected from a single conversation side whereas the test segment durations varies from a few sec-

[37] For simplicity, both non-segmental and segmental HEQ were applied after extracting all the MFCCs from the utterance under consideration.

onds to about a minute (with the majority ranging between 15 and 45 seconds). For utterances less than a particular segment length, all the features extracted from the utterance were used for normalization.

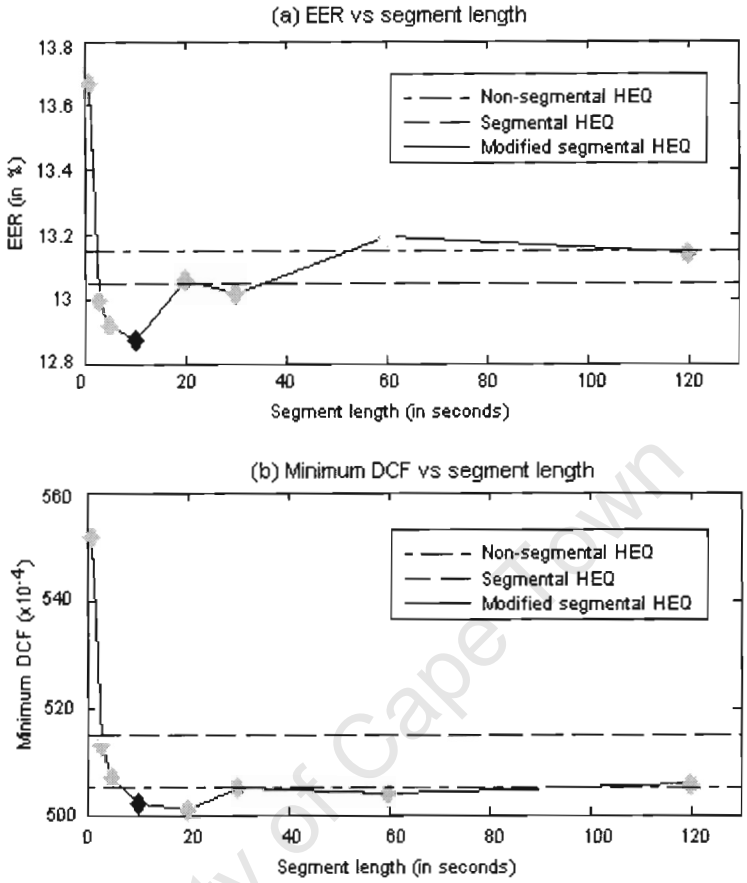

Figure 6-8: System performance versus the segment length used for modified segmental HEQ

As illustrated by Figure 6-8, there is a significant improvement in the system performance with the application of modified segmental HEQ as the segment length is increased for 1 to 10 seconds. When using a segment length of 1 second, the EER is 13.67% and the minimum DCF value is 552 $x10^{-4}$. These results are poorer than those obtained by applying non-segmental HEQ (see Table 6-2). However, when using a segment length of 10 seconds, the EER is reduced to 12.87% and the minimum DCF value is reduced to 502 $x10^{-4}$. Furthermore, this improvement in performance is not only better than that obtained when segmental HEQ and non-segmental HEQ was used (see Table 6-2), but was found to be statistically significant at a level of greater than 95%. Moreover, the performance of modified segmental HEQ translates to a relative reduction in the EER and minimum DCF value of the baseline system (see Table 5-6) of 13.45% and 11.78% respectively. This increase in system performance can be attributed to the estimation of more accurate histograms due to the large number of feature vectors present in each buffer. In contrast, as the segment length is increased beyond 10 seconds, a degradation in system performance is observed.

This could be as a result of larger segment lengths limiting modified segmental HEQ's ability to adapt to changing environmental and recording conditions.

Another observation obtained from the experiments performed using modified segmental HEQ is that when a segment length of 3 seconds was used, performance comparable to that observed when using segmental HEQ (which a employed a 3 second sliding window) was obtained (i.e., an EER of 13.00% and a minimum DCF value of 513 $x10^{-4}$ was obtained). However, timing tests showed that on average it took about two seconds to process a two minute speech utterance using modified segmental HEQ (regardless of the segment length used) and, about 20 seconds using segmental HEQ. These observations are intuitive as segmental HEQ, due to its sliding window approach, processes more speech segments within an utterance than modified segmental HEQ, applied to the same utterance, does (see Figure 6-7). Furthermore, the time taken to perform modified segmental HEQ is independent of the segment length over which it is applied as, the number of segments that need to be processed decreases as the segment length increases. On the other hand, the shorter the segment length, the shorter the time taken to perform modified segmental HEQ. Since modified segmental HEQ and segmental HEQ can be performed while an individual is speaking, the difference in the time taken to perform these techniques will not be noticeable as long as neither technique exceeds the time taken for an individual to complete speaking. However, in applications where limited computational resources are available, modified segmental HEQ may be more appropriate.

In the following section experiments are conducted to determine whether the use of a reference distribution, other than a unimodal Gaussian distribution with zero mean and unity variance, could improve the performance of the baseline system.

6.5 The use of multimodal reference distributions

In [80], Pelecanos and Sridharan speculated that "*since speech is multi-modal in nature, the ideal target distribution would also be multi-modal and representative of the speaker's true feature distribution*". However, these authors did not experimentally verify this conjecture. For the experiments reported on in this section, the overall distribution of the MFCC feature vectors used to train the UBMs discussed in Section 5.2.2 was used to obtain the reference multimodal distribution for each MFCC feature vector component. No feature-based compensation was applied to these MFCC feature vectors. Due to the characteristics of the MFCC distributions depicted in Figure 3-4, these distributions can be expected to be multimodal in nature. Separate reference distributions were constructed for male and female speakers as no cross-gender verification trials are specified for the NIST 2000 database. Data obtained from telephone handsets employing carbon-

button microphones and those employing electret microphones were used in constructing the reference distributions, as the use of handset-dependent reference distributions would increase the mismatch between cross-handset verification trials (e.g., elec/carb and carb/elec).

