Robust Speaker Recognition

Qin Jin

CMU-CS-07-001
January 2007

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
www.lti.cs.cmu.edu

Thesis Committee:
Tanja Schultz, Co-Chair
Alex Waibel, Co-Chair
Alan W. Black
Douglas A. Reynolds, MIT Lincon Laboratory

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in Language and Information Technologies

Keywords: Speaker Identification, Speaker Detection, Speaker Segmentation and Clustering, Far-Field, Phonetic Speaker Recognition, Person Identification

To my great parents Jianren and Lianrui, my dear husband Shimin, and my lovely daughter Ada.

Abstract

The automatic speaker recognition technologies have developed into more and more important modern technologies required by many speech-aided applications. The main challenge for automatic speaker recognition is to deal with the variability of the environments and channels from where the speech was obtained. In previous work, good results have been achieved for clean high-quality speech with matched training and test acoustic conditions, such as high accuracy of speaker identification and verification using clean wideband speech and Gaussian Mixture Models (GMM). However, under mismatched conditions and noisy environments, often expected in real-world conditions, the performance of GMM-based systems degrades significantly, far away from the satisfactory level. Therefore, robustness becomes a crucial research issue in speaker recognition field.

In this thesis, our main focus is to improve the robustness of speaker recognition systems on far-field distant microphones. We investigate approaches to improve robustness from two directions. First, we investigate approaches to improve robustness for traditional speaker recognition system which is based on low-level spectral information. We introduce a new reverberation compensation approach which, along with feature warping in the feature processing procedure, improves the system performance significantly. We propose four multiple channel combination approaches, which utilize information from multiple far-field microphones, to improve robustness under mismatched training-testing conditions. Secondly, we investigate approaches to use high-level speaker information to improve robustness. We propose new techniques to

model speaker pronunciation idiosyncrasy from two dimensions: the cross-stream dimension and the time dimension. Such high-level information is expected to be robust under different mismatched conditions. We also built systems that support robust speaker recognition. We implemented a speaker segmentation and clustering system aiming at improving the robustness of speaker recognition as well as automatic speech recognition performance in the multiple-speaker scenarios such as telephony conversations and meetings. We also integrate speaker identification modality with face recognition modality to build a robust person identification system.

Acknowledgments

First I would like to thank my advisors, Alex Waibel and Tanja Schultz, for the tremendous time, energy, and wisdom they invested in my Ph.D. education. Alex and Tanja taught me everything from choosing research topics, to performing high-quality studies, to writing papers and giving talks. Their guidance and support throughout the years are invaluable.

I would like to thank the other members of my Ph.D. thesis committee, Alan Black and Douglas Reynolds, for their thoughtful comments and invaluable suggestions that have improved the quality of the experimental results and the completeness of this thesis.

I would like to thank my past and current colleagues of ISL, who shared their friendship with me and made the laboratory a big family: Nguyen Bach, Keni Bernardin, Susanne Burger, Paisarn Charoenpornsawat, Matthias Eck, Hazim Ekenel, Wend-Huu (Roger) Hsiao, Christian Fugen, Linda Hager, Isaac Harris, Sanijika Hewavitharana, Chiori Hori, Fei Huang, Szu-Chen Jou, Ian Lane, Kornel Laskowski, Rob Malkin, John McDonough, Kristen Messinger, Florian Metze, Mohamed Noamany, Yue Pan, Matthias Paulik, Sharath Rao, Sebastian Stuker, Yik-Cheung Tam, Thomas Schaaf, Ashish Venugopal, Stephan Vogel, Matthias Wolfel, Jie Yang, Hua Yu, Ying Zhang, Bing Zhao.

My summer research intern at Johns Hopkins University in 2002 expanded my research experience. I would like to thank our group leader, Douglas Reynolds, who encouraged me, and other group members, Barbara Peskin, Jiri Navratil, Joe Campbell, Walter Andrews, David Klusacek, Andre Adami, Joy Abramson, Radu Mihaescu, who made the whole experience joyful and fruitful.

I thank my friends in Pittsburgh, who gave me their helps in my CMU graduate life and added a lot of fun to my life: Peng Chang, Mei Chen, Tao Chen, Xilin Chen, Zhaohui Fan, Bo Gong, Wei Hua, Chun Jin, Rong Jin, Fan Li, Hongliang Liu, Shuo Liu, Yan Liu, Yan Lu, Yong Lv,

Jiazhi Ou, Yanjun Qi, Yan Qu, Minglong Shao, Luo Si, Yanghai Tsin, Lisha Wang, Mengzhi Wang, Zhirong Wang, Rong Yan, Ke Yang, Jun Yang, Yiming Yang, Jing Zhang, Rong Zhang, Yi Zhang, Xiaojing Zhu.

Finally, I must express my deepest gratitude to my family. I owe a great deal to my parents, Jianren and Lianrui, who gave a life, endless love, and persistent encouragement to me. I am deeply indebted to my dear husband, Shimin, who brings to my life so much love and happiness. It is impossible for me to complete this seven and half years of long journey without his support. Last but not least, my two-year-old daughter, Ada, motivated me to finish my thesis with her sweet smiles and cuddles.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Spoken language is the most natural way used by humans to communicate information. The speech signal conveys several types of information. From the speech production point of view, the speech signal conveys linguistic information (e.g., message and language) and speaker information (e.g., emotional, regional, and physiological characteristics). From the speech perception point of view, it also conveys information about the environment in which the speech was produced and transmitted. Even though this wide range of information is encoded in a complex form into the speech signal, humans can easily decode most of the information. Such human ability has inspired many researchers to understand speech production and perception for developing systems that automatically extract and process the richness of information in speech. This speech technology has found wide applications such as automatic dictation, voice command control, audio archive indexing and retrieval etc.

The application defines which information in the speech signal is relevant. For example, the linguistic information will be relevant if the goal is to recognize the sequence of words that the speaker is producing. The presence of irrelevant information (like speaker or environment information) may actually degrade the system accuracy. In this thesis, we deal with automatic systems that recognize who is speaking (the speaker's identity) [35] [13].

It was Lawrence Kersta who made the first major step from speaker identification by humans towards speaker identification by computers when he developed spectrographic voice identification at Bell Labs in the early 1960s. His identification procedure was based on visual comparison of the spectrogram, which was generated by a complicated electro-mechanical device [58]. Although the visual comparison method cannot cope with the physical and linguistic variation in speech, his work encouraged the introduction of automatic speaker recognition. In the following four decades, speaker recognition research has advanced a lot. Some commercial systems have been applied in certain domains. Speaker Recognition technology makes it possible to use a person's voice to control the access to restricted services (automatic banking services), information (telephone access to financial transactions), or area (government or research facilities). It also allows detection of speakers, for example, voice-based information retrieval, recognition of perpetrator on a telephone tap, and detection of a speaker in a multi-party dialog.

Although the rapid development of speaker recognition technology is happening, there are still many problems to be solved. One problem is to understand what characteristics in the speech signal convey the representation of a speakers. This relates to understanding how humans listen to the speech signal and recognize the speaker. The other problem is to make automatic speaker recognition systems robust under different conditions.

1.1 Speaker Recognition Principles

Depending on the application, the general area of speaker recognition can be divided into three specific tasks: identification, detection/verification, and segmentation and clustering [35][13][91][92].

The goal of the *speaker identification* task is to determine which speaker out of a group of known speakers produces the input voice sample. There are two modes of operation that are related to the set of known voices. In the closed-set mode, the system assumes that the to-be-

determined voice must come from the set of known voices. Otherwise, the system is in open-set mode. The closed-set speaker identification can be considered as a multiple-class classification problem. In open-set mode, the speakers that do not belong to the set of known voices are referred to as impostors. This task can be used for forensic applications, e.g., speech evidence can be used to recognize the perpetrator's identity among several known suspects.

In *speaker verification*, the goal is to determine whether a person is who he or she claims to be according to his/her voice sample. This task is also known as voice verification or authentication, speaker authentication, talker verification or authentication, and speaker detection. It can be considered as a true-or-false binary decision problem. It is sometimes referred to as the open-set problem, because this task requires distinguishing a claimed speaker's voice known to the system from a potentially large group of voices unknown to the system. Today verification is the basis for most speaker recognition applications and the most commercially viable task. The open-set speaker identification task can be considered as the merger of the closed-set identification and open-set verification tasks. It performs like closed-set identification for known speakers but must also be able to classify speakers unknown to the system into an "unregistered speaker" category. Speaker verification can be used for security applications, such as, to control telephone access to banking services.

Speaker segmentation and clustering techniques are used in multiple-speaker scenarios. In many speech recognition and speaker recognition applications, it is often assumed that the speech from a particular individual is available for processing. When this is not the case, and the speech from the desired speaker is intermixed with other speakers, it is desired to segregate the speech into segments from the individuals before the recognition process commences. So the goal of this task is to divide the input audio into homogeneous segments and then label them via speaker identity. Recently, this task has received more attention due to increased inclusion of multiple-speaker audio such as recorded news show or meetings in commonly used web searches and consumer electronic devices. Speaker segmentation and clustering is one way to

index audio archives so that to make the retrieval easier.

According to the constraints placed on the speech used to train and test the system, Automatic speaker recognition can be further classified into text-dependent or text-independent tasks. In text-dependent recognition, the user must speak a given phrase known to the system, which can be fixed or prompted. The knowledge of a spoken phrase can provide better recognition results. In text-independent recognition, the system does not know the phrase spoken by the user. Although this adds flexability to an application, it can have redcued accuracy for a fixed amount of speech.

Running a speaker recognition system typically involves two phases.In the first phase, a user enrolls by providing voice samples to the system. The system extracts speaker-specific information from the voice samples to build a voice model of the enrolling speaker. In the second phase, a user provides a voice sample (also referred to as test sample) that is used by the system to measure the similarity of the user's voice to the model(s) of the previously enrolled user(s) and, subsequently, to make a decision. The speaker associated with the model that is being tested is referred to as target speaker or claimant.In a speaker identification task, the system measures the similarity of the test sample to all stored voice models. In speaker verification task, the similarity is measured only to the model of the claimed identity. The decision also differs across systems. For example, a closed-set identification task outputs the identity of the recognized user; besides the identity, an open-set identification task can also choose to reject the user in case the test sample do not belong to any of the stored voice models; a verification task chooses to accept or reject the identity claim.

1.2 Basic Structure of a Speaker Recognition System

Like most pattern recognition problems, a speaker recognition system can be partitioned into two modules: feature extraction and classification. The classification module has two compo-

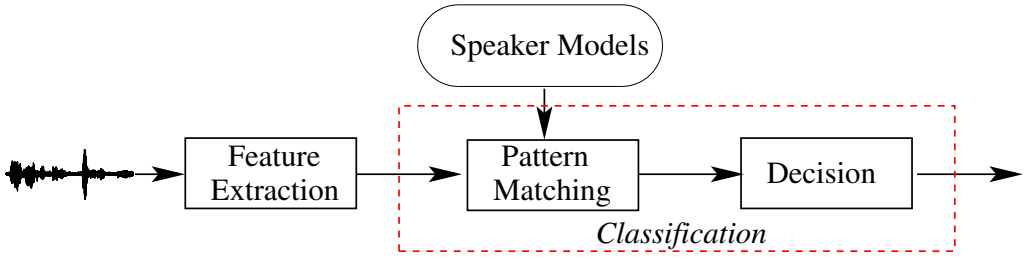

Figure 1.1: *Generic speaker recognition system*

nents: pattern matching and decision. Figure 1.1 depicts a generic speaker recognition system.

The feature extraction module estimates a set of features from the speech signal that represent some speaker-specific information. The speaker-specific information is the result of complex transformations occurring at different levels of the speech production: semantic, phonologic, phonetic, and acoustic [6], [13]. The semantic level deals with transformation caused on the speech signal according to the communicative intent and dialog interaction of the speaker. For example, the vocabulary choice and the sentence formulation can be used to identify the socio-economic status and/or education background of the speaker [81]. The phonological level deals with the phonetic representation of the communicative intent. For example, duration and selection of phonemes, intonation of the sentence can be used to identify the native language and regional information. The phonetic level deals with the realization of the phonetic representation by the vibration of the vocal cords and the movements of articulators (lips, jaw, tongue, and velum) of the vocal tract [88]. For example, speaker can use a different set of articulator movements to produce the same phoneme [81]. The acoustic level deals with the spectral properties of the speech signal. For example, the dimensions of the vocal tract, or length and mass of vocal folds will define in some sense the fundamental and resonant frequencies, respectively. Despite the variety of speaker-specific information, the set of features should have the following characteristics [81], [122]:

- occur naturally and frequently in normal speech

- be easily measurable
- have high variability between speakers
- be consistent for each speaker
- not change over time or be affected by the speaker's health
- not be affected by reasonable background noise nor depend on specific transmission characteristics
- show resistance to disguise or mimicry

In practice, not all of these criteria can be applied to the parameters used by the current systems.

The pattern matching module is responsible for comparing the estimated features to the speaker models. There are many types of pattern matching methods and corresponding models used in speaker recognition [13]. Some of the methods include hidden Markov models (HMM), dynamic time warping (DTW), and vector quantization (VQ). In open-set applications (speaker verification and open-set speaker identification), the estimated features can also be compared to a model that represents the unknown speakers. In a verification task, this module outputs a similarity score between the test sample and the claimed identity. In an identification task, it outputs similarity scores for all stored voice models. The decision module analyzes the similarity score(s) (statistical or deterministic) to make a decision. The decision process depends on the system task. For closed-set identification task, the decision can just select the identity associated with the model that is the most similar to the test sample. In open-set applications, the systems can also require a threshold to verify whether the similarity is valid. Since open-set application can also reject speakers, the cost of making an error need to be considered in the decision process. For example, it is more costly for a bank to allow an impostor to withdraw money, than to reject a true bank customer.

The effectiveness of a speaker recognition system is measured differently for different tasks. Since the output of a closed-set speaker identification system is a speaker identity from a set of known speakers, the identification accuracy is used to measure the performance. For the speaker detection/verification systems, there are two types of error: false acceptance of an impostor and false rejection of a target speaker. The performance measure can also incorporate the cost associated with each error, which depends on the application. For example, in a telephone credit card purchase system, a false acceptance is very costly; in a toll fraud prevention system, false rejection can alienate customers.

1.2.1 Acoustic Features

All audio processing techniques start by converting the raw speech signal into a sequence of acoustic feature vectors carrying characteristic information about the signal. This preprocessing module (feature extraction) is also referred to as "front-end" in the literature. The most commonly used acoustic vectors are Mel Frequency Cepstral Coefficients (MFCC) [22], Linear Prediction Cepstral Coefficients (LPCC) [70], and Perceptual Linear Prediction Cepstral (PLPC) Coefficients [45]. All these features are based on the spectral information derived from a short time windowed segment of speech. They differ mainly in the detail of the power spectrum representation. MFCC features are derived directly from the FFT power spectrum as shown in figure 1.2, whereas the LPCC and PLPC use an all-pole model to represent the smoothed spectrum. The mel-scale filterbank centers and bandwidths are fixed to follow the mel-frequency scale, giving more detail to the low frequencies. LPCC features can be considered as having adaptive detail in that the model poles move to fit the spectral peaks wherever they occur. The detail is limited mostly by the number of poles available. PLPC features are a hybrid between filterbank and all-pole model spectral representation. The spectrum is first passed through a bark-spaced trapezoidal-shaped filterbank and then fit with an all-pole model. The details of the PLPC representation is determined by both the filterbank and the all-pole

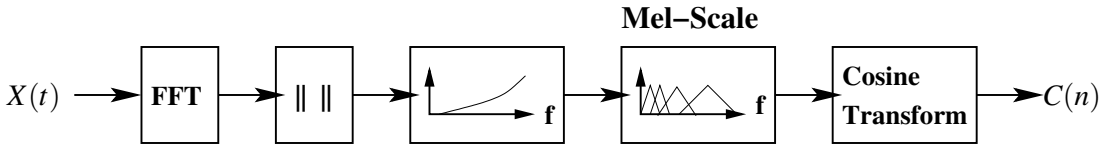

Figure 1.2: *Block-Diagram of Extracting MFCC*

model order. The spectral representation is transformed to cepstral coefficients as a final step. This is done because of the (near) orthogonalizing property of the cepstral transformation. The filterbank representations are transformed directly by a Discrete Cosine Transform (DCT). The all-pole representations are transformed using the recursive formula between prediction coefficients and cepstral coefficients [88]. In all cases, discarding the zeroth cepstral coefficient results in energy normalization. PLPC and MFCC features are used in most state-of-the-art automatic speech recognition systems [72] [123]. The effectiveness of LPCC features for automatic speaker recognition was shown in [5] [6]. However, MFCC features are used in more and more speaker recognition applications. For example, most of the participating systems in NIST speaker recognition evaluations in 1998 used MFCC features and some systems used LPCC features [27]. Following the trends in many state-of-the-art speaker recognition systems (e.g. [94]), MFCC coefficients (without energy term (C_0) and with derivatives) are used as acoustic feature vectors in this thesis, unless otherwise mentioned.

1.2.2 Gaussian Mixture Model (GMM)

A GMM is a mixture of several Gaussian distributions and is used to estimate the Probability Density Function (PDF) of a sequence of feature vectors. The likelihood of a model (GMM) given observation is then estimated as:

$$p(x_n|N(x_n,\mu_i,\Sigma_i)) = \sum_{i=1}^{M} \frac{w_i}{\sqrt{2\pi|\Sigma_i|}} exp\{\frac{-(x_n-\mu_i)^T\Sigma_i^{-1}(x_n-\mu_i)}{2}\} \tag{1.1}$$

where M is the number of Gaussian distributions in the GMM. The parameters of these distributions, w_i, μ_i, and Σ_i, are respectively the weight, mean, and diagonal covariance matrix of the i^{th} distribution in the GMM. Given a sequence of obeservation vectors, the parameters of a GMM can be trained via EM algorithm to maximize the likelihood of the data.

The observations in X are assumed to be independent and identically distributed (i.i.d.). Accordingly, the likelihood of a model (e.g. a GMM or an HMM) parameteried by θ given obeservation sequence X is estimated as:

$$p(X|\theta) = \prod_{n=1}^{N} p(x_n|\theta) \tag{1.2}$$

Although the strong assumption that the observations are independent conceals the temporal aspects of the speech signal, GMM has been extensively used for speaker modeling in text-independent speaker recognition applications [91]. The GMM has several properties that motivate their use for representing a speaker:

- One of the powerful properties of the GMM is its ability to form smooth approximations to arbitrarily shaped density. The GMM can be viewed as a parametric pdf based on a linear combination of Gaussian basis functions capable of representing a large class of arbitrary densities

- GMM can be considered as an implicit realization of probabilistic modeling of speaker dependent acoustic classes with each Gaussian component corresponding to a broad acoustic class such as vowels, nasals and fricatives etc.

1.2.3 Speaker Detection Framework

The goal of the speaker detection task is to determine whether a specified speaker is speaking during a speech segment. Since it is assumed that the speech segment has only speech from

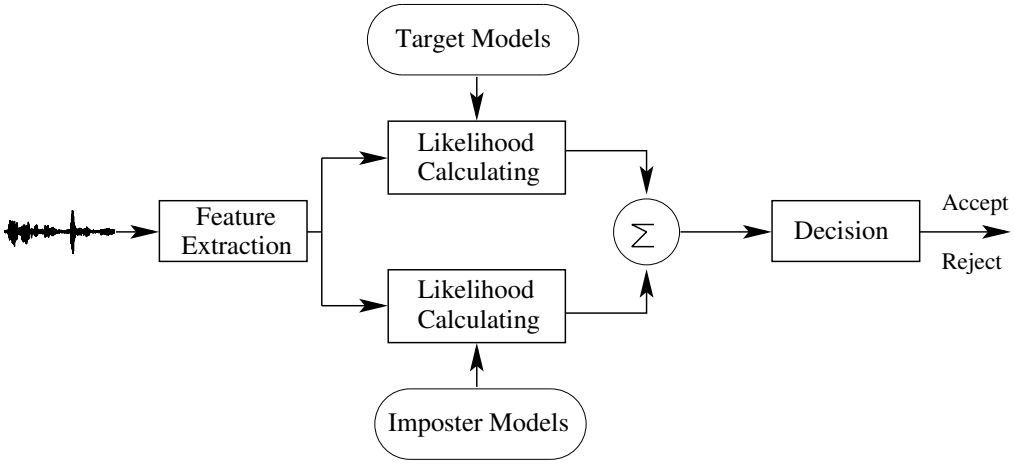

Figure 1.3: *Speaker detection system framework*

one speaker, this task is also known as single-speaker detection [94]. The problem of speaker detection can be formulated as a hypothesis testing of two mutually-exclusive hypotheses:

- H_0: target speaker is present,
- H_1: target speaker is not present.