Multimodal reference distributions can of course be obtained in several other ways. For example, the entire distribution of the training MFCC feature vectors for all the speakers in the NIST 2000 database could be used or the distribution of MFCCs extracted from a number of clean speech utterances could be used. However, the aim of this section is not to find the multimodal reference distribution that results in the highest performance, but rather to gain some insight into the performance obtained when multimodal reference distributions are used. Table 6-3 shows the performance that was obtained when non-segmental HEQ with a multimodal reference distribution for each feature vector component replaced the CMN component in the baseline system. For comparative purposes, the performance obtained for non-segmental HEQ with a unimodal Gaussian reference distribution with zero mean and unity variance for each feature vector component is also shown. Each experiment was run three times and only the average results are tabulated.

Table 6-3: Combined performance for all trials with non-segmental HEQ using different reference distributions (average ± standard deviation)

Non-segmental HEQ	EER	Minimum DCF ($x10^{-4}$)
With a **unimodal** reference distribution for each feature vector component	13.14 ± 0.04%	506 ± 4.93
With a **multimodal** reference distribution for each feature vector component	13.97 ± 0.03%	512 ± 2.08

Table 6-3 shows that the use of non-segmental HEQ with multimodal reference distributions, obtained from UBM training data, degraded the system performance when compared to non-segmental HEQ with unimodal reference distributions. However, one important difference between non-segmental HEQ with unimodal and multimodal reference distributions (besides the number of modes) is that when using the same reference distribution for each feature vector component, the value of each normalized component falls within the same range. Thus, no particular feature vector component dominates the final location of each of the feature vectors in the feature space. Instead, dominant feature vector components are made less prominent which could have led to improved system performance.

In order to take this aspect of HEQ with unimodal reference distributions into account, each multimodal feature vector component distribution was normalized with mean and variance normalization (see Section 3.2.2) so as to have zero mean and unity variance. MVN provides a linear transformation of the feature space and, as such, does not modify the intrinsic shape of a particular dis-

tribution of features. Thus, all reference distributions remain multimodal in nature after the application of MVN. Table 6-4 shows the results obtained when the mean and variance normalized multimodal reference distributions were used to perform both non-segmental HEQ and modified segmental HEQ (over 10 second segment lengths). The results for the use of unimodal reference distributions are also given. All results were averaged over three runs.

Table 6-4: Combined performance for all trials with HEQ using normalized multimodal reference distributions (average ± standard deviation)

HEQ version	EER	Minimum DCF ($x10^{-4}$)
Non-segmental HEQ (unimodal)	13.14 ± 0.04%	506 ± 4.93
Modified segmental HEQ (unimodal)	12.87 ± 0.05%	502 ± 1.53
Non-segmental HEQ (multimodal)	13.17 ± 0.03%	508 ± 0.58
Modified segmental HEQ (multimodal)	12.85 ± 0.10 %	505 ± 2.52

Table 6-4 shows that there are minor differences between the results obtained for the corresponding versions of HEQ with unimodal and multimodal reference distributions. In fact, the differences were found not to be statistically significant. This suggests that the notion that multimodal reference distributions may be more appropriate when HEQ is applied in speaker verification applications is not necessarily the case (especially when the reference distributions are obtained using the UBM training data). In the following section, experiments are conducted to determine whether HEQ indeed has the ability to compensate for distortions regardless of the speech parameterisation used.

6.6 Application of HEQ to a combined feature set

In a number of the papers reviewed in Section 4.4, HEQ was successfully used to (1) compensate for the effects of additive noise on feature distributions; (2) compensate for the effects of residual noise caused by speech enhancement techniques; (3) normalize the features obtained at different stages during the extraction of MFCCs and, to normalize MFCC derivatives. These applications of HEQ suggest that the technique can be used to reduce the effects of a wide range of noise processes affecting various speech parameterisations. This is because HEQ is essentially a technique that maps one histogram to another, regardless of any model of speech production, transmission or perception. In the case of MFCC feature vector distributions however, the application of HEQ is intended to compensate for the spectral variations caused by additive noise and linear filtering

effects on a speech signal. In this section, experiments are conducted in order to determine whether HEQ can be used to compensate for the way in which telephone transmission affects a combined feature set. In particular, MFCC feature vectors are combined with a feature set known as the Maximum Autocorrelation Values (MACVs).

In 2004, the author co-authored a paper in which MFCCs were combined with MACVs and applied to a speaker identification task [118]. The combination of the two feature sets was shown to improve performance above that obtained when only MFCCs were used. The MACV feature set, proposed by Wildermoth and Paliwal in [119], is aimed at extracting pitch and voicing information from a segmented frame of speech. Recall from Section 2.1.2 that MFCCs are aimed at representing information related an individual's unique vocal tract structure. Thus, MFCCs and MACVs contain different information as the one feature set is aimed at extracting physiological information while the other is aimed at extracting psychological information (i.e., information related to an individual's learned manner of speaking).

Given a speech frame $\{s(n),\ n = 0, 1,\ldots, N_s - 1\}$, the MACV feature set is computed as follows [119]:

1. Compute the autocorrelation function, $R(k)$:

$$R(k) = \frac{1}{N_s}\sum_{n=0}^{N_s-1-k} s(n)s(n+k) \qquad k = 0,\ldots,N_s-1 \tag{6.1}$$

2. Normalize $R(k)$ by its maximum value:

$$\hat{R}(k) = \frac{R(k)}{R(0)} \tag{6.2}$$

3. Split the higher portion of $\hat{R}(k)$, from about 2ms to 16ms, into M equal divisions.
4. Find the maximum value of $\hat{R}(k)$ in each of the M divisions.
5. The M maximum autocorrelation values now form an M-dimensional feature vector.

Typically, M is chosen to be equal to 5. It should be noted that the lower portion of the normalized autocorrelation function is not used because it contains information pertaining to the vocal tract structure. This information is already captured by the MFCC feature vectors to which the MACV feature vectors are concatenated. The higher portion of the normalized autocorrelation function is based on the fact that the pitch of a human voice is typically between 60 Hz and 400 Hz (60 to 160 Hz for males and 160 to 400 Hz for females) which translates into a range from 2 milliseconds to 16 milliseconds [119].