Since there are only two hypotheses, the likelihood ratio test is used to make a decision. The likelihood ratio test is a comparison of the likelihood ratio between the two hypotheses and a threshold. A wrong decision can cause two types of errors. Type I error (miss) happens when the null hypothesis (H_0) is rejected when it is true. Type II (false alarm) error happens when the null hypothesis is accepted when the alternative hypothesis (H_1) is true. Furthermore, the application can determine a cost for every decision. For example, the cost of false acceptance decision has a more damaging effect than a false rejection decision in a telephone credit card purchase system. Therefore, the probability and costs associated with the errors have to be considered when making a decision rule (i.e., selecting the decision threshold).

Figure 1.3 shows the main components of a speaker detection system based on the likelihood ratio test. The features extracted from the test segment are used to compute the likelihoods of

the hypotheses. The null hypothesis is represented by the target-speaker model. The alternative hypothesis is represented by the impostor model that characterizes all the unknown speakers. The estimation of the likelihoods depends on specific models used to represent the target and imposter hypothesis. For example, a system can assume that the feature space can be represented by a Gaussian distribution, so that the models are the mean and variance parameters.

The target-speaker and impostor models are estimated a priori. The target-speaker models are estimated using training data from the respective speaker. The estimation of the impostor model poses a more complex task because it must represent the speaker space that is complementary to the target speaker. The method to define a speaker set that represents the speaker space is still under investigation [94], [100]. Typically, the set of unknown speakers can be a large number of speakers [95] or a collection of "cohort" speakers [99]. The impostor model is also know as universal background model (UBM) [94].

1.2.4 Speaker Segmentation and Clustering Framework

The goal of a speaker segmentation and clustering system is to divide a speech signal into a sequence of speaker-homogeneous regions. Thus, the output of such a system provides the answer to the question, "Who spoke when?". Knowing when each speaker is speaking is useful as a pre-processing step in automatic speech recognition(ASR) systems to improve the quality of the output. Such pre-processing may include vocal tract length normalization (VTLN) and speaker adaptation. Automatic speaker segmentation and clustering may also be useful in information retrieval and as part of the indexing information of audio archives.

Dividing an audio recording into speaker-homogeneous regions presents many challenges. One challenge is to identify the locations of the boundaries between speakers - the "speaker segmentation" problem. Another challenge is to identify which portions of the recording belong to which speakers - the "speaker clustering" problem. Additionally, the speaker clustering

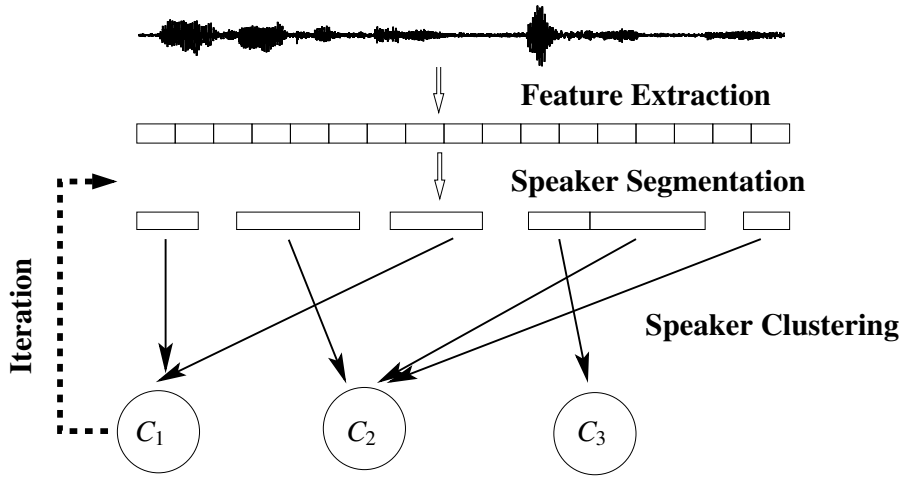

Figure 1.4: *Speaker Segmentation and Clustering System Flow*

problem requires that we correctly identify how many unique speakers occur in the recording. Speech researchers have proposed many techniques for solving the "Who spoke when?" problem. Most of these methods first segment and then cluster the data. The segmentation is either assumed to be known [17][109][113] or is performed automatically prior to clustering [108][42]. However, approaches such as these, in which the segmentation and clustering are performed sequentially, have limitations. In the former case, the correct segmentation is rarely known a priori for practical applications. In the latter case, the errors made in the segmentation step can degrade the performance of the subsequent clustering step.

Figure 1.4 shows a typical speaker segmentation and clustering system work flow. It usually contains two key steps: Speaker Segmentation and Speaker Clustering. After the feature vectors are extracted from the audio stream, the audio stream is divided into homogeneous segments according to speaker identity, environmental condition and channel condition, and then the speech segments are clustered into homogeneous clusters ideally according to speaker identity. Normally segmentation and clustering are conducted sequentially. However, segmentation

and clustering operations can also be conducted interactively, which means the segmentation process can use the clustering feedback and these two steps can iterate multiple times.

1.3 Robust Speaker Recognition Applications

Speaker recognition technologies have wide application areas. Here we list some example applications of speaker recognition technologies.

- Security: speaker recognition technologies can provide transaction authentication, facility or computer access control, monitoring, telephone voice authentication for long-distance calling or banking access etc.
- Personalisation: with speaker recognition technologies, we can implement intelligent answering machines with personalized caller greetings, we can build personlizied dialog systems: a dialog system can recognize the user, greet to the user directly, and direct the user through the system to destination successfully via shorter path according to the user's profile.
- Audio Indexing: speaker recognition technologies can provide automatic speaker labeling of recorded meetings for speaker-dependent audio indexing.
- Information Retrieval: speaker recognition can provide a way to manage and access the multimedia databases, which is to retrieve information according to interested speakers.
- Speaker Tracking: it is desired to know who is speaking in a tele-conference especially when there are many attandants in the tele-conference and the attendants are not very familiar with each other.

All these applications require robust speaker recognition techniques. For example in the telephone-aided services, users may call in under all kinds of acoustic conditions (in the office,

on the street etc.) and use different telephone networks (land-line or cellular). In the meeting scenarios, participants may talk while moving around facing the microphone in different directions and different distances. Mismatched conditions may be encountered at any time in these cases. Therefore robustness is one of the critical factors that decide the success of speaker recognition in these applications.

1.4 The Goal of the Thesis

The most significant factor affecting automatic speaker recognition performance is the variation in the signal characteristics (intersession variability and variability over time). Variations arise from the speakers themselves as well as from the recording and transmission channels, such as:

- Short-term variation due to the speaker's health and emotions
- Long-term changes due to aging
- Different microphones
- Different background noises (closed environment vs. open environment etc.)

It is well known that samples of the same utterance recorded within session are much more highly correlated than samples recorded in separate sessions. This is due to the fact that the speaker and channel effects are bound together in spectrum and hence speaker and channel characteristics are both involved in the features that are used in speaker recognition systems. Therefore anything that affects the spectrum can cause problems in speaker recognition. Unlike speech recognition systems, which may average out these effects using large amounts of speech, speaker recognition systems cannot do this since there is usually limited amount of enrolled speech. So it is important for speaker recognition systems to accommodate to these variations.

A majority of the speaker models, including the Gaussian mixture models, are based on modeling the underlying distribution of feature vectors from a speaker. When the speech is corrupted, the spectral based features are also corrupted and so their distributions are modified. Thus, a speaker model trained using speech from one type of corrupt environment will generally perform poorly in recognizing the same speaker using speech collected under different conditions since the feature distributions are now different. Various studies of speaker recognition systems using degraded or distorted speech have shown a dramatic decrease in performance [47] [38]. Current speaker recognition researches mainly focus on recognition under controlled conditions such as Switchboard telephone speech, which is close-talking speech. A large amount of effort is still needed in research about speaker recognition robustness under unlimited conditions in open environment with distant microphones.

In this thesis, we carry out research to improve the robustness for speaker recognition on distant microphones from two levels: to improve robustness for the traditional system based on low-level acoutstic features and to improve robustness using high-level features.From the low-level, we introduced a reverberation compensation approach and applied feature warping in the feature processing of the distant signals. We proposed multiple channel combination approaches to alleviate the issues of acoustic mismatches on far-field speaker recognition. From the high-level, we explored phonetic speaker recognition, in which we try to capture high-level phonetic speaker information and model speaker pronunciation dynamics using such information.

We also implement systems that support robust speaker recognition. We studied speaker segmentation and clustering and implemented a system aiming at good performance in multiple-speaker scenarios and to be portable across domains. We investigate its impact on automatic speech recognition as well. We also integrate the audio and modio person identification modalities (speaker idenitification and face recognition) to build a robust person identification system.

1.5 Contributions

In this thesis, we conduct research to improve speaker recognition robustness on far-field microphones from two levels with following contributions:

- We investigated far-field speaker recognition on multiple distant microphones, which is a research area has not received much attention. We introduced a reverberation compensation approach and applied feature warping in the feature processing. These approaches bring significant gain. We proposed four multiple channel combination approaches to utilize information from multiple sources to alleviate the channel mismatch effects. These approaches achieve significant improvement over baseline performance, especially in the case that test condition can not be covered in the training.

- We introduced a new approach to model a speaker's pronunciation idiosyncrasy from two complementary dimensions: time dimension and cross-stream dimension. Each dimension contains useful information for distinguishing speakers pronunciation characteristics. Combining both dimensions achieves significant better performance than that of each single dimension. The experimental results sugguest that the proposed approach is language independent. This research along with other research in phonetic speaker recognition has inspired ongoing research by others in using high-level features for speaker recognition. In addition, the proposed approach was applied to other classification tasks, such as language identification and accent identification, and achieved good performance as well.

- We studied speaker segmentation and clustering across domains such as telephone conversations and meetings. We implemented a speaker segmentation and clustering system which was tested within the NIST Rich Transcription evaluations. It is also a very important module in a complete ASR system, such as BN system, meeting system, and lecture recognition system etc. It provides crucial information for speaker adaptation.

- We integrated speaker recognition modality with face recognition modality and built a robust person identification system which was tested in the NIST CLEAR06 evaluation.

1.6 Thesis Organization

The rest of the thesis is organized as follows:

Chapter 2 describes far-field speaker recognition on distance microphones. Reverberation compensation, feature warping and four multiple channel combination approaches are introduced. We will show that all of them bring significant improvement for speaker identification on distant microphones. We also evaluate how our system perform under open-set mode.

Chapter 3 presents phonetic speaker recognition. It explains how we capture phonetic information to model speaker pronunciation dynamics in the time dimension and the cross-stream dimension. We will show that both dimensions contain useful information for distinguishing speakers and combining both dimensions can perform much better than using only one of the dimensions. We will also show its application on far-field speaker recognition.

Chapter 4 discusses speaker segmentation and clustering. We will show the system performance in different domains and discuss its impact on speech recognition.

Chapter 5 presents the person identification system which integrates speaker recognition and face recognition modalities. The system performance in the NIST CLEAR06 evaluation is presented.

Chapter 6 concludes the thesis and discusses some future directions.

Appendix A shows the performance of our speaker identification system under open-set mode and Appendix B presents some extra efforts of applying phonetic speaker recognition approaches on other tasks such as language identification and accent identification.

Chapter 2

Far-Field Speaker Recognition

2.1 Motivation

As discussed in previous chapter speaker recognition technology has a wide range of applications and many potential applications require hands-free sound capture, such as automatic teller machine authentication, the production of video conference transcripts, and security access to buildings or vehicles etc. In such applications, hands-free operation is preferable.

Speaker recognition has achieved fairly good performance under controlled conditions as reported in the NIST annual speaker recognition evaluation [80]. However, real world conditions differ from laboratory conditions. Mismatches exist between training and testing phases, such as wide band vs. narrow band, quite room environment vs. noisy street environment, and land-line channel vs. cell phone channel etc. These factors consequently induce performance degradation in automatic speaker recognition systems. The degradation becomes more prominent as the microphone is positioned more distant from the speaker [53], for instance, in a teleconferencing application. While the topic of far-field speech recognition has been investigated for some time, to date, speaker recognition has not received the same attention.

In this chapter we investigate techniques to improve the robustness of far-field speaker recog-

nition in the meeting scenarios with a multiple hands-free distant microphone setup. We introduce a new reverberation compensation approach, which uses a different noise estimation compared to the standard spectrum subtraction approach. We apply feature warping in the acoustic feature processing in our system. The experimental results show that significant improvements were achieved over the baseline system. Furthermore, multiple hands-free microphones are easy to setup and are realistic in many real scenarios. Therefore, we studied possible gains by using information from more than one far-field microphone. Four multiple channel combination approaches are investigated to capture useful information from multiple distant microphones. These approaches give additional large improvement over the baseline system.

2.2 Related Work

Accurate hands-free far-field speaker recognition is difficult due to a number of factors. Channel mismatch as well as environmental noise and reverberation are the two most prominent ones. During the past years, much research has been conducted towards reducing the effect of channel mismatch. Generally, robustness of a recognizer can be accomplished at three different levels:

- The acoustical level, giving rise to speech enhancement techniques that may improve the SNR of the input signal
- The parametric level, by means of parametric representations of speech characteristics which may show immunity to the noise process
- The modeling stage, combining adequate models of noise and clean signal in order to recognize noisy speech

To provide robustness to additive acoustic noise, for single channel condition, the well-known approach is the spectral subtraction procedure [9]. When the noise process is stationary

and speech activity can be detected, spectral subtraction is a direct way to enhance the noisy speech. For multi-sensor array condition, performing delay-and-sum beam-forming is the most direct approach [67] [41] [83]. The underlying idea of this scheme is based on the assumption that the contribution of the reflections is small, and that we know the direction of arrival of the desired signal. Then, through a correct alignment of the phase function in each sensor, the desired signal can be enhanced, rejecting all the noisy components not aligned in phase. Although microphone arrays have the benefit of providing a high level of enhancement, they need specific equipment and setup as well as knowledge of room acoustics or speaker's location to perform enhancement.

At the parametric level, the most well-known approaches to reduce mismatch are Cepstral Mean Subtraction (CMS) [33] and RASTA [46]. CMS and RASTA attempt to remove convolotional channel effects. In CMS the mean of the cepstral vectors is subtracted in order to high-pass filter the original cepstral coefficients:

$$y[n] = x[n] - \frac{1}{N}\sum_{i=1}^{N} x_i[n]$$

RASTA processing of speech again high-pass filter the cepstral coefficients with the following different equation:

$$y[n] = x[n] - x[n-1] + 0.97 * y[n-1]$$

However, channel mismatch and environmental noise can still cause lots of errors after CMS and RASTA. To deal with additive noise, a feature warping technique had been proposed that transforms the distribution of cepstral features to a standard distribution [83]. This technique was reported to give more improvement than standard techniques.

To provide robustness at the modeling level, one common approach is to assume an explicit model for representing environmental effects on speech features [36] [126] and use this model to construct a transformation which is applied either to the model space or to the feature space to decrease the mismatch. Though this model-based approach shows significant improvement,

it requires prior knowledge of noise statistics and extensive computation to adapt the models of clean speech to a new environment.

Our multiple channel combination approaches are implicitly related to the ensemble methods in machine learning. Ensemble methods are learning algorithms that construct a set of classifiers, then combine their individual decisions in some fashion, in order to classify new examples. It has been shown that even a combination of "weak" classifiers can result in a "strong" composite classifier, whose classification performance is much better than that of any single classifier. The ensemble approach is also known as the Fusion of Models, Mixture of Experts, Committee of Learners, Multiple Classifier System, Consensus Theory, as well as by other names. In [25], Dietterich gives a deeper analysis for when ensembles can improve performance, or why it is not possible to find a single classifier that works as well as an ensemble. He shows that the strength of ensembles lies in its competence and flexibility in dealing with the following three situations: the training data may not provide sufficient information to choose a single best classifier; the learning algorithm may not be able to solve the difficult search problem; and the hypothesis space may not contain the true function. The key issue of constructing a successful ensemble is that the individual classifiers need to perform better than random guessing and be diverse. Although we do not provide strict proof of whether these two factors hold for our combination approaches, the experimental results match our expectations.

2.3 Databases and Experimental Setup

2.3.1 3D Distant Microphone Database (3D DMD)

To investigate robust speaker recognition with distant microphones, a speaker database was collected at the Interactive Systems Laboratories (ISL) in a meeting room using multiple distant microphones. The left hand-side of Figure 2.1 illustrates the distant microphone setup. Five

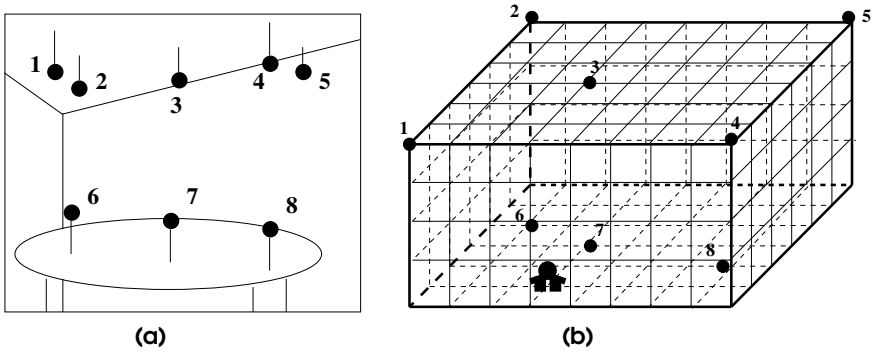

Figure 2.1: *Microphone setup in 3D DMD collection*

microphones (labeled as 1 to 5) are hanging from the ceiling, while three microphones (6, 7, and 8) are set up on the meeting table. We used miniature cardioid condenser microphones that are very similar to omni-directional microphones. The right hand-side of Figure 2.1 illustrates the positioning of these 8 microphones with respect to the speaker. The cubical grid indicates the distance defined by the grid, where each unit corresponds to 0.5 meters. The vertical grid is set to 4. Since the microphones are distributed in the 3D spaces we call this database 3D Distant Microphone Database in order to distinguish it with another database collected at ISL where the microphones are distributed in a 2D space as described in 2.3.2.

Since the speaker (sound source) is not opmi-directional, the microphones which have the same Euclidean distance to the speaker do not receive the same signal. Therefore, the distance of a speaker to a microphone is defined to be the Euclidean grid distance (horizontally and vertically) penalized by both the horizontal and vertical angles between the speaker (sound source) and the microphone (the receiver). For example, the distance of channel 6 is computed as follows:

$$D(6) = \frac{\sqrt{2^2+3^2}}{\cos(\arctan(\frac{2}{3}))} = 4.3 \tag{2.1}$$

which is the Euclidean distance in horizontal plane divided by the cosine of the angle between the sound source and receiver - microphone 6 - in horizontal plane. There is no vertical distance and no vertical angle penalty for this channel because the speaker sits at the table in the same

horizontal plane as the table microphone 6. For example, the distance of channel 2 is computed as follows

$$D(2) = \frac{\sqrt{3^2+5^2+4^2}}{\cos(\arctan(\frac{4}{\sqrt{34}}))\cos(\arctan(\frac{3}{5}))} = 10 \tag{2.2}$$

which is the Euclidean distance in both horizontal and vertical planes divided by the cosine values of the angle in horizontal plane and vertical plane respectively. The distance of other channels is computed similarly.

$$D(7) = \frac{2}{\cos(0)} = 2 \tag{2.3}$$

$$D(8) = \frac{\sqrt{1+3^2}}{\cos(\arctan(3)} = 10 \tag{2.4}$$

$$D(3) = \frac{\sqrt{2^2+4^2}}{\cos(\arctan(2)} = 10 \tag{2.5}$$

$$D(5) = \frac{\sqrt{4^2+5^2+4^2}}{\cos(\arctan(\frac{4}{5}))\cos(\arctan(\frac{4}{\sqrt{4^2+5^2}}))} = 11.4 \tag{2.6}$$

$$D(4) = \frac{\sqrt{1+4^2+4^2}}{[1-\cos(\arctan(4))]\cos(\arctan(\frac{4}{\sqrt{1+4^2}}))} = 12 \tag{2.7}$$

$$D(1) = \frac{\sqrt{1+3^2+4^2}}{[1-\cos(\arctan(3))]\cos(\arctan(\frac{4}{\sqrt{1+3^2}}))} = 14.5 \tag{2.8}$$

There are 24 speakers (4 female, 20 male) in total in the 3D Distant Microphone Database. Each speaker has one session recording, in which the speaker was required to talk about a selection of 10 given topics of personal interest in a spontaneous free speaking style. The speech duration varies from 8 minutes to 20 minutes depending on the subjects' verbosity. Two minutes of speech was randomly chosen from the first 80% of a speaker's entire recording as training data for that speaker. The remaining 20% of speech was split into 20 seconds segments, each of which is used as one test trial. Although using a single session for training and test will produce optimistic results, the degradation due to using microphones at varying locations is captured with this experimental design. There are in total 183 test trials. We assume that the test speaker is one of the enrolled speakers, which means closed-set speaker recognition is

evaluated in this chapter.

2.3.2 2D Distant Microphone Database (2D DMD)

A second database containing speech recorded from microphones at various distances was also collected at the Interactive Systems Laboratories in 2000. The room was different than that used for the 3D DMD. It was larger and more noisy. The database contains 30 speakers (16 female, 14 male) in total. From each speaker five sessions had been recorded where the speaker sits at a table in an office environment, reading an article. The articles are different for each session. Each session is recorded using eight microphones in parallel: one close-talking microphone (Sennheizer headset), one Lapel microphone worn by the speaker, and six other Lapel microphones. The latter six are attached to microphone stands sitting on the table or beyond the table, at distances of 1 foot, 2 feet, 4 feet, 5 feet, 6 feet and 8 feet to the speaker, respectively. Tables and graphs shown in this chapter use "Dis0" to represent close-talking microphone channel, "DisL" to represent speaker-worn microphone channel, and "DisN" ($N > 0$) to refer to the n-feet distance microphone channel. The upper part in Figure 2.2 gives an illustration of the overlook of the microphone arrangement with respect to the speaker. Different from the 3D Distant Microphone Database where the microphones are distributed in different horizontal and vertical planes, the microphones in this database are set in the same vertical plane, therefore we call it 2D Distance Microphone Database. The lower part in Figure 2.2 illustrates the microphone position in the same vertical space. Microphone Dis6 and Dis8, which stand on the floor beyond the table, are higher than the other six microphones.

For each speaker, we randomly select 60 seconds from the first session as training data. The remaining data was split into 20-seconds segments and used as test trials. There are in total 60 test trials.

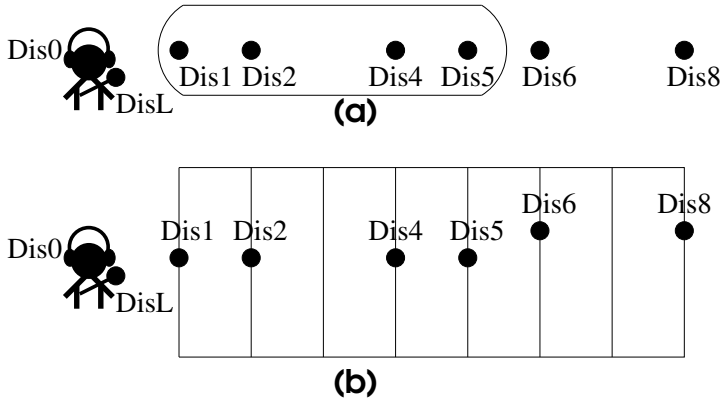

Figure 2.2: *Microphone setup in 2D DMD collection*

2.3.3 ICSI Meeting Database

The ICSI Meeting Database [50] is a collection of 75 meetings with simultaneous multi-channel audio recordings collected at the International Computer Science Institute (ICSI) in Berkeley. There are a total of 53 unique speakers in this corpus. We selected 24 speakers for training and testing based on their positions and whether they have enough total speaking time, Figure 2.3 is a simple diagram of the distant table microphone arrangement in the ICSI meeting room and the speaker position we selected. The table microphones are desktop omni-directional Pressure Zone Microphones (PZM). They were arranged in a staggered line along the center of the conference table. Ninety seconds of speech was randomly selected from meetings for each speaker as training data. The remainder speech was used for testing. We use the manual transcription to keep the test segments as they are if they were not longer than 20 seconds. Otherwise the segment is split into several 20 seconds chunks. There are 397 test trials in total.

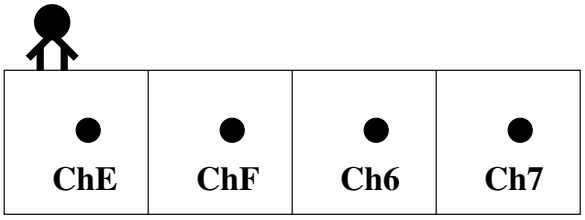

Figure 2.3: *Distant table microphone setup in ICSI meetings*

2.3.4 Speaker Modeling and Performance Measure

In our system a GMM with 128 mixtures was trained for each speaker using the EM algorithm. The identification decision is made as follows

$$s = \arg\max_{k} \left(L(X|\Theta^k) \right), k = 1, 2, \cdots, S \tag{2.9}$$

where s is the recognized speaker identity, S is the total number of speakers, and $L(X|\Theta^k)$ is the likelihood that the feature set X was generated by the GMM Θ^k of speaker k, which contains M weighted mixtures of Gaussian distributions

$$\Theta^k = (\lambda_m, \mathrm{N}(\mu_m, \Sigma_m)), m = 1, 2, \cdots, M \tag{2.10}$$

where M is the number of Gaussians and λ_m, μ_m, and Σ_m, are respectively the weight, mean, and diagonal covariance matrix of the m^{th} distribution in the GMM.

The system performance is measured using recognition accuracy, which is the percentage of correctly recognized test trials over all test trials.

2.4 Feature Processing to Far-Field Effects

2.4.1 Reverberation Compensation

A speech signal recorded with a distant microphone is more prone to be degraded by additive background noise and reverberation. Considering room acoustics as a linear shift-invariant system, the receiving signal $y(t)$ can be written as,

$$y[t] = x[t] * h[t] + n[t] \tag{2.11}$$

where the source signal $x[t]$ is the clean speech, $h[t]$ is the impulse response of room reverberation, and $n[t]$ is room noise. Cepstrum Mean Subtraction has been used successfully to compensate the convolution distortion. In order for CMS to be effective, the length of the channel impulse response has to be shorter than the short-time spectral analysis window which is usually 16ms-32ms. Unfortunately, the duration of impulse response of reverberation usually has a much longer tail, more than 50ms. Therefore traditional CMS will not be as effective under these conditions.

Following the work of Pan [82], we separate the impulse response $h[t]$ into two parts $h_1[t]$ and $h_2[t]$, where,

$$h[t] = h_1[t] + \delta(t-T)h_2[t]$$

$$h_1[t] = \begin{cases} h[t] & t < T \\ 0 & otherwise \end{cases}$$

$$h_2[t] = \begin{cases} h[t+T] & t \geq 0 \\ 0 & otherwise \end{cases}$$

and rewrite formula (2.11) as

$$y[t] = x[t] * h_1[t] + x[t-T] * h_2[t] + n[t]$$

$h_1[t]$ is a much shorter impulse response whose length is smaller than the DFT analysis window, thus it can be compensated by the conventional CMS. For $x[t-T] * h_2[t]$, we treat it the same as additive noise $n[t]$, and apply the noise reduction technique based on spectrum subtraction. Assuming the noise $x[t-T] * h_2[t] + n[t]$ could be estimated from $y[t-T]$, then the spectrum subtraction is performed as,

$$\hat{X}[t,\omega] = max(Y[t,\omega] - a \cdot g(\omega) Y[t-T,\omega], b \cdot Y[t,\omega])$$

where a is the noise overestimation factor, b is the spectral floor parameter to avoid negative or underflow values. We can empirically estimate the optimum a, b and $g(\omega)$ on a development dataset. We found that the system performance is not sensitive to T. Within the range of 20-40 ms there is no significant difference on the effect of the spectra subtraction. However outside that range, there is obvious performance degradation. For the recording setup in this thesis, we found $a = 1.0$, $b = 0.1$ and $g(\omega) = |1 - 0.9e^{j\omega}|$ optimal in most changing conditions based on development data as described in [82]. Standard CMS is applied after spectrum subtraction to eliminate the effect of $h_1[t]$.

2.4.2 Feature Warping

The feature warping method applied here was proposed in [83]. It warps the distribution of a cepstral feature stream to a standardized distribution over a specified time interval. The warping is implemented via CDF matching as described in [125]. The warping can be considered as a nonlinear transformation $\mathcal{T}$, which transforms the original feature X to a warped feature $\hat{X}$, i.e.,

$$\hat{X} = \mathcal{T}(X) \tag{2.12}$$

This can be done by CDF matching, which warps a given feature so that its CDF matches a desired distribution, such as normal distribution. The method assumes that the dimensions of the MFCC vector are independent. So each dimension is processed as a separate stream. The

CDF matching is performed over short time intervals by shifting a window. Only the central frame of the window is warped every time. The warping executes as follows:

- for $i = 1, \cdots, d$, where d is the number of feature dimensions
- sorting features in dimension i in ascending order in a given window
- warping raw feature value x in dimension i of the central frame to its warped value $\hat{x}$, which satisfies

$$\phi = \int_{-\infty}^{\hat{x}} f(y)dy \tag{2.13}$$

where $f(y)$ is the probability density function (PDF) of standard normal distribution, i.e.

$$f(y) = \frac{1}{\sqrt{2\pi}} exp(-\frac{y^2}{2}) \tag{2.14}$$

and ϕ is its corresponding CDF value. Suppose x has a rank r and the window size is N. Then the CDF value can be approximated as

$$\phi = \frac{(r - \frac{1}{2})}{N} \tag{2.15}$$

- $\hat{x}$ can be quickly found by lookup in a standard normal CDF table.

In our experiments, the window size is 300 frames and the window shift is one frame. Zeros are padded at the beginning and at the end of the raw feature stream.

2.4.3 Experimental Results for Noise Compensation

The front-end processing of the baseline system relies on MFCC analysis. The signal is characterized by 13-dimensional MFCC every 10ms. A speech detection process based on normalized energy is used in order to remove non-informative frames. The energy threshold is set empirically. The same threshold is applied on all microphone channels. The mean feature vector

Table 2.1: *Detailed baseline system performance (in %) on 3D DMD*

Test / Train	Ch1	Ch2	Ch3	Ch4	Ch5	Ch6	Ch7	Ch8
Ch1	**95.6**	94.0	76.0	83.6	72.7	77.6	71.6	83.1
Ch2	61.2	**100.0**	86.3	70.0	84.2	94.0	89.1	88.0
Ch3	38.3	63.4	**98.4**	49.2	59.0	71.6	78.7	78.7
Ch4	71.0	83.1	70.5	**87.4**	59.6	83.1	77.6	84.2
Ch5	54.1	86.9	76.0	59.6	**91.8**	85.3	84.7	84.7
Ch6	49.2	77.1	78.1	47.0	76.5	**90.7**	90.7	76.0
Ch7	38.8	68.9	75.4	52.5	72.1	86.3	**92.9**	80.9
Ch8	62.8	85.3	78.1	65.0	86.9	85.3	89.6	**95.1**

is computed on the informative frames only. The non-informative frames are discarded during training speaker models as well as in testing, which means only the informative frames are used to compute likelihood scores against speaker models. The baseline system consists of following components:

- speech detection: energy based
- front-end processing: 13-dimensional MFCC
- speaker models: GMM with 128 gaussian mixtures

The improved baseline system adds reverberation compensation (RC) and feature warping (Warp) in the front-end processing while keeping other system components the same as in the baseline system.

Table 2.1 presents the detailed speaker recognition accuracy of the baseline system when trained on different channels and tested on different channels using the 3D DMD. The rows

refer to different training channels and the columns refer to different test channels. For example, the number in row Ch1 and column Ch5 presents the recognition accuracy when test data is from channel 5 and speaker models are trained on data from channel 1. The table shows that accuracies under matched conditions (numbers in bold) are much better than under the mismatched conditions (off the diagonal). Again, by "matched condition" we mean that the training and test data are from the same channel, for example: both training and test are on channel 1 (microphone 1 in 3D DMD) and so on. By "mismatched condition" we mean that the training channel is different from the test channel, for example, test data is from channel 1 but the speaker models are trained on channel 2 etc.

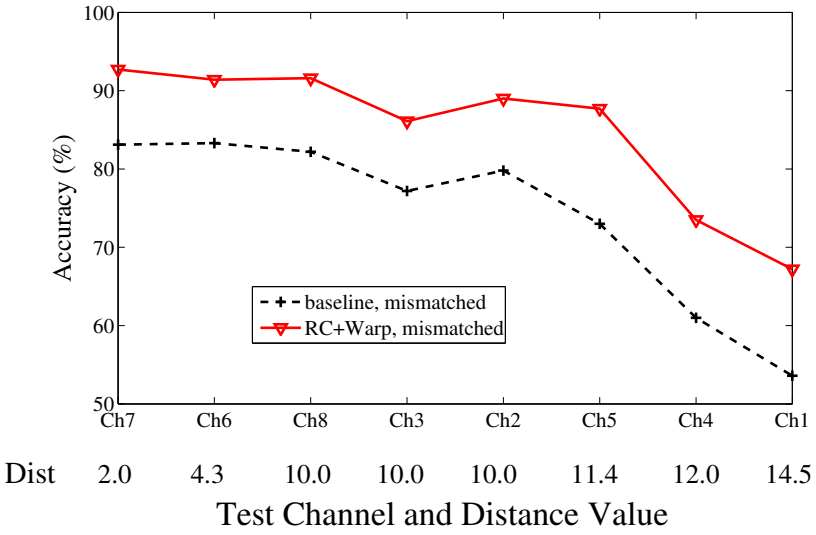

Figure 2.4: *Relationship between performance and distance on the 3D DMD*

Figure 2.4 shows the relationship between recognition accuracy and channel distance on the 3D Distant Microphone database. The distance is defined as in section 2.3.1. Apparently the performance is a function of the distance value: after surpassing a critical distance between speaker and microphone (mic 5,4,1) the performance decreases significantly. Please notice that microphone 1 and 4 are the two ceiling microphones behine the speaker.

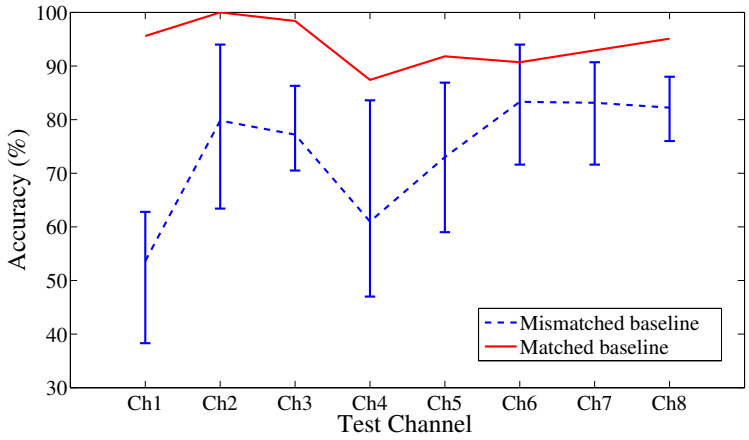

Figure 2.5: *Baseline performance under matched vs. mismatched conditions on 3D DMD*

Figure 2.5 summarizes the baseline system performance on different test channels under matched and mismatched conditions. The curve for the matched condition corresponds to those bolded numbers on the diagonal lines in table 2.1. The curve under mismatched conditions corresponds to the numbers computed by averaging the numbers in each column excluding the diagonal number in Table 2.1. For example, based on the 3D Distant Microphone Database, if we name the recognition accuracy as $Acc(Te_2Tr_1)$ for the case that test data is from channel 2 but the speaker models are trained on channel 1, then the average recognition accuracy ($Acc(Te_2)$) on test channel 2 under mismatched conditions is computed as:

$$Acc(Te_2) = \frac{1}{7} \sum_{j=1, j\neq 2}^{8} Acc(Te_2Tr_j)$$

So the bars in figure 2.5 refer to the range of the performance under different mismatched conditions for each test microphone channel. The average accuracies under matched and mismatched conditions are 94.0% and 74.2% respectively. We can see that the system performance degrades a lot under mismatched conditions. Also, the performance on one test channel under mismatched conditions varies when evaluated on speaker models trained on different channels.

Table 2.2: *RC and Warp impact on 3D DMD*

System	Matched	Mismatched
baseline	94.0	74.2
RC	94.8	78.1
relative improvement	(13.3%)	(15.1%)
Warp	96.4	79.1
relative improvement	(40.0%)	(19.0%)
RC+Warp	96.7	84.9
relative improvement	(45.5%)	(41.6%)

Table 2.2 shows the performance improvement by reverberation compensation alone, feature warping alone and reverberation compensation plus feature warping on the 3D Distant Microphone Database. Each of the two approaches improves performance under both matched and mismatched conditions. Combining both approaches provide more improvement, which indicates that both techniques take care of different aspects of degraded signal.

Figure 2.6 shows the reverberation compensation plus feature warping (RC+Warp) impact on system performances on all three data sets. We can see that significant improvements were achieved under both matched and mismatched conditions on all three data sets. On average, 45.5% and 41.6% relative improvements are achieved under matched and mismatched conditions respectively on the 3D Distant Microphone Database, 20.0% and 17.7% on the 2D Distant Microphone Database, and 31.9% and 34.1% on the ICSI Meeting Database, demonstrating that the applied methods are robust for different channel distances and under different recording conditions. Therefore, reverberation compensation and feature warping are used in the feature processing step in all the following experiments and we will refer to the performance with these two approaches applied over the baseline as “improved baseline”. Table 2.3 shows the detailed performance of the “improved baseline” system under both matched and

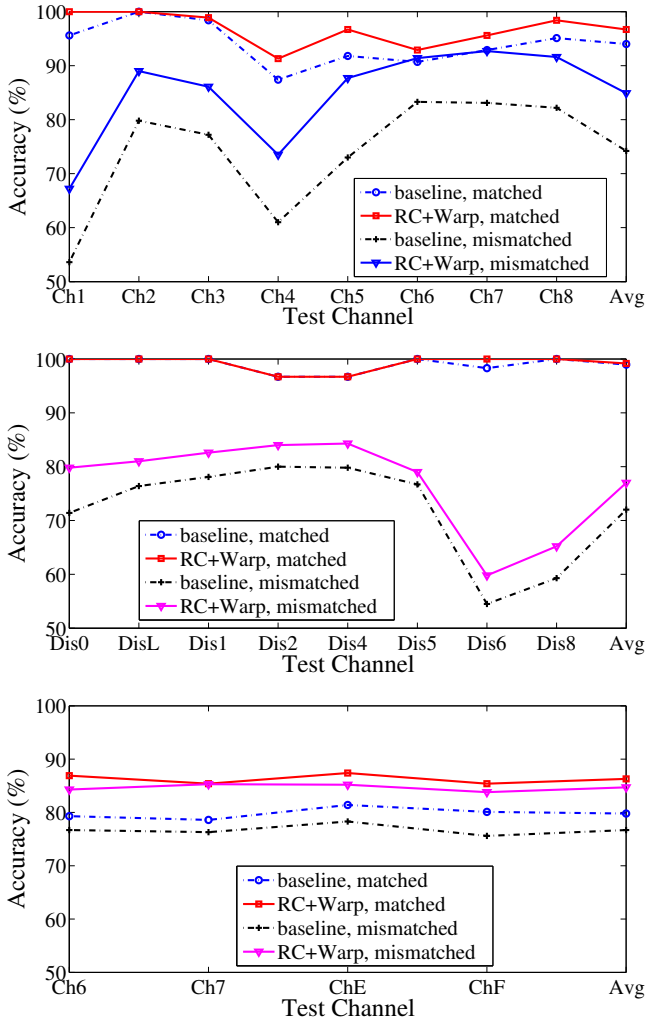

Figure 2.6: *Performance improvement by RC+Warp;*
upper: on 3D DMD, middle: on 2D DMD, lower: on ICSI Meeting Database

mismatched conditions. In the following sections we will show multiple channel combination approaches' improvement over this "improved baseline".