For the experiments conducted in this section, 5-dimensional MACV feature vectors were appended to the 18-dimensional MFCC feature vectors extracted from each frame of speech in the

NIST 2000 database. Table 6-5 shows the results obtained when MACVs were appended to the MFCC feature vectors. Each experiment was run three times and the average of the results obtained was taken. No feature-based compensation was used.

Table 6-5: Combined performance for all trials when MFCCs are combined with MACVs (average ± standard deviation)

Feature type	EER	Minimum DCF ($x10^{-4}$)
18 MFCCs	26.84 ± 0.01%	834 ± 3.61
18 MFCCs combined with 5 MACVs	25.85 ± 0.04%	790 ± 1.53

The results in Table 6-5 consolidate the observations in [118] that the combination of MFCCs and MACVs leads to an improvement in speaker recognition performance above that obtained for MFCCs alone. There is a 3.69% relative reduction in the EER and a 5.28% relative reduction in the minimum DCF value.

When the same MACVs were combined with MFCC feature vectors normalized by modified segmental HEQ applied over 10 second segment lengths (with unimodal reference distributions), an improved EER of 14.42% and minimum DCF value of $537x10^{-4}$, was obtained. These results are poorer than those obtained without the addition of MACVs (see Table 6-4). Two possible reasons for this degradation in performance could be that (1) telephone transmission causes a mismatch between the MACV features extracted from a particular speaker's training and test data; or (2) the contribution from the MACV feature set dominates the location of the combined feature vectors in the feature space and, as such, reduces system performance.

In the remainder of this section, modified segmental HEQ applied over 10 second segment lengths (with unimodal reference distributions) was used to compensate for these two distortions. Figure 6-9 shows the results of this experiment. As illustrated, the application of HEQ to both feature sets leads to a substantial improvement in the performance obtained for the combined feature set. In fact, the average EER of 12.47% and minimum DCF value of $483x10^{-4}$ is lower than that obtained when only MFCC feature vector distributions are normalized. It must however be noted that normalization of the MFCC feature vectors led to the largest improvement in the performance of the combined feature set. Subsequent normalization of the MACV feature vectors only led to marginal, albeit statistically significant, improvements in performance. Still, this result reinforces the versatility of the HEQ technique and its ability to compensate for a wide range of distortions. Furthermore, the application of HEQ to both feature sets translates to a relative reduction in the EER and the minimum DCF value of the baseline system (see Table 5-6) of 16.14% and 15.11% respectively.

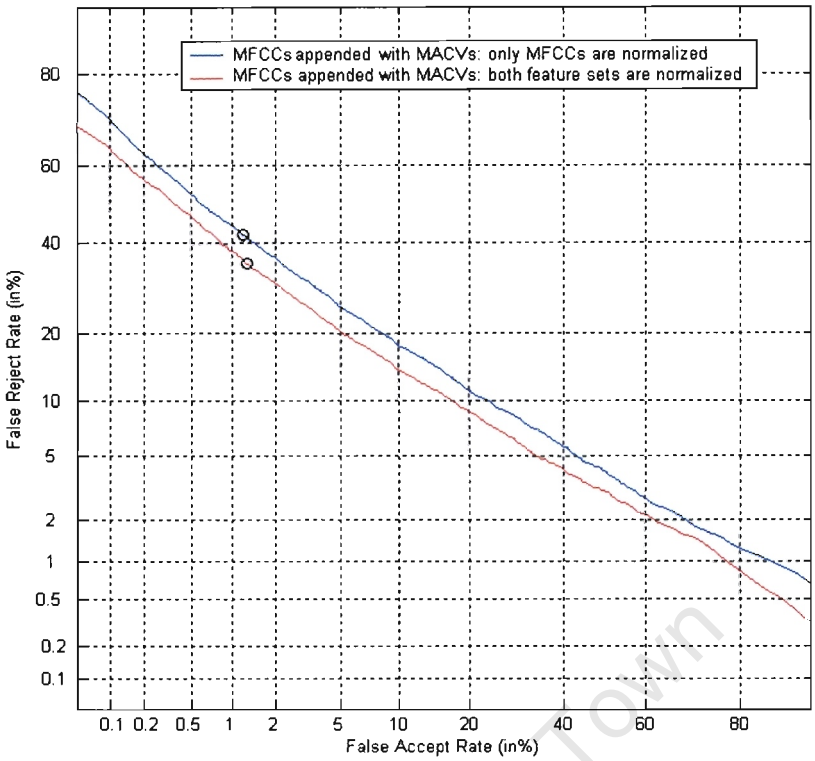

Figure 6-9: System performance when different components of the combined feature set are normalized

6.7 Summary

This chapter experimentally evaluated the Histogram Equalization technique proposed in this study. The experiments conducted here verified that the technique indeed allows one to map one histogram to another, and also determined the optimal parameters to use for the HEQ algorithm. In addition, non-segmental HEQ was shown to outperform other feature-based compensation techniques, and was shown to reduce the equal error rate by 11.63% and the minimum detection cost function by 11.07% relative to the baseline system described in Chapter 5. Furthermore, a proposed variation of HEQ, termed modified segmental HEQ, was shown to outperform segmental HEQ using a feature warping approach and non-segmental HEQ. Also, a relative reduction in the equal error rate and minimum detection cost function value of the baseline system of 13.45% and 11.78% respectively, was obtained when using this form of HEQ.

It was also shown that normalizing MFCC feature vector component distributions with HEQ using either unimodal or normalized multimodal reference distributions did not lead to any significant variations in performance. Finally in Section 6.6, modified segmental HEQ (over 10 second adjacent speech segments) was applied to a combined feature set namely, mel-frequency cepstral coefficients concatenated with a pitch-based feature set known as the Maximum Autocorrelation Values. This application of HEQ produced results superior to that of normalizing MFCCs alone

and, showed that HEQ indeed has the ability to compensate for a wide range of distortions, even those affecting a combined feature set. A relative reduction in the equal error rate and minimum detection cost function value of the baseline system of 16.14% and 15.11% respectively, was obtained.