Table 2.3: *Improved baseline performance (in %) on 3D DMD*

Test Channel	Ch1	Ch2	Ch3	Ch4	Ch5	Ch6	Ch7	Ch8	Avg
Matched	100.0	100.0	98.9	91.3	96.7	92.9	95.6	98.4	**96.7**
Mismatched	67.2	89.0	86.1	73.5	87.7	91.4	92.7	91.6	**84.9**

2.5 Multiple Channel Combination

Hands-free multiple distant microphones are easy to set up and quite common in applications such as meetings and lectures. In order to benefit from the multiple channel setup, four multi-channel combination approaches are investigated which are: "Data Combination", "Frame based Score Competition", "Segment based Score Fusion", and "Segment based Decision Voting".

2.5.1 Data Combination (DC)

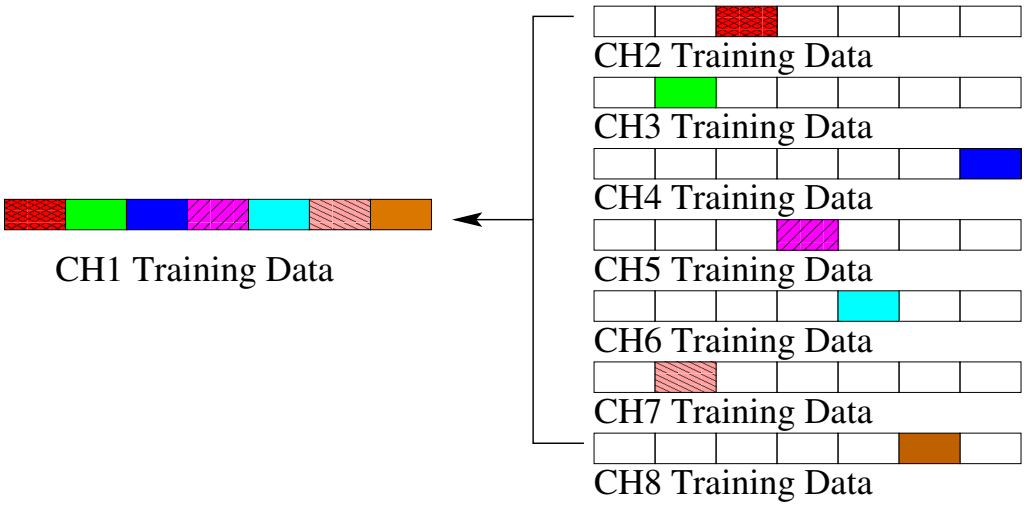

Figure 2.7: *Illustration of Data Combination on 3D DMD*

In "Data Combination" approach, the speaker models are trained using data from multiple mismatched channels. For example, on the 3D Distant Microphone Database, for test on channel 1, the speaker models are trained using data from all mismatched channels (channel 2 to channel 8) except the matched channel (channel 1). Consequently, the training data does not cover the test channel, so that the tests are performed under mismatched condition. In order to discriminate gains achieved by more data from those achieved by a larger variety of data, we keep the size of the training data the same as in the baseline system. As illustrated in figure 2.7, based on the 3D Distant Microphone Database, the training data for a speaker on channel 1 (CH1) is formed by randomly selecting $\frac{1}{7}$ data from the original training data on each of the mismatched channels (CH2 to CH8).

2.5.2 Frame based Score Competition (FSC)

Let us first review how a GMM system calculates likelihood scores and makes decisions based on the scores. The identification decision is made as follows

$$s^* = \arg\max_k \left(LL(X|\Theta^k) \right), k = 1, 2, \cdots, S \tag{2.16}$$

where s^* is the recognized speaker identity, S is the total number of enrolled speakers, and $LL(X|\Theta^k)$ is the log likelihood score that the entire test feature set X was generated by the GMM Θ^k of speaker k, which contains M weighted mixtures of Gaussian distributions as in 2.10.

The likelihood of an observation (for example one feature vector x_n) given a GMM model Θ^k (2.10) of speaker k is estimated as

$$p(x_n|\Theta^k) = \sum_{i=1}^{M} \frac{\lambda_i}{\sqrt{2\pi|\Sigma_i|}} exp\{\frac{-(x_n-\mu_i)^T\Sigma_i^{-1}(x_n-\mu_i)}{2}\} \tag{2.17}$$

Also, the entire set of feature vectors X are assumed to be independent and identically distributed (i.i.d.). Accordingly, the likelihood of observation sequence X given Θ^k is estimated

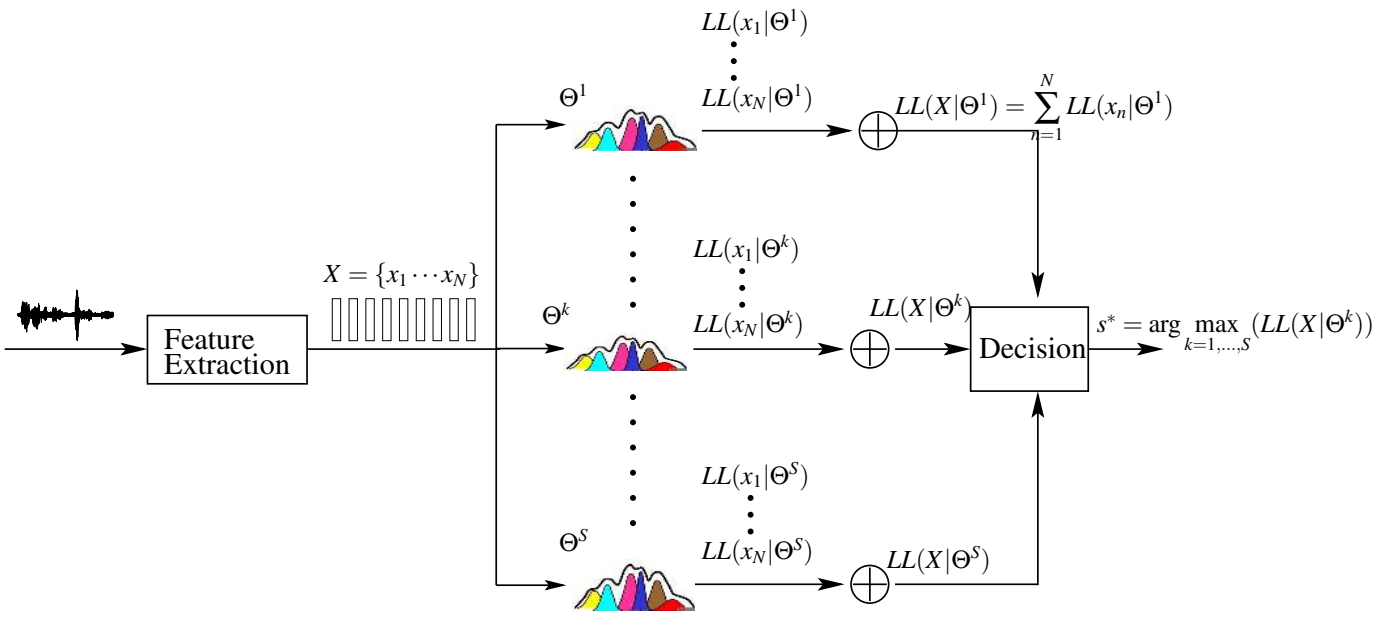

Figure 2.8: *Standard speaker recognition procedure*

as

$$p(X|\Theta^k) = \prod_{n=1}^{N} p(x_n|\Theta^k) \tag{2.18}$$

$$LL(X|\Theta^k) = \log p(X|\Theta^k) = \sum_{n=1}^{N} \log p(x_n|\Theta^k) = \sum_{n=1}^{N} LL(x_n|\Theta^k) \tag{2.19}$$

Figure 2.8 explains the standard GMM-based speaker recognition procedure. We call the log likelihood value as "score" in the following sections.

In the multiple microphone setup, if we have speech samples from different channels, we can build multiple models for each speaker with one for each channel. Let us name it as Θ^{k,Ch_i} for the GMM model of speaker k on channel i. So the model set for speaker k is $\Theta^k = \{\Theta^{k,Ch_1} \cdots \Theta^{k,Ch_C}\}$, where C is total number of channels. We propose this "Frame based Score Competition (FSC)" approach to compute the likelihood of an observation given a set of GMM models for each speaker. In this approach we compare a feature vector of each frame to all the GMMs $\{\Theta^{k,Ch_1} \cdots \Theta^{k,Ch_C}\}$ of speaker k excluding the one GMM which is trained on the same

channel as that of test samples. The highest log likelihood score is chosen as the score for this frame. So the log likelihood score of the entire set of test feature vectors X from channel h is estimated as

$$LL(X|\Theta^k) = \sum_{n=1}^{N} LL(x_n|\Theta^k) = \sum_{n=1}^{N} \max\{LL(x_n|\Theta^{k,Ch_j})\}_{j=1,j\neq h}^{C} \tag{2.20}$$

This competition process differs from the standard scoring process with only one microphone in that per-frame log likelihood scores for different speakers are not necessarily derived based on the same microphone. The FSC approach can be considered as the model version of the DC approach with increasing amount of training data.

Figure 2.9 illustrates the speaker recognition procedure with "Frame based Score Competition". Basically, the part that is circled in the standard procedure is replaced by the part that the arrow points to. The difference from the standard procedure lies in the score computation per frame.

2.5.3 Segment based Score Fusion (SSF)

By "segment" we refer to the entire test utterances or the entire set of test feature vectors X. The "Segment based Score Fusion" approach computes the score of test data given a set of models $\Theta^k = \{\Theta^{k,Ch_1} \cdots \Theta^{k,Ch_C}\}$ for speaker k by combining the scores from all the mismatched GMM models, each of which is trained on one of the mismatched channels.

$$LL(X|\Theta^k) = \sum_{j=1,j\neq h}^{C} w_j * LL(X|\Theta^{k,Ch_j}) \tag{2.21}$$

where C is the total number of channels, h is the test channel and w_j is the fusion weight.

Figure 2.10 illustrates the speaker recognition procedure with "Segment based Score Fusion". Again, the part that is circled in the standard procedure is replaced by the part that the arrow points to. The difference between this approach and the standard procedure lies in the score computation per segment.

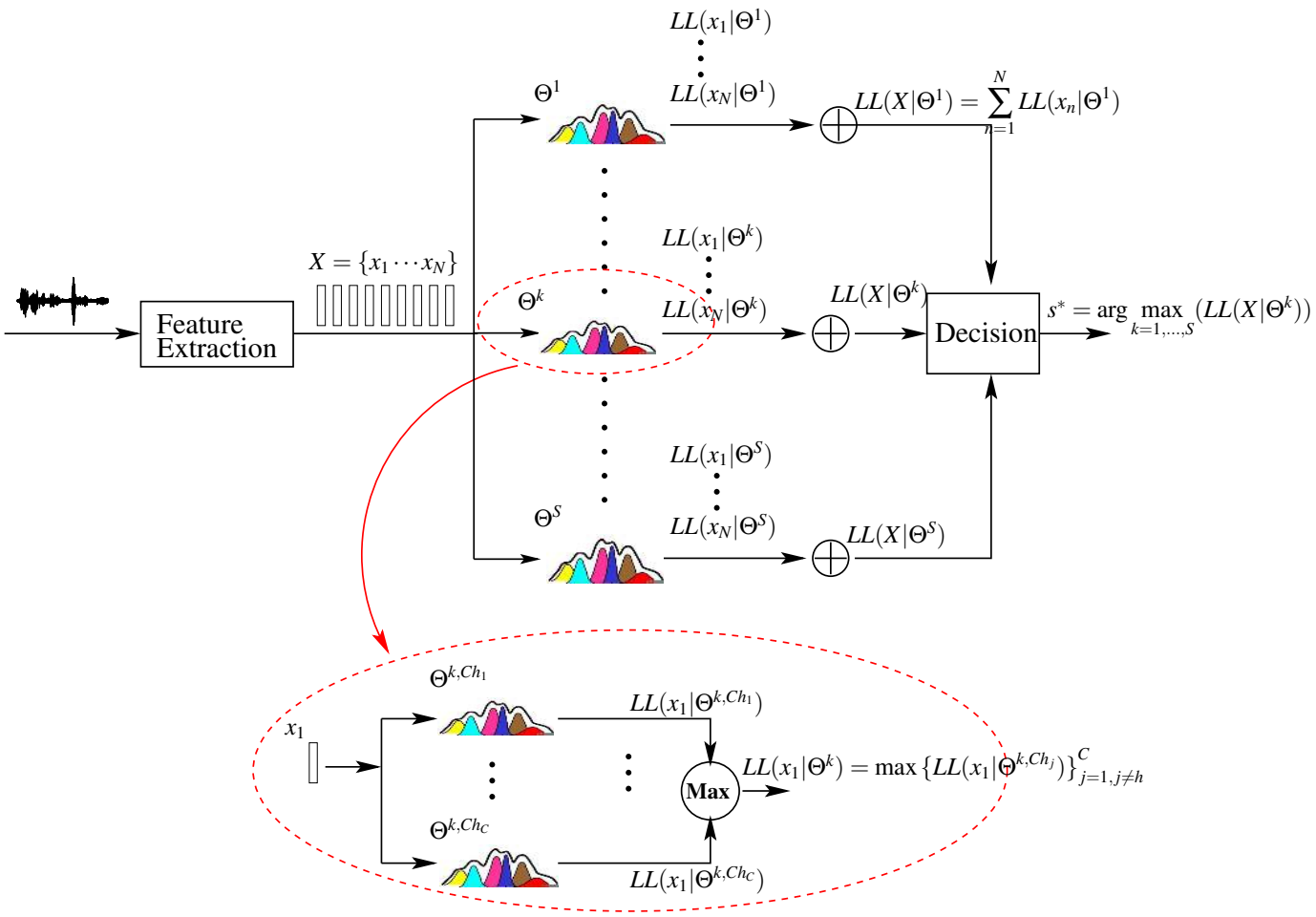

Figure 2.9: *Speaker recognition procedure with FSC*

2.5.4 Segment based Decision Voting (SDV)

Figure 2.11 illustrates the speaker recognition procedure with the approach "Segment based Decision Voting". In this approach, the entire set of feature vectors X extracted from the test trial goes through recognition part circled in the standard procedure multiple times. Each time the speaker models are trained on one of the mismatched channels. Therefore, the speaker identity decision is made multiple times $(C-1)$ with one on each mismatched channel. The identity which appears most times among these $C-1$ decisions is picked as the final decision.

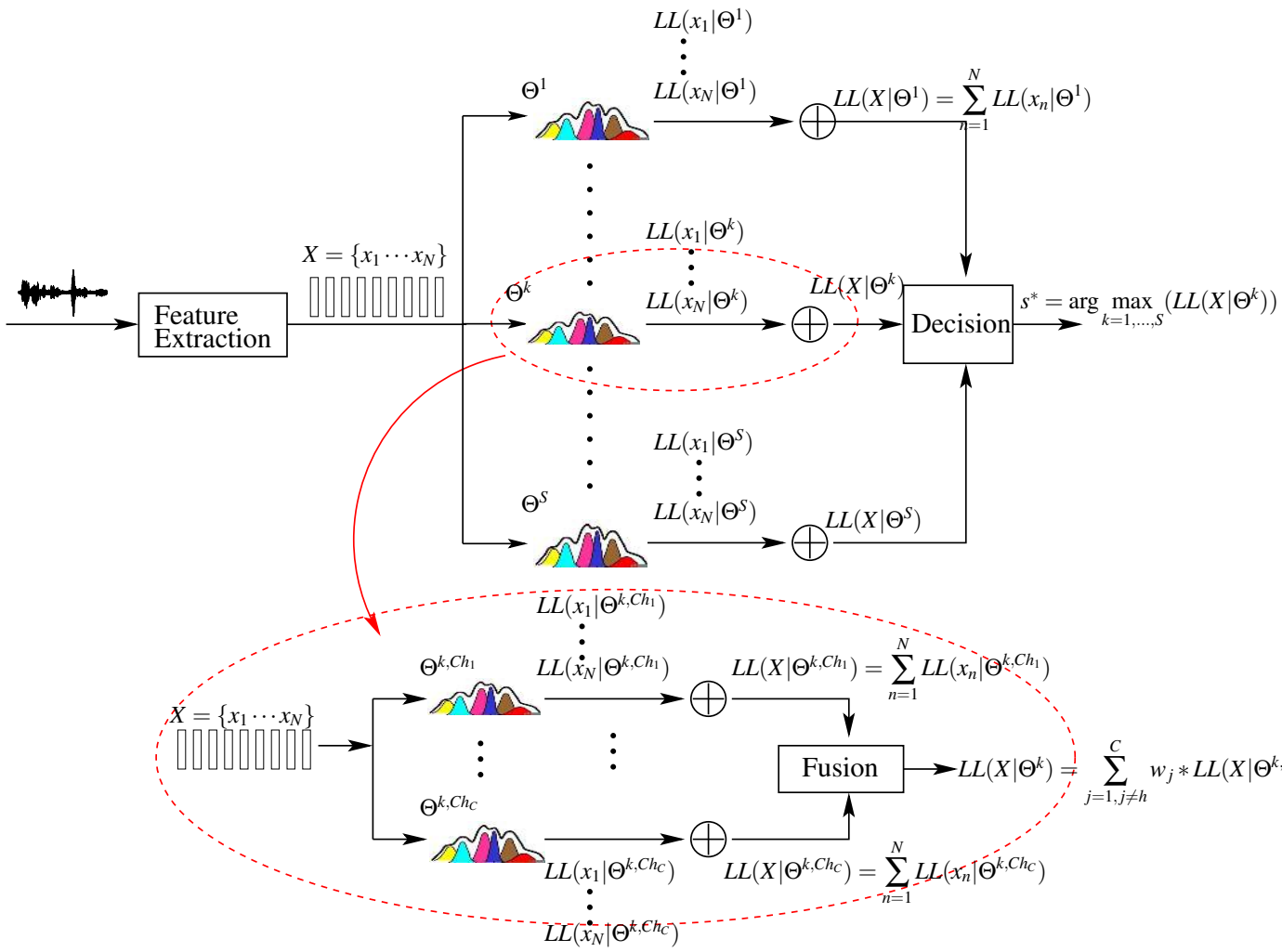

Figure 2.10: *Speaker recognition procedure with SSF*

If there is a tie, the one has the highest log likelihood score will be the winner.

2.5.5 Experimental Results for Multiple Channel Combination

Figure 2.12 presents improvements achieved by the four multi-channel combination approaches under mismatched conditions on the 3D Distant Microphone Database. Significant improvements are achieved by all combination approaches. On average, "Data Combination" brings

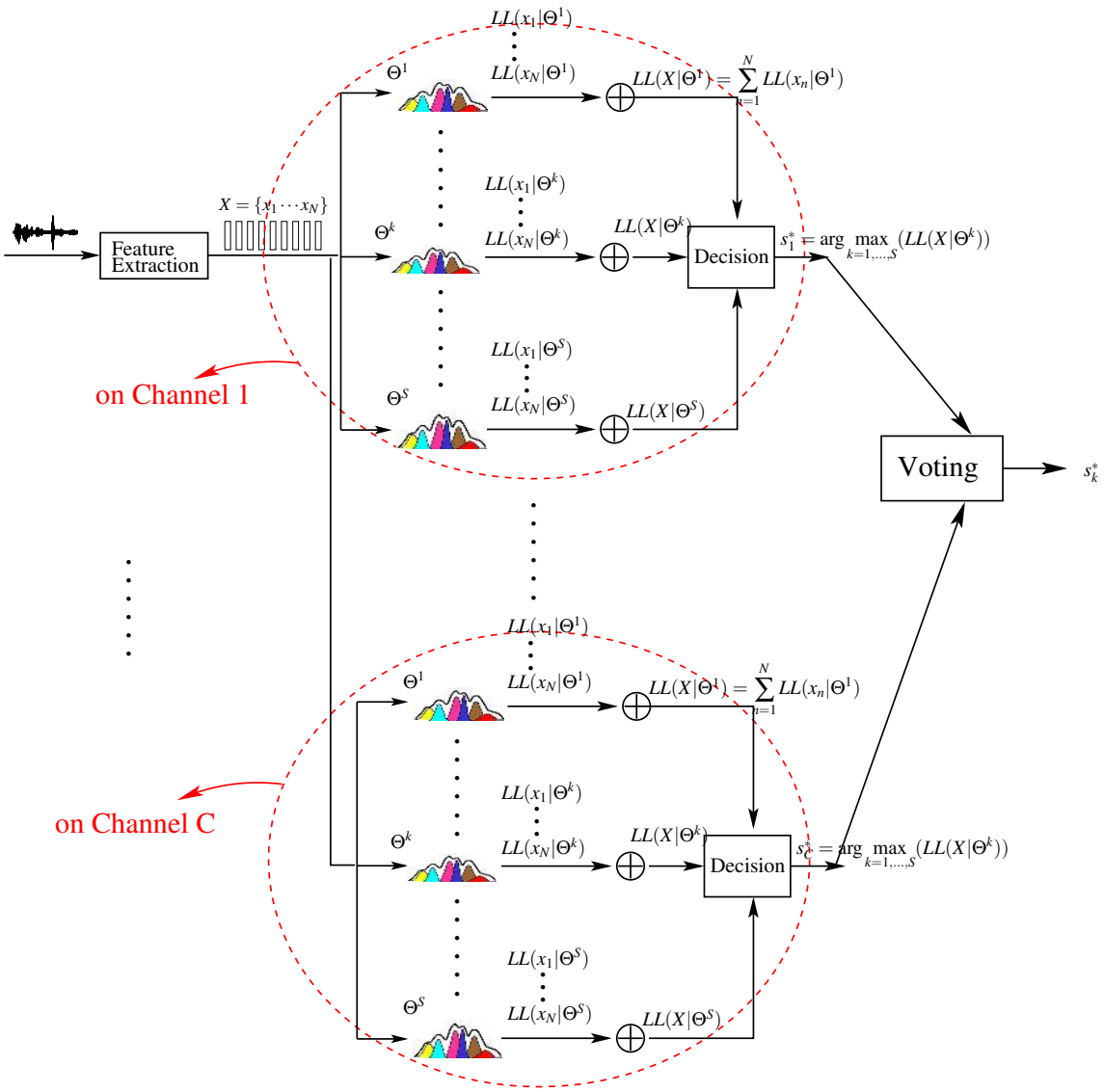

Figure 2.11: *Speaker recognition procedure with SDV*

72.8% relative improvement over the improved baseline and 84.1% relative improvement over the baseline under the mismatched condition, which means this combination approach achieves significant additional gain in addition to the reverberation compensation and feature warping approaches to the baseline. We will show the improvement over the improved baseline only in the following results since reverberation compensation and feature warping are always applied

in the feature processing. We want to point out that in "Data Combination" approach, we control the amount of training data to be the same as in the baseline system by randomly choosing $\frac{1}{7}$ data from each of the original mismatched channel. So the improvement indicates that seeing more variability in training improves the recognition robustness. 77.8% relative improvement was achieved over the improved baseline by "Frame based Score Competition" and 62.4% relative improvement over the improved baseline was achieved by "Segment based Score Fusion" and 57.9% relative improvement over the improved baseline was achieved by "Segment based Decision Voting". This indicates that it is beneficial to use information from multiple sources even though each of them is not very powerful. On the 2D Distant Microphone Database, 81.9%, 91.0%, 77.4%, and 64.7% relative improvement is achieved over the improved baseline under mismatched condition by respectively "Data Combination", "Frame based Score Competition", "Segment based Score Fusion", and "Segment based Decision Voting" approaches. On the ICSI Meeting Database, 9.7%, 11.4%, 6.8%, 3.5% relative improvement is achieved by respectively "Data Combination", "Frame based Score Competition", "Segment based Score Fusion", and "Segment based Decision Voting" approaches over the improved baseline under mismatched conditions.

Table 2.4 summarizes the relative improvements the four multiple channel combination approaches gained on the three databases. We can see that "Frame based Score Competition" approach achieves highest improvement among the four approaches while "Segment based Decision Voting" achieves lowest improvement among the four approaches.

2.5.6 Discussions

We have shown the impact of the four combination approaches' on the system performance under mismatched conditions. Note that this does not mean that we need to have the prior knowledge about which channel the test speech comes from. The whole purpose is to show that even if you have no samples of the test condition in your training, with the multiple channel

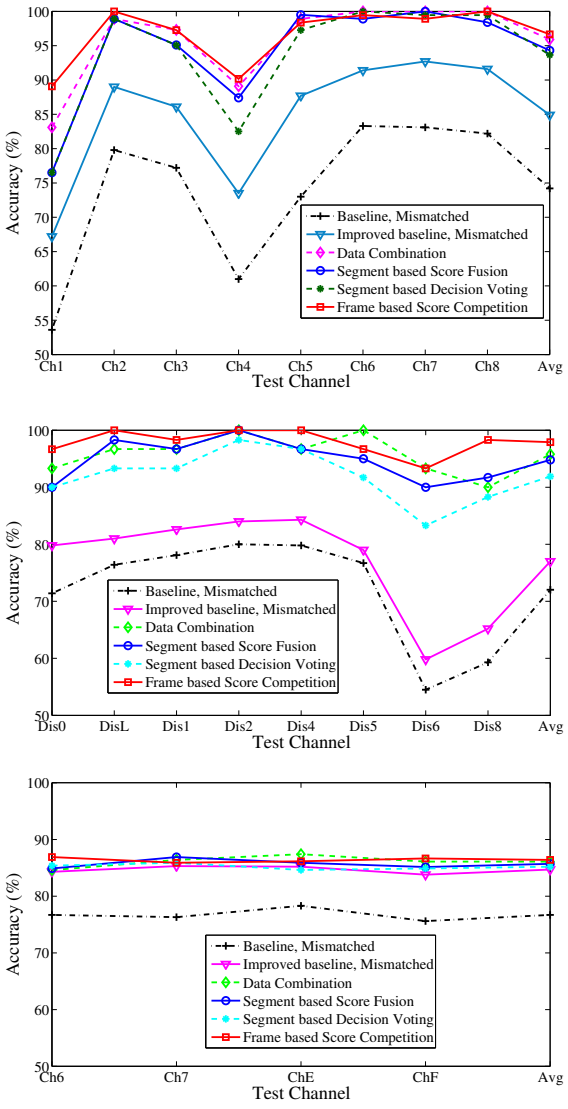

Figure 2.12: *Performance improvement by combination approaches; upper: on 3D DMD, middle: on 2D DMD, lower: on ICSI Meeting Database*

combination approaches you still can get very good performance. The next question is then if you combine all the channels including the matched channel, will the performance get better? Our expectation is that it will be better than the performance of combining only mismatched channels. But will the performance beat the one under the matched conditions?

Table 2.4: *Relative improvement by multiple channel combination approaches*

Database / Approach	3D DMD	2D DMD	ICSI
Data Combination	72.8%	81.9%	9.7%
Frame based Score Competition	77.8%	91.0%	11.4%
Segment based Score Fusion	62.4%	77.4%	6.8%
Segment based Decision Voting	57.9%	64.7%	3.5%

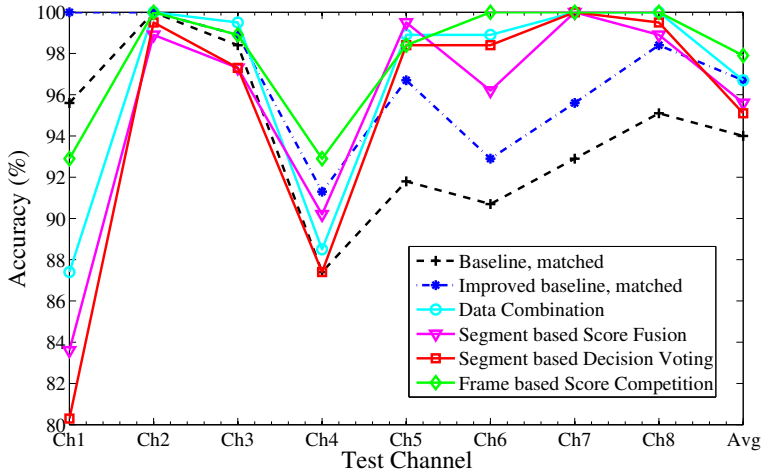

Figure 2.13: *Impact of combination approaches when applied on all channels on the 3D DMD*

Figure 2.13 compares the performance when combining all channels with the four combination approaches with that of the improved baseline under matched conditions. We can see that "Frame based Score Competition" approach and "Data Combination" approach beat the improved baseline performance under matched conditions. Although the other two combination approaches can not beat the improved baseline under matched conditions, the performance are compatible.

When less training data are available, the combination approaches become more important.

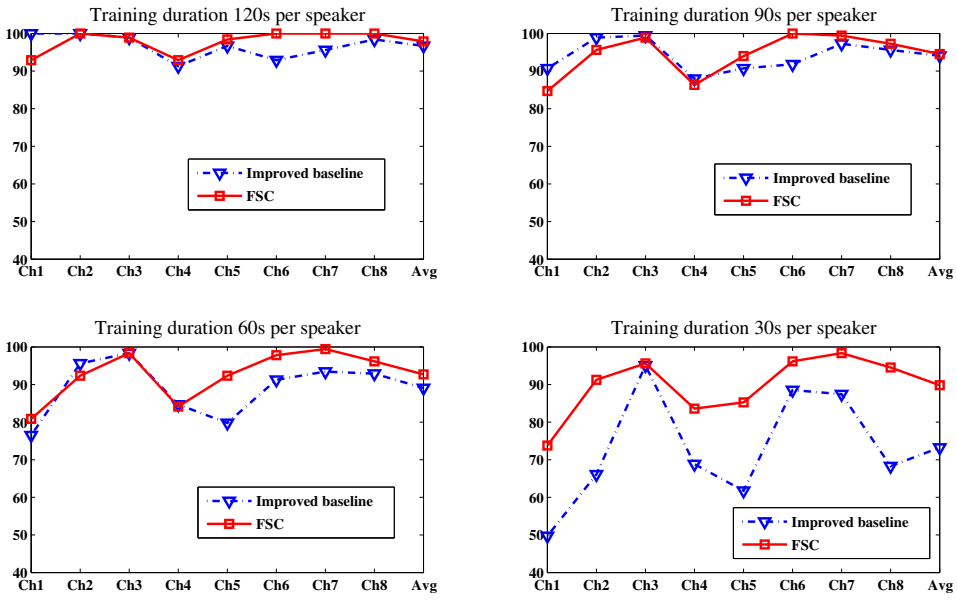

Figure 2.14: *Impact of FSC when applied on all channels with different training durations on 3D DMD*

Table 2.5: *Relative improvement by FSC with different training durations*

Training Duration	Baseline, matched	FSC	Relative Improvement
120s	96.7%	97.5%	24.2%
90s	94.1%	94.5%	8.0%
60s	89.1%	92.7%	33.1%
30s	73.2%	89.8%	62.0%

Figure 2.14 compares improved baseline under matched conditions vs. "Frame based Score Competition" performance when training durations per speaker vary. We see that the performance difference between improved baseline under matched conditions and FSC gets larger when less training data is available. Table 2.5 summarizes the average performance of improved baseline under matched conditions, FSC, and the relative improvement that FSC gains

over the improved baseline. More relative improvement is achieved by FSC when training duration gets shorter.

2.6 Chapter Summary

In this chapter we presented our robust speaker recognition system in a meeting scenario with multiple distant microphones. We applied a new reverberation compensation approach plus feature warping in the feature processing step. These two approaches significantly improved the system robustness under both matched and mismatched training-testing conditions. A 41.6% relative improvement is achieved on the 3D Distant Microphone Database, a 17.1% relative improvement is achieved on the 2D Distant Microphone Database, and a 34.1% relative improvement is achieved on the ICSI Meeting Database under mismatched conditions. Four multi-channel combination approaches are investigated in order to capture useful information from multiple channel sources including "Data Combination (DC)", "Frame based Score Competition (FSC)", "Segment based Score Fusion (SSF)", and "Segment based Decision Voting (SDV)". All these four approaches bring additional gains to the system performance under mismatched conditions. We observed 72.8.1%, 77.8%, 62.4%, and 57.9% relative improvements over the improved baseline on the 3D Distant Microphone Microphone Database by DC, FSC, SSF, and SDV respectively. The improvement carries over to the other two databases. We observed 81.9%, 91.0%, 77.4%, and 64.7% relative improvements on the 2D Distant Microphone Database. We observed 9.7%, 11.4%, 6.8%, and 3.5% relative improvements on the ICSI Meeting Database. The experimental results show that seeing more variability in training and combining supplementary information from multiple sources improves the system robustness. These approaches are effective across data set with different multiple distant microphone settings.

Table 2.6 shows the improvements over the baseline under mismatched conditions by all the

Table 2.6: *Relative improvement by reverberation compensation, feature warping, and multiple channel combination approaches*

Database Approach	3D DMD	2D DMD	ICSI
RC+Warp+Data Combination	84.1%	85.1%	40.5%
RC+Warp+Frame based Score Competition	87.1%	92.6%	41.6%
RC+Warp+Segment based Score Fusion	78.1%	81.4%	38.6%
RC+Warp+Segment based Decision Voting	75.4%	71.0%	36.4%

approaches together including reverberation compensation and feature warping in the feature processing step and four multiple channel combination approaches.

Chapter 3

Phonetic Speaker Recognition

3.1 Motivation

What do we rely on in the speech signal to recognize a speaker's identity? This is one of the central questions addressed by automatic speaker recognition research. Generally, humans often have the ability of recognizing speakers from the speech signal using multiple levels of speaker information conveyed in the speech signal [101]. This even works under various conditions and contexts. At the lowest level, we recognize a person based on sounds patterns in his/her voice (e.g., low/high pitch, bass/tenor, nasality, etc.). But we also use other types of information in the speech signal to recognize a speaker, such as a unique laugh, particular phrase usage, or speed of speech, among other things. The human performance seems to be a result of the robust and adaptive method of exploiting several levels of information [62] [63] [66].

Roughly we can categorize these information sources into a hierarchy running from low-level perceptual cues, related to physical traits of the vocal apparatus, to high-level perceptual cues, related to learned habits and style. Figure 3.1 shows the hierarchy of the perceptual cues [44] [92].

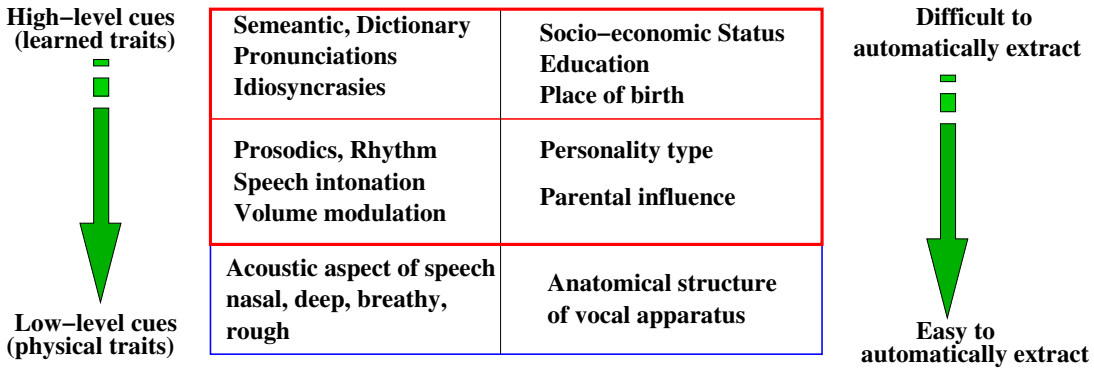

Figure 3.1: *Hierarchy of Perceptual Cues*

The high-level cues include such as word usage (idiolect), pronunciation, prosody, laughter, and other idiosyncratic supra-segmental information. These cues may be decided by a person's life situation such as socio-economic status, personality, education, etc.. Thus these cues are also termed as "learned traits". The low-level cues, on the other hand, are more directly related to the actual sound of a person's voice decided by the physical traits of a speaker's vocal apparatus. Although all levels of cues carry speaker information and are used by humans to identify speakers, automatic speaker recognition systems have relied almost exclusively on low-level information via short-term features related to the speech spectrum [95]. Traditional systems have several drawbacks. First, robustness is an issue because channel effects can dramatically change the measured acoustics of a particular individual. For instance, a system relying only on acoustics might have difficulty confirming that an individual speaking on a land-line telephone is the the same as an individual speaking on a cell phone [87]. Second, traditional systems also rely upon different methods than human listeners [102]. Human listeners are aware of prosody, word choice, pronunciation, accent, and other speech habits (laughs etc.) when recognizing speakers, while traditional systems only rely on seemingly one source of information. Due to the use of multiple high-level cues, human listeners are less affected by various conditions and

context than traditional automatic algorithms.

Observing the human speech processing model, we can improve the reliability and accuracy of speaker recognition systems by exploiting other sources of information in the speech signal. Not only the addition of information can improve the system accuracy by providing extra levels of discriminative information, but also it can increase the robustness by providing information that is less susceptible to degradation under varying condition and contexts. Furthermore, the published research in [28] and [60], before we started this piece of thesis research, tried to use n-grams counts on word and phone sequences for speaker verification and provided strong indications that potential gains are possible by the inclusion of higher levels of information available in the speech signal.

In this thesis, we propose approaches to capture the high-level phonetic information and to model a speaker's pronunciation idiosyncrasy based on this high-level information. We enrich the existing phonetic speaker recognition algorithms, which are based on ngram counts on phone sequences independently in multiple languages, by proposing new approaches to model dependencies across multiple phone streams.

3.2 Related Work

Most conventional speaker recognition systems use Gaussian mixture models (GMMs) to capture frame-level characteristics of a person's voice, where the speech frames are assumed to be independent of one another. Because of this independence assumption, GMMs often fail to capture certain types of speaker-specific information that evolve over time scales of more than one frame. For example, since words usually span many frames, GMMs tend to be poorly suited for modeling differences in word usage (idiolect) between speakers. In recent times, automatic speaker recognition research has expanded from utilizing only the acoustic content of speech to examining the use of higher levels of speech information, commonly referred to as

"high-level features." A promising direction in high-level feature research has been the use of n-gram based models to capture speaker specific patterns in the phonetic and lexical content of speech. In [28], Doddington performed an important initial study about using the lexical content of speech for speaker recognition, and introduced an n-gram based technique for modeling a speaker's idiolect. This direction in research was continued by Andrews, Kohler, and Campbell among others [60] [4], who used similar n-gram based models to capture speaker pronunciation idiosyncrasies through analysis of automatically recognized phonetic events. This line of research is generally referred to as "Phonetic Speaker Recognition." The research of Andrews et al. and Doddington showed word and phone n-gram based models to be quite promising for speaker recognition. There have been myriad attempts, especially since the Johns Hopkins 2002 Workshop [93] [52] [77] [59] [1] to harness the power of all kinds of high-level features.

The current "state-of-the-art" in phonetic speaker recognition uses relative frequencies of phone n-grams as features for training speaker models and for scoring test-target pairs [60] [4]. Typically, these relative frequencies are computed from a simple 1-best phone decoding of the input speech. This line of phonetic speaker recognition research work has been extended in various ways by introducing different modeling strategies and different methods of utilizing the source information such as described in [77] [59] [14] [43].

Navratil [77] proposed a method involving binary-tree-structured statistical models for extending the phonetic context beyond that of standard n-gram (particularly bigrams) by exploiting statistical dependencies within a longer sequence window without exponentially increasing the model complexity, as is the case with n-grams. The described approach confirms the relevance of long phonetic context in phonetic speaker recognition and represents an intermediate stage between short phone context and word-level modeling without the need for any lexical knowledge. Binary-tree models represent a step towards flexible context structuring and extension in phonetic speaker recognition, consistently outperforming standard smoothed bigrams as well as trigrams.