Chapter 7

Conclusions

This chapter presents a summary of the achievements of this study and, conclusions based on the research and experimental work done, are also drawn. Finally, directions for future work are presented.

7.1 Summary of work done

Recall from Chapter 1 that this thesis had three main objectives. The first objective was to provide a comprehensive review of contemporary speaker verification literature with particular emphasis on techniques that are generally regarded as standard practice in the field and techniques that have been used to improve the robustness of speaker verification systems. This objective was fulfilled by Chapters 2 and 3 of this document. Chapter 2 focussed specifically on techniques used to construct contemporary speaker verification systems while Chapter 3 described several strategies for achieving robustness in speaker verification. In addition, Chapter 1 provided a very general overview of the area of speaker recognition so as to allow the user to become familiar with the various terms and concepts used throughout this document. A discussion of the various factors affecting the performance of speaker verification systems was also presented in this chapter. Mismatched training and test conditions were highlighted as the specific problem that this thesis addresses.

Using the methodologies described in Chapters 2 and 3, the second objective of this thesis was to design and implement a baseline text-independent speaker verification system. This objective was fulfilled by Chapter 5 which, described in detail, the construction of such a system. The system was shown to perform similarly to other systems evaluated under similar conditions and, provided an experimental framework for evaluating the technique proposed in this study. Furthermore, its performance was used as a benchmark against which all subsequent improvements were com-

pared. Additionally, the development of the baseline system consolidated many of the findings reported in contemporary literature (see Section 5.4).

This thesis proposed a feature-based compensation technique known as Histogram Equalization. This technique has the ability to minimise the mismatch between two distributions. The main motivation for the use of this technique, when applied in speaker verification, was that it could be used to reduce the mismatch between feature distributions obtained in different training and test conditions and hence, lead to improved performance. The final objective of this thesis was to implement and evaluate Histogram Equalization. As such, the mathematical formulation and the background of the technique, as well as a simple algorithm for implementing the technique were presented in Chapter 4. A review of the previous applications of the technique showed that it had had limited application in the area of speaker recognition and on speech degraded by telephone transmission.

An evaluation of the technique, as well as an analysis of the various results obtained, was provided in Chapter 6. The results showed that Histogram Equalization could be used to improve the robustness of speaker verification systems operating in telephone environments. It was shown to outperform several feature-based compensation techniques and led to significant improvements in performance above the baseline system developed in Chapter 5. Furthermore, a proposed variation of the technique, in which the algorithm was applied over 10 second adjacent segments within an utterance, was shown to outperform the original non-segmental version of HEQ used in related literature. The majority of the software components involved in building the baseline text-independent speaker verification system and evaluating HEQ were implemented by the author. The following section highlights key conclusions based on the work done in this study.

7.2 Conclusions

Based on the research and experiments conducted in this study, the following conclusions are drawn:

- Additive noise and linear filtering effects corrupt MFCC feature distributions. These effects change the shape, scale, spread and location of MFCC feature distributions. This is one of the primary reasons that speaker verification systems perform poorly in mismatched training and test conditions as, the resulting MFCC feature distributions will be different. The application of CMN, MVN and various forms of HEQ alleviates some of this disparity in feature distributions as they are aimed at making feature distributions more consistent across different recording conditions. Thus, feature-based compensation is a crucial step in obtaining good speaker verification performance.

- Histogram Equalization can be used to improve the performance of speaker verification systems operating in telephone environments (i.e., speaker verification systems evaluated on speech degraded by telephone transmission). This is due to its ability to minimise the mismatch between feature distributions obtained in mismatched training and test conditions. Furthermore, non-segmental HEQ outperforms CMN and MVN due to its ability to compensate for both linear and non-linear distortions of the feature space, which normalizes the shape, scale, spread and location of MFCC feature distributions.

- For the NIST 2000 database, the largest improvement in performance, when using feature-based compensation, is due to normalization of the mean of the feature distributions. Normalization of other moments of the feature distributions leads to marginal, albeit statistically significant, improvements in performance. This result makes sense, as for the NIST 2000 database, the speech data is primarily degraded by linear filtering effects due to transmission by telephone.

- Performing HEQ over features extracted from 10 second adjacent segments of speech (i.e., modified segmental HEQ) outperforms non-segmental HEQ. This is due to modified segmental HEQ's ability to adapt to changing environmental conditions. This form of HEQ is also more suitable for on-line applications as it can be performed while an individual is speaking. Furthermore, this form of HEQ outperforms HEQ applied over a 3 second sliding window (i.e., segmental HEQ) due to the estimation of more accurate cumulative distribution histograms.

- For unimodal Gaussian reference distributions, the number of bins used when estimating the histograms required for HEQ is directly proportional to the time taken to perform HEQ. However, no significant variation in performance is observed when the number of bins varies between 100 and 2000. Also, changing the variance of the distribution from 0.75 to 2.0 leads to no noteworthy difference in the performance of HEQ. Furthermore, there is no significant disparity in the performance of HEQ with unimodal and normalized multimodal reference distributions (obtained from UBM training data). However, when using multimodal reference distributions that are not normalized, performance degrades due to unequal contributions from each feature vector component.

- HEQ has the ability to compensate for a wide range of distortions. This is due to the fact that HEQ does not depend on any model of speech production, perception or transmission. In this work, HEQ applied to MFCC feature vectors concatenated with MACV feature vectors improved performance above that of applying HEQ to MFCC feature vectors alone. This application of HEQ also shows that normalizing the contribution from each feature set is a crucial step when combined feature sets are employed and, confirms the

knowledge that MFCCs combined with MACVs improves speaker recognition performance.

7.3 Directions for future research

As a result of the scope, limitations, findings and conclusions of this study, the following recommendations for future work involving HEQ are made:

- In this study, the compensating effect of HEQ has only been evaluated on MFCCs (and to a smaller extent on MACVs). As such, further research should be conducted into the application of HEQ on other feature sets and, at different stages during feature extraction. This would provide further insight into the ability of HEQ to compensate for various distortions regardless of the speech parameterisation used.
- HEQ has been shown to improve the performance of a speaker verification system operating in telephone environments. It is recommended that an investigation be conducted into the application HEQ on speech contaminated by various types of additive noise (at different signal-to-noise ratios). Furthermore, an investigation into the ability of HEQ to improve the robustness of a system trained with clean speech and tested with contaminated speech would also be very interesting. This would provide some indication of possible applications and situations in which the technique could be useful.
- The performance obtained when HEQ was applied using either a unimodal or multimodal reference distribution for each feature vector component was shown to be very similar. The multimodal reference distributions were obtained from feature vectors extracted from UBM training data. Whether obtaining the reference distributions from feature vectors extracted from clean speech leads to improved performance still needs to be determined.
- HEQ is based on a simple and repetitive algorithm and, as such, lends itself to implementation on a digital signal processor. A study of the feasibility of such an approach, as well the trade-off between performance and memory and computational constraints should also be conducted. This knowledge could promote the use of HEQ in mobile devices.
- In this study, a variation of HEQ, in which the algorithm is applied over adjacent segments of speech, instead of over an entire utterance, was shown to improve the compensating ability of technique as this approach allowed it to adapt to changing environmental conditions. This technique should be able to provide fast feature-based compensation in on-line applications as it can be applied while an individual is speaking. However, the technique is yet to be applied in a real-world on-line application. Such a study would pro-

vide some insight into the real-world applicability of HEQ as well as how it compares with other techniques such as feature warping for example.

- HEQ is a feature-based compensation technique that has been applied to a speaker verification system employing a statistical pattern matching technique, namely, Gaussian mixture models. It would be interesting to see whether HEQ improves the performance of speaker verification systems employing discriminative classification techniques such as SVMs for example.

- In this study, HEQ has been applied to each feature vector component separately. However, it still to be ascertained whether certain feature vector components rely more on feature-based compensation than others and, whether the use of different reference histograms for each component leads to improvements in performance.

- To date, there is no standard technique for determining the statistical significance of improvements between different algorithms. McNemar's test was used in this study due to it being used in other speaker recognition studies. However, there is a need to develop a standard strategy for determining whether newly proposed algorithms indeed lead to statistically significant improvements in performance.

- In this study, HEQ was only applied to the MFCC feature vectors extracted from speech in the NIST 2000 database. A survey of the application of HEQ on other databases still needs to be conducted. This would indicate whether the improvements reported for the NIST 2000 database generalise to other databases and, to what extent the values of the parameters selected for HEQ need to be modified when HEQ is applied on other databases.

The work done in this study showed that feature-based compensation, in the form of Histogram Equalization, can be used to significantly improve the robustness of speaker verification systems operating in mismatched training and test conditions. While the objectives of this thesis have been accomplished, it is clear that there is still much work to be done in improving the performance of speaker verification systems in adverse environments as an equal error rate of 0% is yet to be achieved. However, speaker verification is an emerging technology and, as such, considerable research is being conducted into making speaker verification architectures more robust to diverse acoustic environments, mobile and fixed-line telephony networks and excessive speaker variability. As a result, the future of speaker verification looks extremely promising.

Bibliography

[1] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology - Special Issue on Image- and Video-based Biometrics*, vol. 14, pp. 4-20, 2004.

[2] J. P. Campbell, "Speaker recognition: A tutorial," in *Proceedings of the IEEE*, vol. 85, pp. 1437-1462, 1997.

[3] D. A. Reynolds, "An overview of automatic speaker recognition technology," in *Proceedings of IEEE ICASSP 2002*, vol. 4, pp. 4072-4075, 2002.

[4] G. R. Doddington, M. A. Pryzbocki, A. F. Martin, and D. A. Reynolds, "The NIST speaker recognition evaluation - Overview, methodology, systems, results, perspective," *Speech Communication*, vol. 31, pp. 225-254, 2000.

[5] R. L. Klevans and R. D. Rodman, *Voice Recognition*: Arctech House, INC, 1997.

[6] L. G. Kersta, "Voiceprint Identification," *Nature*, vol. 196, pp. 1253-1257, 1962.

[7] F. Bimbot, J. Bonastre, C. Fredouille, G. Gravier, I. Margin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, and D. A. Reynolds, "A tutorial on text-independent speaker verification," *EURASIP Journal on Applied Signal Processing*, vol. 4, pp. 430-451, 2004.

[8] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical Pattern Recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 4-37, 2000.

[9] J. M. Naik, "Speaker verification: A tutorial," *IEEE Communications Magazine*, pp. 42-48, 1990.

[10] J. Wolf, "Efficient acoustic parameters for speaker recognition," *Journal of the Acoustical Society of America*, vol. 51, pp. 2044-2056, 1972.

[11] D. A. Reynolds, "Automatic speaker recognition using Gaussian mixture speaker models," *The Lincoln Laboratory Journal*, vol. 8, pp. 173-192, 1995.

[12] J. A. Markowitz, "Voice Biometrics," *Communications of the ACM*, vol. 43, pp. 66-73, 2000.

[13] J. A. Markowitz, "Speaker verification," *Biometric Technology Today*, vol. 9, pp. 9-11, 2001.

[14] J. A. Markowitz, "Speaker recognition," *Information security technical report*, vol. 3, pp. 14-20, 1998.

[15] L. Lu and H. Zhang, "Speaker change detection and tracking in real-time news broadcasting analysis," in *Proceedings of the 10th ACM International Conference on Multimedia*, pp. 602-610, 2002.

[16] R. B. Dunn, D. A. Reynolds, and T. F. Quatieri, "Approaches to Speaker Detection and Tracking in Conversational Speech," *Digital Signal Processing*, vol. 10, pp. 93-112, 2000.

[17] D. A. Reynolds, "Large population speaker identification using clean and telephone speech," *IEEE Signal Processing Letters*, vol. 2, 1995.

[18] H. Gish and M. Schmidt, "Text-independent speaker identification," *IEEE Signal Processing Magazine*, pp. 18-32, 1994.