Klusacek [59] proposed a conditional pronunciation modeling method. It uses time-aligned streams of phones and phonemes to model a speaker's specific pronunciation. The system uses phonemes drawn from a lexicon of pronunciations of words recognized by an automatic speech recognition system to generate the phoneme stream and an open-loop phone recognizer to generate a phone stream. The phoneme and phone streams are aligned at the frame level and conditional probabilities of a phone, given a phoneme, are estimated using co-occurrence counts. A likelihood detector is then applied to these probabilities for the speaker detection task. This approach achieves a relatively high accuracy in comparison with other phonetic methods in the SuperSID project at the Johns Hopkins 2002 Workshop [114] [90].

Campbell [14] performed phonetic speaker recognition with support vector machines (SVM). By computing frequencies of phones in conversations, speaker characterization was performed. A new kernel was introduced based on the standard method of log likelihood ratio scoring. The resulting SVM method reduced error rates dramatically over standard techniques.

Hatch [43] compared 1-best phone decodings vs. lattice phone decodings for the purposes of performing phonetic speaker recognition. The results indicate that lattice decodings provide a much richer sampling of phonetic patterns than 1-best decodings.

All the state-of-the-art phonetic speaker recognition approaches try to model phonetic dependencies along the time scale, or in time dimension. In the following sections, we will present our contributions in the phonetic speaker recognition research. We introduce a phonetic speaker recognition approach that aims at modeling the statistical pronunciation patterns based on the phonetic information from two "orthogonal" dimensions: time dimension and cross-stream dimension. It will be shown that comparable or better results are achieved by the proposed approach.

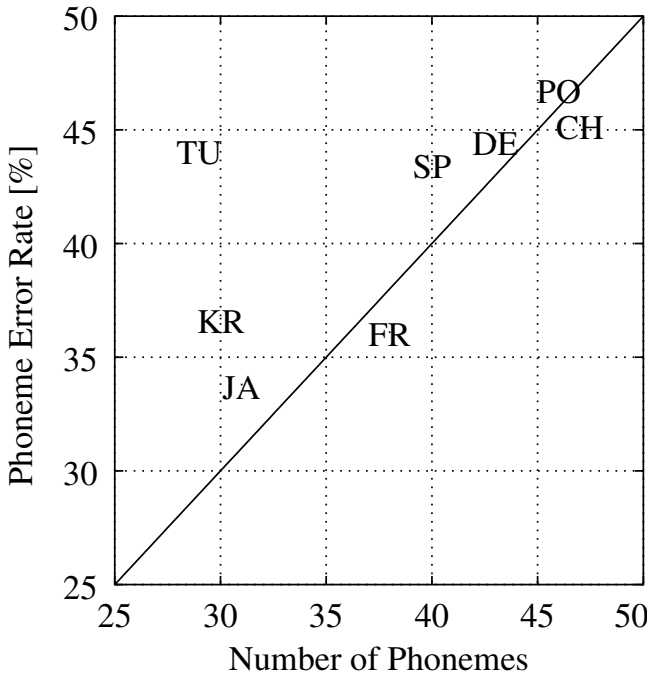

Figure 3.2: *Error rate vs number of phones in 8 languages*

3.3 Phone Sequence Extraction

Phone sequence extraction for the speaker recognition process is performed using the GlobalPhone Phone Recognizers. We have phone recognizers built in twelve languages available: Arabic (AR), Mandarin Chinese (CH), German (DE), French (FR), Japanese (JA), Korean (KO), Croatian (KR), Portuguese (PO), Russian (RU), Spanish (SP), Swedish (SW), and Turkish (TU). All the phone recognizers are trained and evaluated in the framework of the GlobalPhone project [105]. Phone recognition is performed with a Viterbi search using a fully connected null-grammar network of mono-phones; note that equal-probable language model is used in the decoding process, which means no prior knowledge about phone statistics is used. Figure 3.2 shows phone error rates per language in relation to the number of modeled phones

(in 8 languages). See [104] for further details.

After a "raw" phone stream was obtained from the phone recognizer, additional processing was performed to increase robustness. First, speech activity detection marks were used to eliminate phone segments where no speech was present. Second, silence labels of duration greater than 0.5 seconds were wrapped together as an end of an utterance. The idea in this case is to capture some information about how a speaker interacts with others, for example, does the speaker pause frequently, etc. Finally, extraneous silence was removed at the beginning and end of the resulting segments.

3.4 Language-dependent Speaker Phonetic Model

A Language-dependent Speaker Phonetic Model (LSPM) is generated using the n-grams modeling technique. The LSPMs used in this thesis are bi-gram models created using the CMU-Cambridge Statistical Language Modeling Toolkit (CMU-SLM) [19]. Unlike typical Gaussian Mixture Model-Universal Background Model (GMM-UBM) systems [94], the n-gram speaker phonetic models are not adapted from the universal background phonetic model, but instead are estimated directly from the speaker's available training data. Recent work also tried to train speaker phonetic models by adapting on the universal background model as described in [43]. In following sections, we use $LSPM_i^k$ to represent the phonetic model for speaker k in language i. Figure 3.3 shows the procedure of training LSPMs for speaker k. Each of the M phone recognizers ($PR_1, \cdots, PR_M$) decodes the training data of speaker k to produce M phonetic sequences. Based on these M phonetic sequences, M LSPMs are created for speaker k, one in each language.This procedure does not require transcription at any level. All the phone recognizers are open-loop recognizers. This means that during the decoding, the language model assigns same probabilities for all phones.

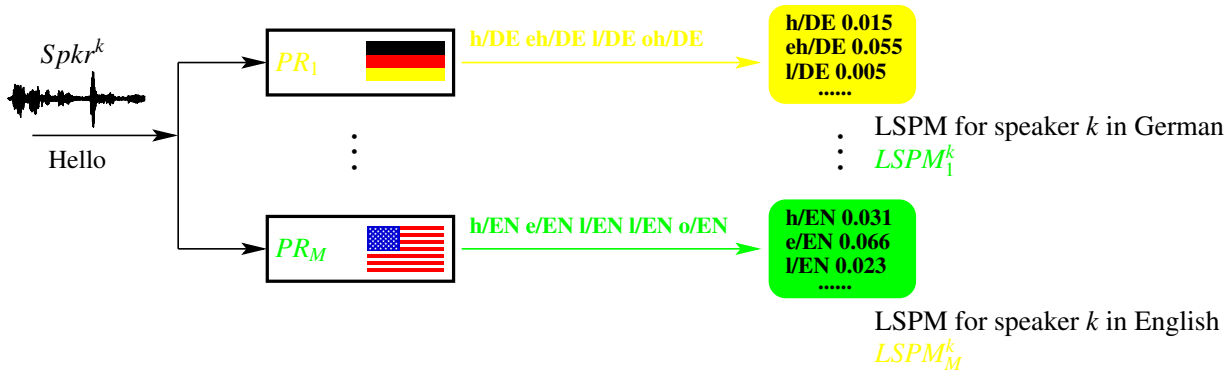

Figure 3.3: *Training Speaker Phonetic Model*

3.5 Phonetic Speaker Recognition in Time Dimension

The basic idea of phonetic speaker recognition is to identify a speaker via the statistical pronunciation model trained using phonetic sequences derived from that speaker's utterance. Although the phonetic sequences are produced using acoustic features, the identification decision is made based solely on the phonetic sequences. The assumption behind the phonetic approach is that phonetic sequences can cover a speaker's idiosyncratic pronunciation.

Generally, phonetic speaker identification in the time dimension using a single-language phone recognizer is performed in three steps: Firstly, the phone recognizer processes the test speech utterance to produce a test phone sequence. Secondly, the test phone sequence is compared to all previously trained LSPMs to compute decision scores. Finally, the speaker identity is decided based on the decision scores. Since the phone sequences are decoded from the speech which is a time series and the LSPMs are trained based on phone sequences along the temporal direction, we call it phonetic speaker identification in Time Dimension.

This process can be expanded to use multiple phone sequences from a parallel bank of phone recognizers trained on different languages. In this case, each phone stream is independently scored and the scores are fused together to form a single decision score.

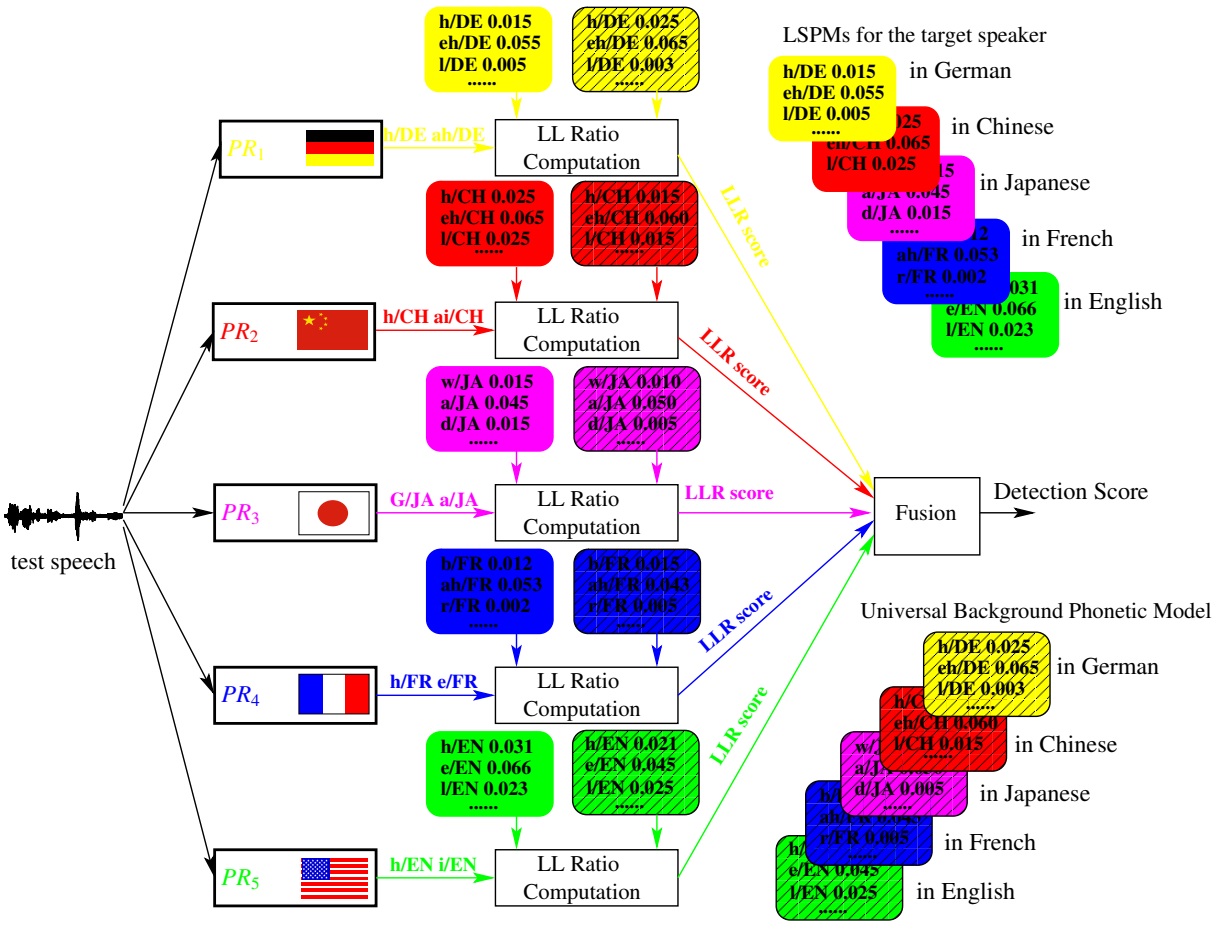

Figure 3.4: *Phonetic Speaker Detection in Time Dimension*

Figure 3.4 illustrates the work-flow of phonetic speaker detection/verification in time dimension. Generally, phonetic speaker detection in time dimension using a single-language phone recognizer is performed in three steps: Firstly, the phone recognizer decodes the test speech to produce a phonetic sequence. Secondly, the phonetic sequence is compared to the LSPM of the target speaker and a Universal Background Phonetic Model (UBPM) to compute likelihood scores. Finally, the Log of the Likelihood Ratio (LLR) is computed as the detection score. This process can be expanded to use multiple phone sequences decoded by a parallel bank of phone recognizers in different languages. In this case, each phone stream is independently scored and

the scores are combined together to form a single detection score.

Formula 3.1 defines the LLR detector for a phonetic speaker detection system with single language, where LL^k is the log likelihood score of the test sequence X against speaker k's phonetic model $LSPM^k$ and LL^U is the log likelihood score of the test sequence X against the universal background phonetic model $UBPM$. The detection score is the log of the ratio of these two likelihood scores.

$$Score^k = \log \frac{P(X|LSPM^k)}{P(X|UBPM)} = \log(P(X|LSPM^k)) - log(P(X|UBPM)) = LL^k - LL^U \quad (3.1)$$

For a multilingual phonetic speaker detection system, the scores from each of the languages are fused together such as:

$$Score^k = \sum_{i=1}^{M} w_i * Score_i^k$$

where i is used to index multiple languages, w_i is the fusion weight, and $Score_i^k = LL_i^k - LL_i^U$ is the detection score against speaker k in language i.

3.5.1 Database Description and Experimental Setup

The speaker detection experiments are conducted within the framework of the SuperSID project [114]. The text-independent speaker detection using the extended data task from the 2001 NIST Speaker Recognition Evaluation [86] was selected as our testbed. This task was introduced to allow exploration and development of techniques that can exploit significantly more training data than is traditionally used in NIST evaluations. The extended data task uses the complete Switchboard-I corpus of conversational telephone speech for training and test material. The corpus includes roughly 500 speakers and 2,500 conversations, each conversation involving a different speaker pairing. In order to use the full collection of speakers as target talkers, NIST defined an elaborate jack-knifed test design, splitting the corpus into 6 partitions (or splits). Speakers within each split are used as target or impostor talkers for that split, and speakers in the

other 5 splits may be used for training and normalization without fear of speaker contamination. Cycling through all 6 splits effectively uses the complete Switchboard-I corpus.

There are several different training conditions specified: using 1, 2, 4, 8, or 16 conversation sides for training the target talker models. All test segments use one entire conversation side. Since Switchboard conversations generally run about 5 minutes (for about 2.5 min of speech per conversation side) and sometimes as high as 10 minutes, this provides considerably more data than has been available in past evaluations, both for training and for testing, especially for the larger training conditions. Consequently, the Extended Data Task finally provided a testbed well-suited to the exploration of higher-level features such as word usage, speaker-characteristic expressions or events, and other features that can exploit significantly more training data than traditionally used in NIST evaluations. In following sections, we will use the 8-conversation training condition as our main reference point, though we will show performance across the range of training conditions for several of the experiments. We are most interested in performance when the systems have "sufficient" training material. We focus on 8- rather than 16-side training because relatively few speakers in Switchboard-I participated in 16 calls, so the speaker population is too small to provide robust statistics in that case.

We use the detection error trade-off (DET) [71] curve to plot the systems performance. The DET curve uses the false alarm and the miss probabilities as the x- and y-axes in normal deviate scale.

3.5.2 Phonetic Speaker Detection Results in the Time Dimension

Figure 3.5 shows the phonetic speaker detection performance in the time dimension for different training conditions (1, 2, 4, 8, or 16 training conversations). The Equal Error Rate (EER) is 8.4% for the 8-conversation training condition. The 8-conversation training condition is the most representative and statically significant condition in the extended task [93]. The compar-

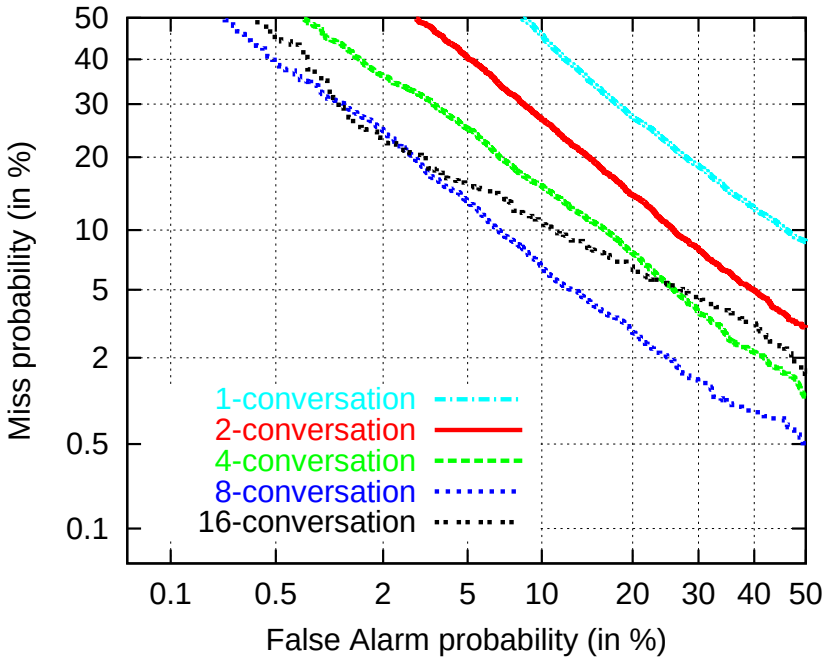

Figure 3.5: *Phonetic Speaker Detection in the Time Dimension*

ison of performance from different approaches will mainly focus on this training condition in following sections.

3.6 Phonetic Speaker Recognition in the Cross-Stream Dimension

The assumption behind the approach of phonetic speaker recognition in time dimension is that phonetic sequences can cover a speaker's idiosyncratic pronunciation by modeling the phonetic dependencies along the phone sequence in each language. For a speaker-specific pronunciation, ideally there should be some fixed set of phones from each of the multiple languages to rep-

resent it. For example, speaker *A* always pronounces "Hi" as "h ai", while speaker *B* likes to pronounce it as "h ei". Therefore, through phonetic dependencies captured by bigrams, the pronunciation idiosyncrasies will be distinguished between speaker *A* and *B*. In phonetic speaker recognition in time dimension, we use multiple phone recognizers to decode speech utterances and then model the phonetic dependencies in each language independently of other languages. However, if the phone recognizers decode the speaker-specific speech consistently, then there would be some fixed phones across multiple languages to represent speaker-specific pronunciation. Again, let's take the above example, speaker *A* always pronounces "Hi" as "h ai", while speaker *B* likes to pronounce it as "h ei". Then the English phone recognizer decodes speaker *A*'s "Hi" as "h/EN ai/EN", the Chinese phone recognizer decodes speaker *A*'s "Hi" as "h/CH ah/CH". While the English phone recognizer decodes speaker *B*'s "Hi" as "h/EN ei/EN", the Chinese phone recognizer decodes speaker *A*'s "Hi" as "h/CH eh/CH". Therefore, we can code speaker-dependent pronunciation dynamics across multiple-language phone sequences. For example, "ai/EN ah/CH" will represent speaker *A* while "ei/EN eh/CH" will represent speaker *B*. We call this approach "phonetic speaker recognition in cross-stream dimension." Similar to the time dimension, the LSPM and the UBPM are created in the cross-stream dimension and detection is done based on the log likelihood ratio. The detailed procedure of how we process phone sequences in multiple languages in phonetic speaker recognition in cross-stream dimension is described in the following subsections.