[19] J. Naik, "Field trial of a speaker verification service for caller identity verification in the telephone network," *2nd IEEE Workshop on Interactive Voice Technology for Telecommunications Applications (IVTTA94)*, pp. 125-128, 1994.

[20] D. James, H. Hutter, and F. Bimbot, "CAVE - Speaker Verification in Banking and Telecommunications," in *Proceedings of the Ubilab Conference '96*, 1996.

[21] "T-Netix Inmate calling system," Available At: http://www.t-netix.com/products/default.asp?m=callsys [Last Accessed: 14/10/2004].

[22] J. Koolwaaij and L. Boves, "On decision making in forensic casework," *Forensic Linguistics*, vol. 6, pp. 242-264, 1999.

[23] J. Koolwaaij, *Automatic speaker verification in telephony: A probabilistic approach*: PrintPartners Iskamp B.V., Enschede, 2000.

[24] P. J. Moreno and R. M. Stern, "Sources of degradation of speech recognition in the telephone network," in *Proceedings of IEEE ICASSP 1994*, vol. 1, pp. 109-112, 1994.

[25] D. A. Reynolds, M. A. Zissman, T. F. Quatieri, and G. C. O'Leary, "The effects of telephone degradations on speaker identification peformance," in *Proceedings of IEEE ICASSP 1995*, pp. 329-332, 1995.

[26] D. A. Reynolds, "A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification.," *PhD Thesis*: Georgia Institute of Technology, 1992.

[27] S. Furui, "Recent advances in speaker recognition," *Pattern Recognition Letters*, vol. 18, pp. 859-872, 1997.

[28] H. A. Murthy, F. Beaufays, L. P. Heck, and M. Weintraub, "Robust text-independent speaker identification over telephone channels," *IEEE Transactions on Speech and Audio Processing*, vol. 7, pp. 554-568, 1999.

[29] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 72-83, 1995.

[30] M. Skosan and D. J. Mashao, "Improving speaker identification performance for telephone-based applications," in *Proceedings of SATNAC 2004*, 2004.

[31] M. Skosan and D. J. Mashao, "Matching feature distributions for robust speaker verification," in *Proceedings of PRASA 2004*, pp. 93-97, 2004.

[32] J. Bonastre, F. Bimbot, L. Boe, J. P. Campbell, D. A. Reynolds, and I. Magrin-Chagnolleau, "Person authentication by voice: A need for caution," in *Proceedings of Eurospeech 2003*, pp. 33-36, 2003.

[33] H. Hermansky, "Perceptual linear predictive analysis of speech," *Journal of the Acoustical Society of America*, vol. 87, pp. 1738-1752, 1990.

[34] D. A. Reynolds, "Experimental Evaluation of Features for Robust Speaker Identification," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 639-643, 1994.

[35] D. J. Mashao, "Auditory-based speaker identification system," in *Proceedings of PRASA 2001*, pp. 147-153, 2001.

[36] L. Rabiner and B. Juang, *Fundamentals of speech recognition*: Prentice Hall, 1993.

[37] O. Ghitza, "Auditory nerve representation as a front-end for speech recognition in a noisy environment," *Computer Speech and Language*, vol. 1, pp. 109-130, 1986.

[38] N. Baloyi and D. J. Mashao, "Improvements in the speaker identification rate using feature sets on a large population database," in *Proceedings of Eurospeech 2001*, vol. 4, pp. 2833-2836, 2001.

[39] D. J. Mashao, "Computations and evaluations of an optimal feature set for an HMM-based recognizer," *PhD Thesis*: Brown University, 1996.

[40] N. P. H. Thian, C. Sanderson, and S. Bengio, "Spectral subband centroids as complementary features for speaker authentication," *International Conference on Biometric Authentication*, pp. 631-639, 2004.

[41] B. Gajic and K. K. Paliwal, "Robust feature extraction using subband spectral centroid histograms," in *Proceedings of IEEE ICASSP 2001*, vol. 1, pp. 85-88, 2001.

[42] B. R. Wildermoth, "Text-independent speaker recognition using source based features," *Master's Thesis*: Griffith University, Australia, 2001.

[43] I. Magrin-Chagnolleau, G. Gravier, M. Seck, O. Boeffard, R. Blouet, and F. Bimbot, "A further investigation on speech features for speaker characterization," in *Proceedings of ICSLP 2000*, vol. 3, pp. 1029-1032, 2000.

[44] T. Kinnunen, "Spectral features for automatic text-independent speaker identification," *Licentiate's Thesis*: University of Joensuu, Joensuu, Finland, 2003.

[45] P. A. Lynn and W. Fuerest, *Introductory digital signal processing with computer applications*: John Wiley and Sons Ltd., 1993.

[46] H. Melin, "Databases for speaker recognition: Activities in COST250 working group 2," *COST 250 - Speaker Recognition in Telephony, Final Report 1999, European Commission DG-XIII*, 2000.

[47] J. P. Campbell and D. A. Reynolds, "Corpora for the evaluation of speaker recognition systems," in *Proceedings of IEEE ICASSP 1999*, pp. 829-832, 1999.

[48] J. Godfrey, D. Graff, and A. Martin, "Public databases for speaker recognition and verification," in *Proceedings of ESCA workshop on Automatic Speaker Recognition, Identification and Verification*, pp. 39-42, 1994.

[49] F. de Wet, "Automatic speech recognition in adverse acoustic conditions," *PhD Thesis*: University of Nijmegen, The Netherlands, 2003.

[50] R. P. Ramachandran, K. R. Farrell, R. Ramachandran, and R. J. Mammone, "Speaker recognition - general classifier approaches and data fusion methods," *Pattern Recognition*, vol. 35, pp. 2801-2821, 2002.

[51] S. Roweis, "Speech processing background," 1998, Available At: http://www.dna.caltech.edu/courses/cns187/references/Roweis_spblet.ps [Last Accessed: 17/01/2005].

[52] R. J. Mammone, X. Zhang, and R. P. Ramachandran, "Robust speaker recognition - A feature based approach," *IEEE Signal Processing Magazine*, vol. 13, pp. 58-71, 1996.