3.6.1 Cross-Stream Alignment

To discover the underlying dependencies of phones across multiple languages, we need first to align the multiple phone sequences. This alignment is done simply by aggregating all time boundaries from all phone sequences. As illustrated in Figure 3.6, the phones are duplicated to the their smallest unified time slots in each language in order to unify the boundaries across languages. According to the smallest time overlap across the three languages, the English

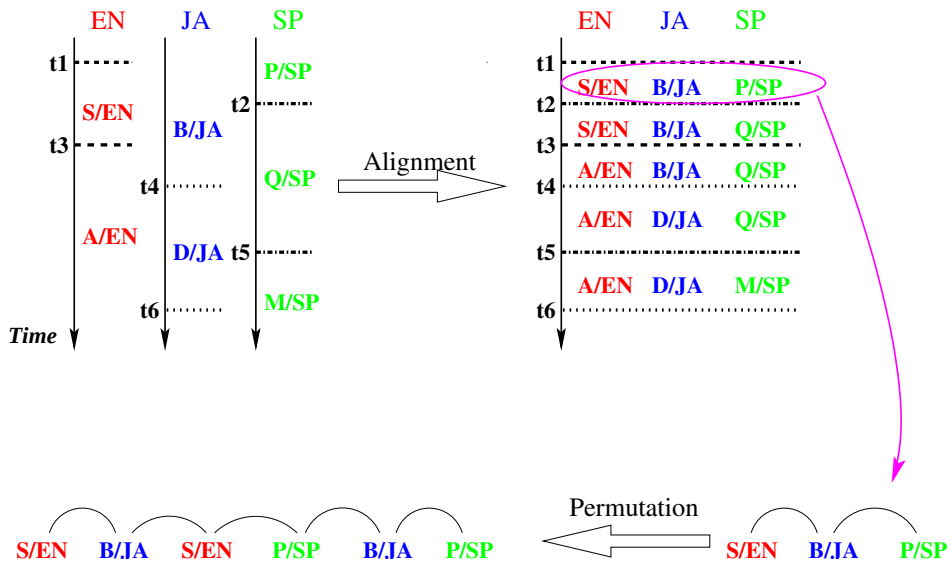

Figure 3.6: *Temporal Alignment and Permutation of Multiple Phone Sequences*

phone S/EN originally in the time slot $[t1,t3]$ is duplicated two times into time slots $[t1,t2]$ and $[t2,t3]$ and the Japanese phone B/JA originally in the time slot [t1, t4] is duplicated three times into time slots $[t1,t2]$, $[t2,t3]$ and $[t3,t4]$. Similarly, other phones are duplicated into their smallest time slots across the three languages.

3.6.2 Cross-Stream Permutation

A straight forward way to model the pronunciation dynamics in the cross-stream dimension is to model the statistical dependencies across streams. For this, as in the time dimension, we use n-grams by treating the aligned phones at each time slot as one input ”sentence” for the n-gram modeling. In the above example, we will have five ”sentences” to train the n-gram model: ”S/EN B/JA P/SP”, ”S/EN B/JA Q/SP”, ”A/EN B/JA Q/SP”, ”A/EN D/JA Q/SP”, and ”A/EN D/JA M/SP”. Since we want to model the bigram dependencies across all streams, it would be

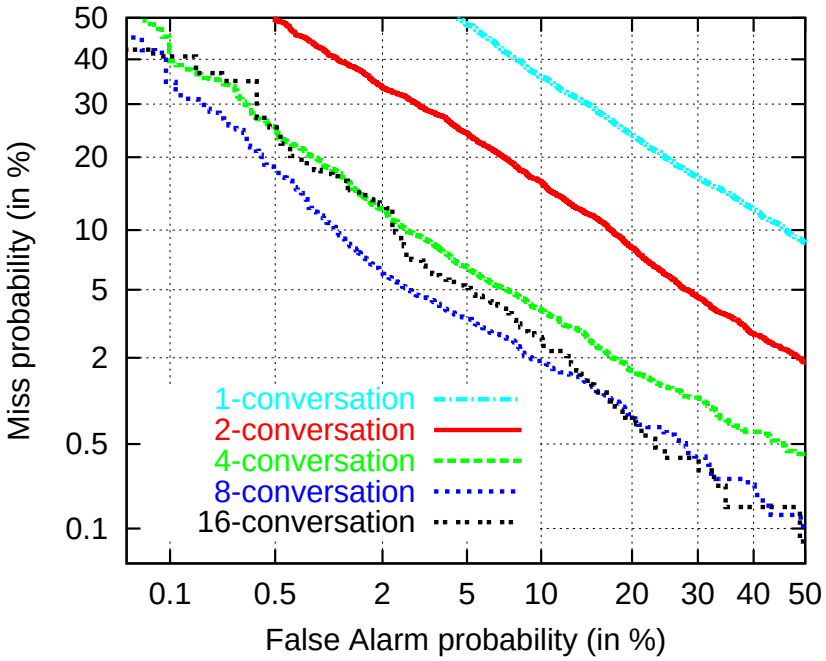

Figure 3.7: *Phonetic Speaker Detection in Cross-Stream Dimension*

better to model all possible pair dependencies. From the above alignment, however, bigrams can only model the dependencies of EN-JA pairs and of JA-SP pairs, but not of EN-SP pairs. Therefore, we simply permute the aligned phones at each time slot as shown in Figure 3.6, thus modeling all possible pairs from all languages at a given time. A bigram phonetic model is built for each speaker based on the aligned and permuted phone streams.

3.6.3 Phonetic Speaker Detection Results in Cross-Stream Dimension

Figure 3.7 shows the phonetic speaker detection performance in the cross-stream dimension with different training conditions. Figure 3.8 compares the performance in the cross-stream vs. time dimension under the 8-conversation training condition. In the cross-stream dimen-

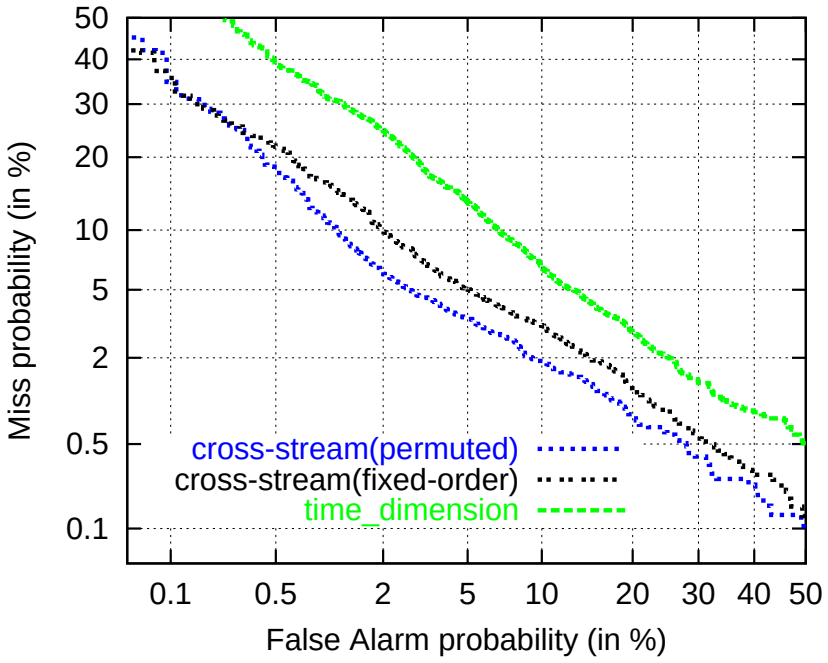

Figure 3.8: *Performance Comparison in Time Dimension vs. Cross-Stream Dimension*

sion, experimental results with and without permutation after alignment are shown. Under the 8-conversation training condition, the cross-stream system achieves 4.0% EER with permutation and 5.1% EER without permutation; both significantly outperformed the time dimension system, where the EER was 8.4%.

3.6.4 Combination of Time and Cross-Stream Dimensions

Modeling pronunciation dynamics in the cross-stream dimension is expected to carry complementary information to that in the time dimension and, hence, potentially can improve performance when combined. As mentioned in the relative work, Navratil [77] proposed maximum-likelihood binary-decision tree methods for phonetic speaker recognition in the time dimension,

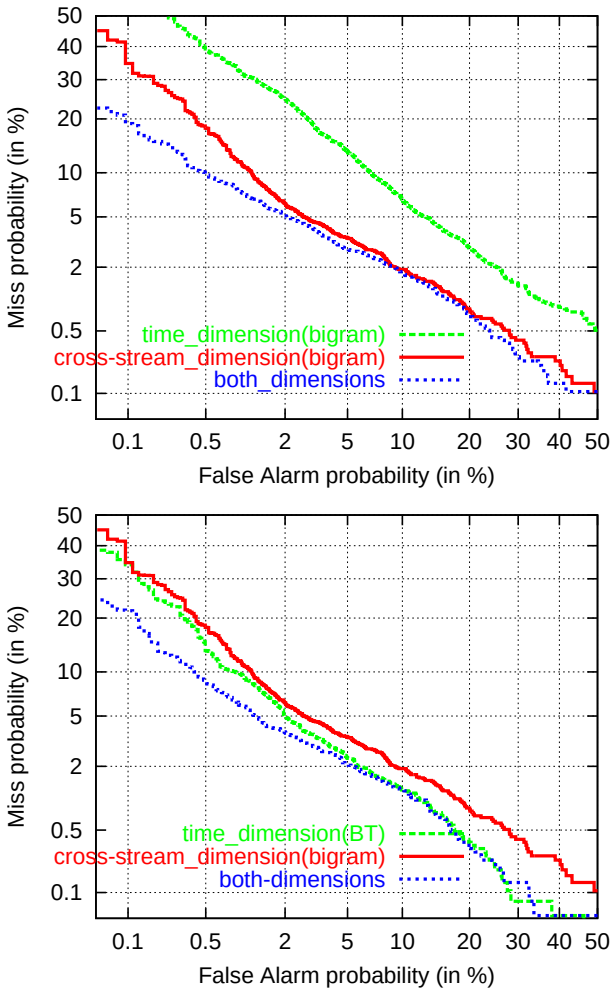

Figure 3.9: *Combination of Cross-Stream Dimension and Time Dimension: (upper) LSPMs are bigrams in both dimensions; (lower) LSPMs are bigrams in time dimension and binary trees in cross-stream dimension*

which aim to capture phonetic dependencies across longer time scales. We proposed n-gram based phonetic speaker recognition in the cross-stream dimension and we have the n-gram based system in the time dimension. Therefore, given the pallet of approaches outlined above, we next set out to examine fusion of the different dimensions of information to see if they

are indeed providing complementary information to improve speaker recognition accuracy. A simple linear combination with equal weights was used to fuse the detection scores from both systems. The upper part in Figure 3.9 shows the performance of combining both dimensions. Bigrams were used in both dimensions. The EER of the combination is reduced to 3.6%, compared to 8.4% in the time dimension alone and 4.0% in the cross-stream dimension alone. The lower part in figure 3.9 shows the performance of the system using the Binary decision Tree (BT) models in the time dimension alone [77], the performance of the system using bigrams in cross-stream dimension alone, and the performance of combining both systems under the 8-conversation training condition. The EER of the combination is further reduced to 3.0%, compared to 3.4% in the time dimension and 4.0% in the cross-stream dimension. Both experimental results indicate that the two dimensions do contain complementary information

3.7 PSR for Far-field Speaker Recognition

In this section, we apply phonetic speaker recognition approaches in time on the far-field speaker recognition task. We proposed two phonetic speaker identification approaches which we call LSPM-pp and LSPM-ds. These two approaches have the same phonetic language model training step as shown in 3.3. The difference between LSPM-pp and LSPM-ds is how the LSPMs of each speaker are applied during the identification.

3.7.1 LSPM-pp Speaker Identification

Figure 3.10 illustrates how the identification decision score (IDS) is computed for the test speech against one enrolled speaker. First, each of the M phone recognizers PR_i (in the figure, we use 5 phone recognizers as example), decodes the test speech and produces a phonetic sequence in each language. Secondly, each phonetic sequence is scored against the LSPM in the matched language for speaker k and the perplexity score PP_i^k is produced. Finally, the per-

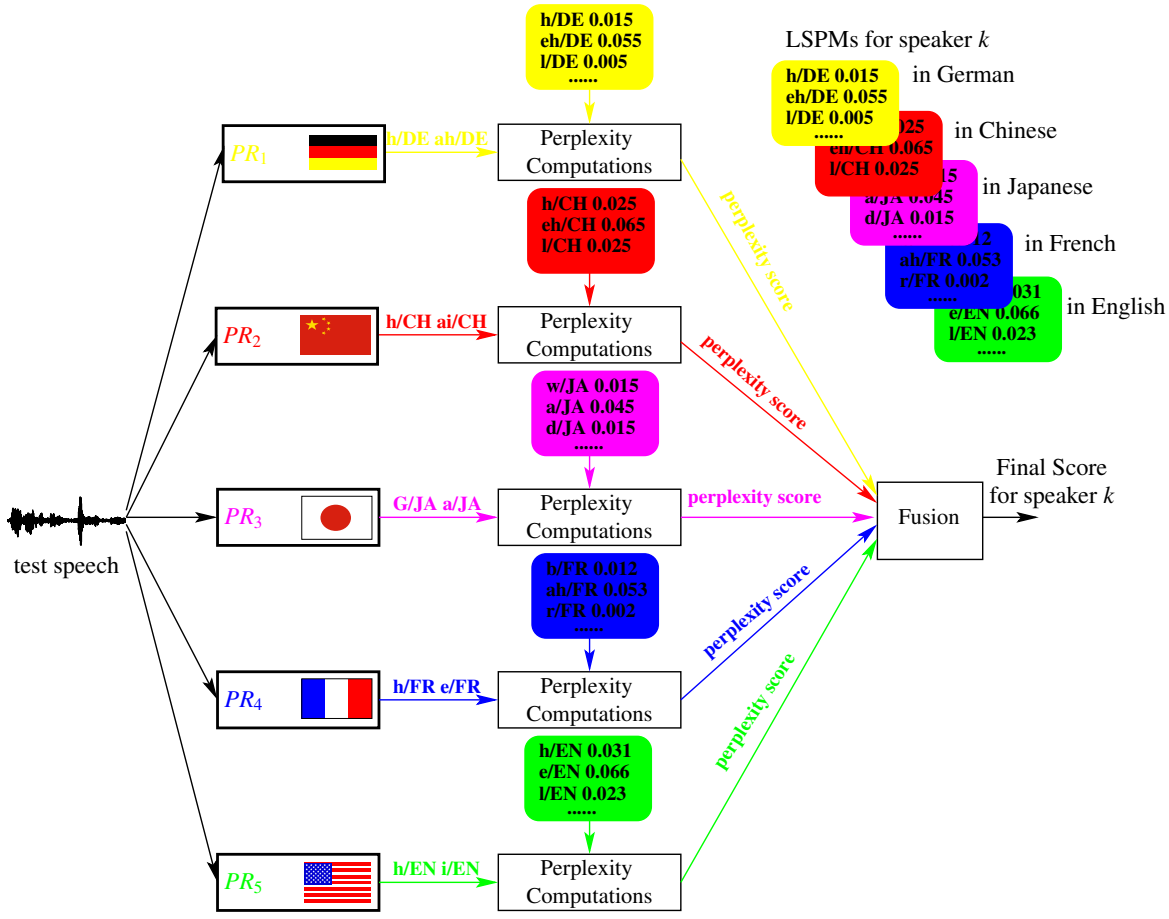

Figure 3.10: *Decision score computation against one enrolled speaker with LSPM-pp*

plexity scores from all the M languages are fused together as the final identification decision score IDS^k for speaker k.

$$IDS^k = \sum_{i=1}^{M} w_i * PP_i^k$$

where M is the total number of languages, PP_i^k is the perplexity score against speaker k in language i, and w_i is the fusion weight for each language. Our decision rule is to identify an unknown speaker as speaker s^* given by

$$s^* = \arg \min_{k=1}^{S} \{IDS^k\}$$

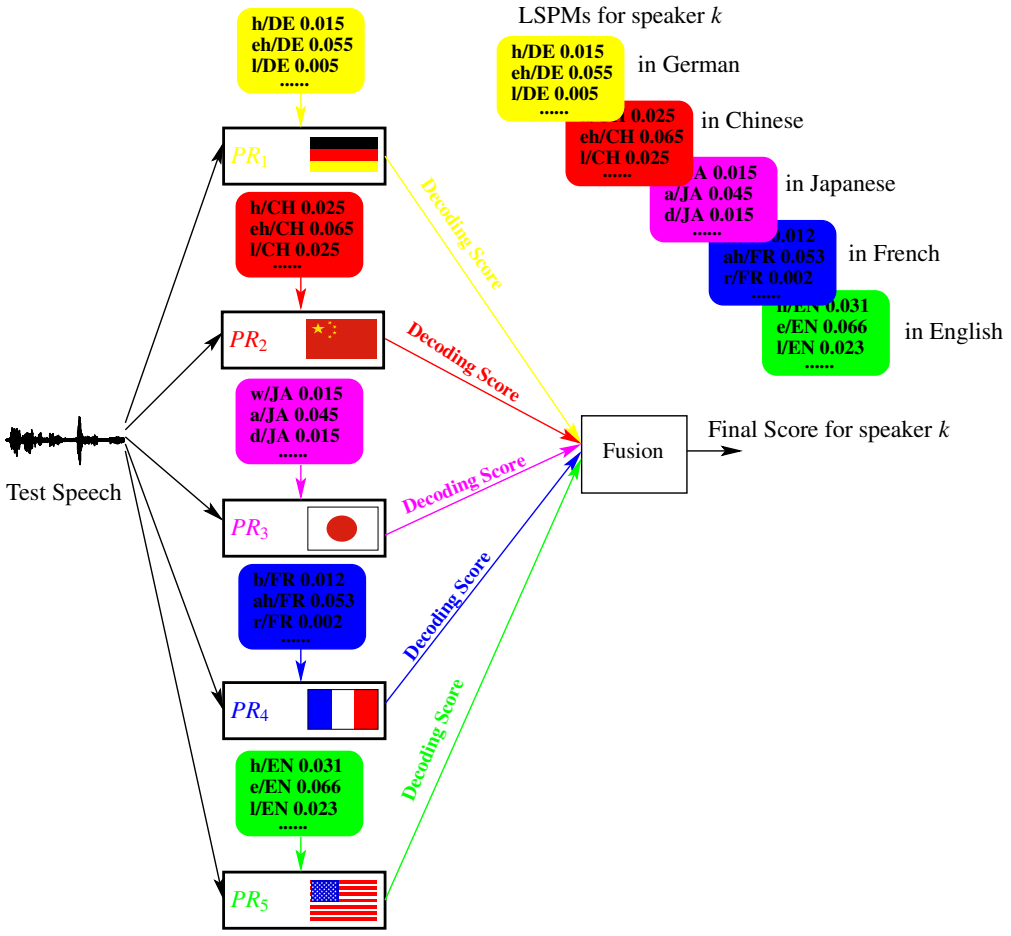

Figure 3.11: *Decision score computation against one enrolled speaker with LSPM-ds*

where S is the total number of enrolled speakers.

3.7.2 LSPM-ds Speaker Identification

In the LSPM-pp approach, the test speech is decoded by M phone recognizers (PR_i) using equal-probable language models. In contrast, for the LSPM-ds approach, we replace the equal-probable language model with the LSPMs in matched language. Figure 3.11 illustrates the procedure of computing the identification decision score for the test speech against one enrolled

speaker. The key idea of the LSPM-ds approach is to use the speaker-dependent LSPMs directly to decode the test speech. The underlying assumption is that a lower decoding distance score is produced on a matched speaker's LSPM than on a mismatched speaker's LSPM. The decoding score (*DS*) (including both acoustic and language model score) from all the M languages are fused together to form the final identification decision score IDS^k for speaker k.

$$IDS^k = \sum_{i=1}^{M} w_i * DS_i^k$$

where M is the total number of languages, DS_i^k is the decoding score against speaker k in language i, and w_i is the fusion weight for each language. Our decision rule is to identify an unknown speaker as speaker s^* given by

$$s^* = \arg\min_{k=1}^{S} \{IDS^k\}$$

where S is the total number of enrolled speakers.

As illustrated in the Figure 3.11, the test utterance is decoded M times against each of the enrolled speakers. Therefore, it leads to the disadvantage of the LSPM-ds approach in which the test utterance will be decoded $M * S$ times as opposed to M times for LSPM-pp approach. Furthermore, the success of this approach relies more on the ability to produce reliable speaker phonetic models from the training data.