[53] C. Sanderson, "Automatic Person Verification Using Speech and Face Information," *PhD Thesis*: Griffith University, 2002.

[54] D. O'Shaughnessy, *Speech Communication - Human and Machine*: Addison-Wesley, New York, 1987.

[55] C. Barras and J. Gauvain, "Feature and score normalization for speaker verification of cellular data," in *Proceedings of IEEE ICASSP 2003*, vol. 2, pp. 49-52, 2003.

[56] F. Soong and A. Rosenberg, "On the use of instantaneous and transitional spectral information in speaker recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 871-879, 1988.

[57] S. Furui, "Comparison of speaker recognition methods using static features and dynamic features," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, pp. 342-350, 1981.

[58] D. Kim, J. Jeong, S. Lee, and R. M. Kil, "A comparison of front-ends for robust speech recognition," *Journal of the Acoustical Society of Korea*, vol. 17, pp. 3-11, 1998.

[59] D. A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech Communication*, vol. 17, pp. 91-108., 1995.

[60] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models," *Digital Signal Processing*, vol. 10, pp. 19-41, 2000.

[61] R. Jhumka, "Evaluation of different support vector machines kernels," *Master's Thesis*: University of Cape Town, South Africa, 2004.

[62] V. Wan, "Speaker verification using support vector machines," *PhD Thesis*: University of Sheffield, United Kingdom, 2003.

[63] D. A. Reynolds, "Comparison of background normalization methods for text-independent speaker verification," in *Proceedings of Eurospeech 1997*, vol. 2, pp. 963-966, 1997.

[64] J. A. Bilmes, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," *Technical Report ICSI-TR-97-021*, 1998.

[65] R. A. Finan, A. T. Sapeluk, and R. I. Damper, "Impostor cohort selection for score normalisation in speaker verification," *Pattern Recognition Letters*, vol. 18, pp. 881-888, 1997.

[66] L. P. Heck and M. Weintraub, "Handset-dependent background models for robust text-independent speaker recognition," in *Proceedings of IEEE ICASSP 1997*, vol. 2, pp. 1071-1074, 1997.

[67] G. Gravier and G. Chollet, "Comparison of normalization techniques for speaker verification," *Reconnaissance du Locuteur et ses Applications Commerciales et Criminalistiques (RLA2C)*, 1998, Available At: http://www.irisa.fr/metiss/ggravier/biblio/gravier-rla2c.ps [Last Accessed: 12/01/2005].

[68] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," in *Proceedings of EuroSpeech 1997*, pp. 1895-1898, 1997.

[69] S. Bengio and J. Mariethoz, "A statistical significance test for person authentication," in *Proceedings of ODYSSEY 2004*, pp. 237-244, 2004.

[70] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, pp. 1895-1923, 1998.

[71] T. G. Dietterich, "Statistical tests for comparing supervised classification learning algorithms," 1997, Available At: http://www.isip.msstate.edu/projects/speech/support/help/bibliography/machine_learning/tdietterich_significance_tests.ps.gz [Last Accessed: 12/01/2005].

[72] Y. Roggo, L. Duponchel, and J. Huvenne, "Comparison of supervised recognition methods with McNemar's statistical test," *Analytica Chimica Acta*, vol. 477, pp. 187-200, 2003.

[73] Y. Sun, T. S. Buttler, A. Shafarenko, R. Adams, M. Loomes, and N. Davey, "Segmenting handwritten text using supervised classification techniques," in *Proceedings of IEEE IJCNN 2004*, pp. 657-662, 2004.

[74] S. van Vuuren, "Speaker verification in a time-feature space," *PhD Thesis*: Oregon Graduate Institute of Science and Technology, 1999.

[75] L. Lerato, "Hierarchical methods for large population speaker identification using telephone speech," *Master's Thesis*: University of Cape Town, South Africa, 2003.

[76] J. Ortega-García and J. González-Rodríguez, "Overview of speech enhancement techniques for automatic speaker recognition," in *Proceedings of ICSLP 1996*, vol. 2, pp. 929-932, 1996.

[77] M. Padilla and T. F. Quatieri, "A comparison of soft and hard spectral subtraction for speaker verification," in *Proceedings of ICSLP 2004*, pp. 642-645, 2004.

[78] I. A. McCowan, J. Pelecanos, and S. Sridharan, "Robust speaker recognition using microphone arrays," in *Proceedings of ODYSSEY 2001*, pp. 101-106, 2001.

[79] J. P. Campbell, D. A. Reynolds, and R. B. Dunn, "Fusing high- and low-level features for speaker recognition," in *Proceedings of Eurospeech 2003*, pp. 2665-2668, 2003.

[80] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proceedings of ODYSSEY 2001*, pp. 213-218, 2001.

[81] P. J. Moreno, "Speech recognition in telephone environments," *Master's Thesis*: Carnegie Mellon University, Pittsburg, Pensylvania, 1992.

[82] B. 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, pp. 1304-1312, 1974.

[83] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Communication*, vol. 25, pp. 133-147, 1998.

[84] H. Hermansky and N. Morgan, "RASTA Processing of Speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 587-589, 1994.

[85] S. van Vuuren and H. Hermansky, "On the importance of components of the modulation spectrum for speaker verification," in *Proceedings of ICSLP 1998*, pp. 3205-3208, 1998.

[86] S. van Vuuren, "Comparison of text-independent speaker recognition methods on telephone speech with acoustic mismatch," in *Proceedings of ICSLP 1996*, pp. 1784-1787, 1996.

[87] B. Xiang, U. Chaudhari, J. Navratil, G. Ramaswamy, and G. Gopinath, "Short-time guassianization for robust speaker verification," in *Proceedings of IEEE ICASSP 2002*, vol. 1, pp. 681-684, 2002.

[88] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, pp. 42-54, 2000.

[89] A. de la Torre, A. M. Peinado, J. C. Segura, J. L. Perez, M. C. Benítez, and A. J. Rubio, "Histogram equalization of the speech representation for robust speech recognition," *IEEE Transactions on Speech and Audio Processing*, (Accepted for publication).

[90] S. Molau, "Normalization in the Acoustic Feature Space for Improved Speech Recognition," *PhD Thesis*: Aachen, Germany, 2003.

[91] J. C. Segura, M. C. Benítez, A. de la Torre, A. J. Rubio, and J. Ramírez, "Cepstral domain segmental nonlinear feature transformations for robust speech recognition," *IEEE Signal Processing Letters*, vol. 11, pp. 517-520, 2004.

[92] S. Dharanipragada and M. Padmanabhan, "A Nonlinear Unsupervised Adaptation Technique for Speech Recognition," in *Proceedings of ICSLP 2000*, pp. 556-559, 2000.

[93] W. Niblack, *An Introduction to Digital Image processing*: Prentice-Hall International (UK) Ltd., 1986.

[94] A. R. Weeks Jr., *Fundamentals of electronic image processing*: (co-published) SPIE optical engineering press & IEEE Press., 1996.

[95] J. Matthews, "Histogram Equalization," 2004, Available At: http://www.generation5.org/content/2004/histogramEqualization.asp [Last Accessed: 31/01/2005].

[96] R. Balchandran and R. J. Mammone, "Non-parametric estimation and correction on non-linear distortion in speech systems," in *Proceedings of IEEE ICASSP 1998*, vol. 2, pp. 749-752, 1998.

[97] S. Molau, D. Keysers, and H. Ney, "Matching Training and Test Data Distributions for Robust Speech Recognition," *Speech Communication*, vol. 41, pp. 579-601, 2003.

[98] S. Molau, M. Pitz, and H. Ney, "Histogram based normalization in the acoustic feature space," in *Prodeedings of ASRU 2001*, pp. 21-24, 2001.

[99] S. Molau, F. Hilger, D. Keysers, and H. Ney, "Enhanced histogram normalization in the acoustic feature space," in *Proceedings of IEEE ICASSP 2002*, vol. 1, pp. 1421-1424, 2002.

[100] F. Hilger and H. Ney, "Quantile Based Histogram Equalization for Noise Robust Speech Recognition," in *Proceedings of Eurospeech 2001*, vol. 2, pp. 1135-1138, 2001.

[101] F. Hilger, S. Molau, and H. Ney, "Quantile Based Histogram Equalization for Online Applications," in *Proceedings of ICSLP 2002*, vol. 1, pp. 237-240, 2002.

[102] J. C. Segura, M. C. Benítez, A. de la Torre, S. Dupont, and A. J. Rubio, "VTS residual noise compensation," in *Proceedings of IEEE ICASSP 2002*, vol. 1, pp. 409-412, 2002.

[103] J. C. Segura, M. C. Benítez, A. de la Torre, and A. J. Rubio, "Feature Extraction Combining Spectral Noise Reduction and Cepstral Histogram Equalization for Robust ASR," in *Proceedings of ICSLP 2002*, pp. 225-228, 2002.

[104] J. C. Segura, J. Ramírez, M. C. Benítez, A. de la Torre, and A. J. Rubio, "Improved feature extraction based on spectral noise reduction and non-linear feature normalization," in *Proceedings of Eurospeech 2003*, pp. 353-356, 2003.

[105] A. de la Torre, J. C. Segura, M. C. Benítez, A. M. Peinado, and A. J. Rubio, "Non-linear transformations of the feature space for robust speech recognition," in *Proceedings of IEEE ICASSP 2002*, pp. 401-404, 2002.

[106] Y. Obuchi and R. M. Stern, "Normalization of Time-Derivative Parameters using Histogram Equalization," in *Proceedings of Eurospeech 2003*, pp. 665-668, 2003.

[107] F. de Wet, J. de Veth, L. Boves, and B. Cranen, "Additive noise as a source of non-linear mismatch in the cepstral and log-energy domain," *Computer Speech and Language*, vol. 19, pp. 31-54, 2005.

[108] B. Noe, J. Sienel, D. Jouvet, L. Mauuary, L. Boves, J. de Veth, and F. de Wet, "Noise reduction for noise robust feature extraction for distributed speech recognition," in *Proceedings of Eurospeech 2001*, pp. 433-436, 2001.

[109] A. Martin and M. Przybocki, "The NIST speaker recognition evaluations: 1996-2001," in *Proceedings of ODYSSEY 2001*, pp. 39-43, 2001.

[110] M. Przybocki and A. Martin, "NIST's assessment of text-independent speaker recognition performance," *The Advent of Biometrics on the Internet, A COST 275 Workshop*, 2002.

[111] M. Przybocki and A. Martin, "Odyssey text-independent evaluation data," in *Proceedings of ODYSSEY 2001*, pp. 21-24, 2001.

[112] "NIST 2000 Speaker Recognition evaluation," Available At: http://www.nist.gov/speech/tests/spk/2000/ [Last Accessed: 17/05/2005].

[113] R. D. Zilca, "Text independent speaker verification using covariance modeling," *IEEE Signal Processing Letters*, vol. 8, pp. 97-99, 2001.

[114] R. D. Zilca, "Text-independent speaker verification using utterance level scoring and covariance modeling," *IEEE Transactions on Speech and Audio Processing*, vol. 10, pp. 363-370, 2002.

[115] R. D. Zilca and Y. Bistritz, "Distance-based Gaussian mixture model for speaker recognition over the telephone," in *Proceedings of ICSLP 2000*, pp. 1001-1003, 2000.

[116] V. Faber, "Clustering and the continuous k-means algorithm," *Los Alamos Science*, pp. 138-144, 1994.

[117] Y. Blanco-Archilla, Z. Santiago, and J. C. Principe, "Alternative statistical gaussianity measure using the cumulative density function," *International Conference on ICA and Signal Separation*, pp. 537-542, 2000.

[118] B. L. Appanna, M. Skosan, and D. J. Mashao, "Using high-level and low-level feature concatenation for speaker identification," in *Proceedings of PRASA 2004*, pp. 103-106, 2004.

[119] B. R. Wildermoth and K. K. Paliwal, "Use of voicing and pitch information for speaker recognition," in *Proceedings of the 8th Australian International Conference on Speech Science and Technology*, pp. 324-328, 2000.