3.7.3 Data Description and Experimental Setup

We test phonetic speaker recognition approaches in time dimension on the 2D Distant Micorphone Database. The reason we evaluate on this database is because that there is more data for each speaker in this database which is the requirement for the success of phonetic speaker recognition. This database, collected at ISL in 2000, contains 30 speakers in total. From each speaker five sessions had been recorded where the speaker sits at a table in an office environment, reading an article. The articles are different for each session. Each session is recorded

using eight microphones in parallel: one closed-talking microphone (Sennheizer headset), one Lapel microphone worn by the speaker, and six other Lapel microphones. The latter six are attached to microphone stands sitting on the table or beyond the table, at distances of 1 foot, 2 feet, 4 feet, 5 feet, 6 feet and 8 feet to the speaker, respectively. Tables and graphs shown in this chapter use "Dis0" to represent closed-talking microphone channel, "DisL" to represent speaker-wearing microphone channel, and "DisN" ($N > 0$) to refer to the n-feet distance microphone channel. We call this dataset "2D Distance Microphone Database". The data of the first four sessions, together 7 minutes of spoken speech (about 5000 phones) are used for training the LSPMs. Testing is carried out on the remaining fifth session adding up to one minute of spoken speech (about 1000 phones).

We first developed a speaker identification system using phonetic sequences from phone recognizers trained on multiple languages. We call this our multilingual system. This system uses phonetic sequences produced by context-independent phone recognizers from multiple languages instead of traditional short-term acoustic vectors [54], [103]. Since this information comes from complementary phone recognizers, we anticipate greater robustness. Furthermore, this approach is somewhat language independent since the recognizers are trained on data from different languages. We also developed a speaker identification system using phonetic sequences produced by single language phone recognizers trained on multiple conditions, which we call our multi-engine system. This system uses phonetic sequences produced by three different context-independent English phone recognizers. The system performance is measured using identification accuracy, which is the percentage of correctly recognized test trials over all test trials.

3.7.4 Multilingual LSPM-pp Speaker Identification Results

Table 3.1 shows the detailed language-dependent identification accuracy of LSPM-pp approach at different test utterance length under the matched condition, where both testing and training

Table 3.1: *Detailed performance in each language on Dis*0 *under matched condition (in %)*

Test Duration / Language	60s	40s	10s	5s
CH	100	100	56.7	40
DE	80	76.7	50	33.3
FR	70	56.7	46.7	16.7
JA	30	30	36.7	26.7
KR	40	33.3	30	26.7
PO	76.7	66.7	33.3	20
SP	70	56.7	30	20
TU	53.3	50	30	16.7
fusion of all languages	**96.7**	**96.7**	**96.7**	**93.3**

are recorded at distance *Dis*0. We can see from the table that with decreasing test duration, the performance based on single language gets very low, however this is overcame by fusing the multilingual information derived from all eight languages. After a fusion with equal weights of all languages the SID performance clearly outperforms the one on single language.

Table 3.2 compares the multilingual LSPM-pp identification results for all distances on different test durations under matched and mismatched conditions respectively. Under matched conditions, training and testing data are from the same distance. Under mismatched conditions, without knowing the test speech distance; we make use of all $D*M$ language-dependent and channel-dependent phonetic models ($LSPM^k_{i,d}$) for speaker k, where D is the total number of distant channels and M is total number of languages. In this case, the final identification decision score of the test speech against the $D*M$ LSPMs for speaker k is computed as:

$$IDS^k = \sum_{i=1}^{M} w_i * \min_{d=1}^{D} \{Score^k_{i,d}\}$$

Table 3.2: *LSPM-pp performance under matched and mismatched condition (in %)*

Test Length / Test Channel	Matched				Mismatched			
	60s	40s	10s	5s	60s	40s	10s	5s
Dist 0	96.7	96.7	96.7	93.3	96.7	96.7	96.7	90
Dist L	96.7	96.7	86.7	70.0	96.7	100	90.0	66.7
Dist 1	90.0	90.0	76.7	70.0	93.3	93.3	80.0	70.0
Dist 2	96.7	96.7	93.3	83.3	96.7	96.7	86.7	80.0
Dist 4	96.7	93.3	80.0	76.7	96.7	96.7	93.3	80.0
Dist 5	93.3	93.3	90.0	76.7	93.3	93.3	86.7	70.0
Dist 6	83.3	86.7	83.3	80.0	93.3	86.7	83.3	60.0
Dist 8	93.3	93.3	86.7	66.7	93.3	93.3	86.7	70.0

where $Score_{i,d}^{k}$ is the decision score in language i on the d distant channel. Here the *Score* will be *PP* in the LSPM-pp approach and *DS* in the LSPM-ds approach. The decision rule is as follows:

$$s^* = \arg\min_{k=1}^{S} \{IDS^k\}$$

where k is the index of enrolled speakers and S is the total number of enrolled speakers.

Figure 3.12 summarizes the average performance on different test durations under matched and mismatched conditions. We see that the performance under matched and mismatched conditions are comparable, with better performance under mismatched conditions when test duration is longer than 5 seconds.

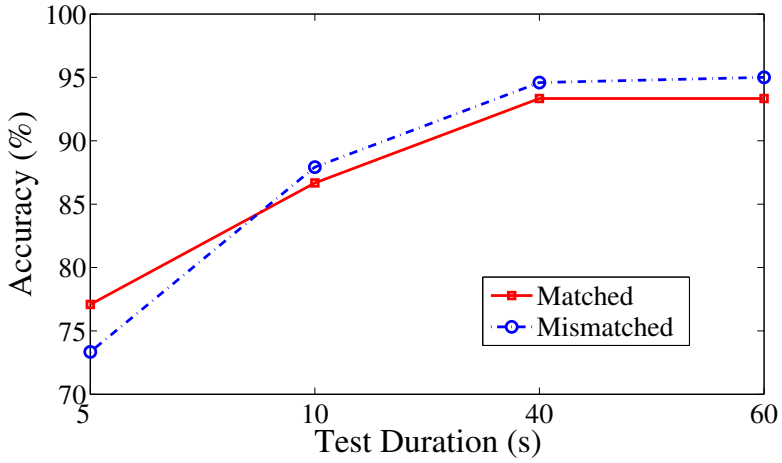

Figure 3.12: *Average SID performance under matched vs. mismatched conditions*

3.7.5 Comparison of LSPM-pp vs. LSPM-ds

Table 3.3 compares the performance of the LSPM-ds approach at Dis0 under matched conditions with that of LSPM-pp approach when test duration is 60-sec. Even though LSPM-ds is far more expensive than LSPM-pp, its performance (60%) is much worse than LSPM-pp (96.7%). The poor performance of LSPM-ds seems to support the assumption made earlier that the speaker phonetic models we produced, which perform well within the LSPM-pp framework, are not sufficiently reliable to be used during decoding as required by LSPM-ds. Therefore in the rest phonetic speaker identification experiments, LSPM-pp is the approach we applied.

In order to test the performance of the LSPM-ds approach when enough data is available for training a reliable LSPM, we conducted the following gender identification experiment. We used the NIST 1999 speaker recognition evaluation dataset [79] with a total of 309 female and 230 male speakers. For each speaker there are two minutes of training speech with each minute from one telephone channel type and one-minute test speech of unknown channel type. We group the training speech from speakers in same gender to train phonetic models of each

Table 3.3: *Performance Comparison of LSPM-pp and LSPM-ds on distant data (in %)*

Approach Language	LSPM-pp	LSPM-ds
CH	100	53.3
DE	80	40
FR	70	23.3
JA	30	26.7
KR	40	26.7
PO	76.7	30
SP	70	26.7
TU	53.3	26.7
Fusion of of all Languages	**96.7**	**60**

gender. We conducted gender identification using both the LSPM-pp and LSPM-ds approaches. We randomly choose 200 test trials containing 100 females and 100 males. The results in Table 3.4 indicate that given enough training data from which we can get a reliable speaker phonetic model, the LSPM-pp and LSPM-ds produce comparable results.

3.7.6 Multi-Engine LSPM-pp Speaker Identification Results

To investigate whether the reason for the success of the multilingual LSPM-pp approach is related to the fact that different languages contribute useful information or that it simply lies in the fact that different recognizers provide complementary information, we conducted the following set of experiments. We replaced the eight multilingual phone recognizers with three English phone recognizers which were trained on very different conditions, namely: Switchboard (telephony, highly conversational), Broadcast News (various channel conditions, planed

Table 3.4: *Performance Comparison of LSPM-pp and LSPM-ds on gender ID (in %)*

Approach Language	LSPM-pp	LSPM-ds
CH	88.5	89.5
DE	89.5	88.5
FR	89	91
JA	86.5	89
KR	87.5	88
PO	89	91.5
SP	92	92
TU	90	89
Fusion of all Languages	**94**	**94**

speech), and Verbmobil English (high quality, spontaneous). For a fair comparison between the three English engines and the eight multilingual engines, we generated all possible language triples out of the set of eight languages (56 triples) and calculated the average, minimum and maximum performance for each. Table 3.5 compares the results of the multilingual system to the multi-engine system. The results show that the best performance of the multilingual triples always outperforms the performance of the multi-engine triple. From these results we draw the conclusion that multiple English phone recognizers provide less useful information for the classification task than do multiple language phone recognizers. This is at least true for our given choice of multiple English engines in the context of speaker identification. The multiple languages have the additional benefit of being language independent. This results from the fact that the actual spoken language is not covered by the used multiple language phone recognizers. For example, in our experiments, the test language is English, which is not covered by the multilingual languages. The multi-engine system, which has the matched language "English" with

Table 3.5: *Performance comparison of LSPM-pp multilingual vs multi-engine (in %)*

System / Test Channel	Multilingual Avg (Min - Max)	Multi-Engine
Dist 0	87.92 (66.7 - 100)	93.3
Dist L	88.21 (63.3 - 96.7)	86.7
Dist 1	83.57 (66.7 - 93.3)	86.7
Dist 2	93.63 (86.7 - 96.7)	76.7
Dist 4	81.43 (56.7 - 96.7)	86.7
Dist 5	86.07 (66.7 - 96.7)	83.3
Dist 6	81.96 (66.7 - 93.3)	63.3
Dist 8	87.14 (63.3 - 93.3)	63.3

the test language, does not outperform the multi-lingual system. This indicates the potential of language independence.

3.7.7 Combination of Multilingual and Multi-Engine Systems

In order to investigate whether combining the multilingual system and the multi-engine system can provide more improvement for the speaker identification task, we conducted a second set of experiments. Table 3.6 compares the speaker identification performance of using the multilingual system alone with those of combining the multilingual system with the three multiple English phone recognizers. The combination is realized as adding more languages to the multiple languages. In Table 3.6, we use ML to represent the multilingual system and ME to represent the multi-engine system. SWB, BN and VE are used to represent single English phone recognizer trained on Switchboard, Broadcast News and Verbmobil English respectively. The results indicate that the interpolation of multilingual and multi-engine could not give any

Table 3.6: *Combination of Multilingual and Multi-Engine systems (in %)*

System / Test Channel	ML	ML+ME	ML+SWB	ML+BN	ML+VE
Dist 0	96.7	93.3	93.3	93.3	93.3
Dist L	96.7	96.7	96.7	93.3	96.7
Dist 1	93.3	90.0	90.0	90.0	90.0
Dist 2	96.7	96.7	96.7	96.7	96.7
Dist 4	96.7	93.3	93.3	93.3	93.3
Dist 5	93.3	93.3	93.3	93.3	93.3
Dist 6	93.3	80.0	80.0	83.3	83.3
Dist 8	93.3	90.0	90.0	93.3	93.3

further improvement. But we cannot conclude from these results that adding English language can not provide more complimentary information for speaker identification, since the three English phone recognizers are trained differently from those 8 language phone recognizers. To clarify this question, we further investigate the relationship between number of languages and identification performance as described in the following section.

3.7.8 Number of Languages vs. Identification Performance

In this set of experiments, we investigated the influence of the number of phone recognizers in different languages on speaker identification performance. These experiments were performed on an improved version of our phone recognizers in 12 languages trained on the GlobalPhone data. AR, KO, RU and SW are available in this version in addition to the 8 languages (CH, DE, FR, JA, KR, PO, SP, TU). Figure 3.13 plots the speaker identification rate over the number m of languages used in the identification process under matched conditions on 60-second test

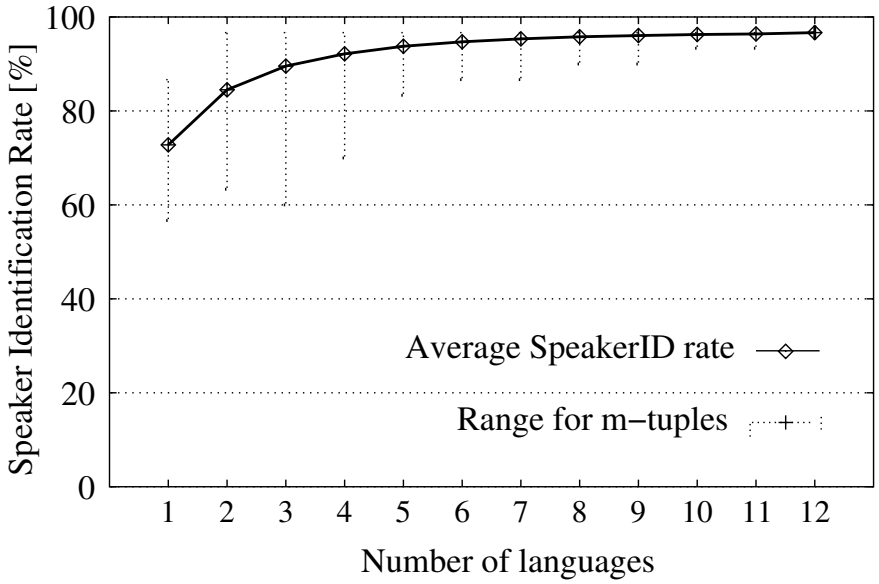

Figure 3.13: *Speaker Identification Performance vs. number of phone recognizers*

duration for all distant channels. The performance is given in average and range over the $\binom{12}{m}$ (m out of 12) language m-tuples. Figure 3.13 indicates that the average speaker identification rate increases with the number of involved phone recognizers. It also shows that the maximum performance of 96.7% can already be achieved using only two languages; in fact, among the total $\binom{12}{2} = 66$ language pairs, two pairs achieved best results: CH-KO, and CH-SP. However, the lack of a strategy for finding the best suitable language pair does not make this very helpful. On the other hand, the increasing average indicates that the probability of finding a suitable language-tuple that optimizes performance increases with the number of available languages. While only 4.5% of all 2-tuples achieved best performance, as many as 35% of all 4-tuples, 60% of all 6-tuples, 76% of all 8-tuples and 88% of all 10-tuples were likewise found to perform optimally in this sense.

3.8 Chapter Summary

Despite the qualities of the human speech process and richness of information in speech, most speaker recognition systems rely only on one source of information, acoustic features extracted from short segments of speech [95]. Other high-level information, such as particular word usage (idiolect), related to learned habits and style, are ignored by such systems. This exciting research area, speaker recognition using high-level information, pioneered by Doddington [28], has attracted a lot of research effort. Although we don't have a quantitative measurement of what level is more important among the different levels of information, we know that for automatic speaker recognition systems, it is not what you say but how you say it that is important. The particular content being conveyed is not as important as how the words sound (i.e. pronunciations).

In this chapter, we proposed a phonetic speaker recognition approach that aims at modeling the statistical pronunciation patterns based on the phonetic information from two "orthogonal" dimensions: time dimension and cross-stream dimension. The basic idea of phonetic speaker recognition is to identify a speaker via the statistical pronunciation model trained using phonetic sequences derived from that speaker's utterance. Although the phonetic sequences are produced using acoustic features, the identification decision is made based solely on the phonetic sequences. The assumption behind the phonetic approach is that phonetic sequences can cover a speaker's idiosyncratic pronunciation. Bigram modeling of the phone dependencies across tokenizers in multiple languages achieves 4% EER, a significant improvement over 8.4% EER in the time dimension on the NIST 2001 Speaker Recognition Evaluation Extended Data Task. A linear combination of systems in both dimensions at the score level reduces the EER to 3%, which indicates that the information captured in the cross-stream dimension is complementary to that in the time dimension. Also, the proposed approach works without the need for any lexical knowledge, which suggests its language independence.

Chapter 4

Speaker Segmentation and Clustering

4.1 Motivation

The rapid advance in speed and capacity of computers and networks have allowed the inclusion of audio as a data type in many modern computer applications. Multimedia databases or file systems can easily have thousands of audio recordings, including broadcasts, voice mails, meetings or other "spoken documents." However, an audio file is usually treated as an opaque collection of bytes with only the most primitive information tags: name, file format, sampling rate, etc. In order to make the audio data more accessible, new technologies that enable the efficient retrieval of desired information from speech archives are of increasing interest. Speaker segmentation and clustering consists of identifying who is speaking and when, in an audio stream. Ideally, a speaker segmentation and clustering system will discover how many people are involved in the audio stream, and output clusters corresponding to each speaker. Therefore, speaker segmentation and clustering technique can provide valuable inputs for automatic indexing of speech data. Speaker segmentation and clustering can also significantly improve speech recognition performance via enabling unsupervised adaptation on each cluster. In 2002, NIST started an evaluation paradigm, called Rich Transcription evaluation, which seeks to en-

rich speech-to-text (STT) transcription with Metadata Extraction (MDE). "Who Spoke When" speaker segmentation and clustering for English broadcast news and conversational telephony speech is one of the tasks in MDE. However, it is even more challenging to segment and cluster speakers involved in meetings. This is due to the occurance of speaking overlap and the use of distant microphones in meetings. Therefore, NIST initiated a similar evaluation on meetings in the spring of 2004 [96].

4.2 Background Knowledge

In this section, we briefly present some of the state-of-the-art statistical tools used in most speaker segmentation and clustering techniques, mainly to define notations and abbreviations.

4.2.1 Hypothesis Testing

Setting up and testing hypotheses is an essential part of statistical inference and is used in this thesis. In each problem considered, the question of interest is simplified into two competing claims/hypotheses between which we have to choose: the null hypothesis, denoted H_0, against the alternative hypothesis, denoted H_1. The null hypothesis H_0 represents a theory that has been put forward, either because it is believed to be true or because it is to be used as a basis for argument, but has not been proved. The alternative hypothesis H_1 is contrary to the null hypothesis

In the case of pattern recognition, the goal is to determine to which category or class a given observation (or sequence of observations) X belongs. If the probability density function (PDF) of each class is known, this becomes a problem in statistical hypothesis testing where H_0 and H_1 are defined as:

$$H_0\text{: } X \text{ is from class } C_1$$

and

$$H_1\text{: } X \text{ is from class } C_2$$

The optimum test according to the Bayes decision rule for minimum error to decide "not to reject" H_0 is a likelihood ratio test [29] given by:

$$\frac{p(X|H_0)}{p(X|H_1)} \geq \frac{P(H_1)}{P(H_0)} \tag{4.1}$$

where $p(X|H_i), i = 0, 1$, is the likelihood of the data X under the hypothesis H_i and $P(H_i)$ is the prior probability of hypothesis H_i. If we assume the prior probability of the two classes to be equal, the term on the right hand side of 4.1 is equal to one. The term of the left hand side of 4.1 is referred to as a likelihood ratio. Strictly speaking, the likelihood ratio test is only optimal when the likelihood functions are known exactly. In practice this is rarely the case and the likelihood ratio test usually involves a threshold value. A decision to "reject H_0 in favor of H_1" is made only if the likelihood ratio is less than threshold.

When the PDFs of two classes (hypotheses) are Gaussian densities or Gaussian mixture densities (which is always the case in this thesis), it is more convenient to compute and write the Log Likelihood Ratio (LLR) rather than writing the likelihood ratio itself. The LLR is computed by taking the logarithm of the likelihood ratio and now the decision rule becomes:

$$\log p(X|H_0) - \log p(X|H_1) \lessgtr threshold \tag{4.2}$$

In many situations, the problem is first formulated as a hypothesis testing problem and LLR is then used to make a decision. LLR is widely used in speaker recognition research [39] [94]. Sometimes, it is also referred to as Generalized Likelihood Ratio (GLR) [39]. LLR is used in this chapter calculate a similarity measure between two PDFs.

4.2.2 Model Selection

In many situations, we are forced to choose among nested classes of parametric models, e.g. models with different number of parameters. The Maximum Likelihood (ML) principle [30] (i.e. maximizing the likelihood of available training data) was developed only for a single parametric family, and hence it is not guaranteed to yield a sensible selection criterion in such situations. Schwarz [106] proposed a Bayesian approach to the model selection problem known as Bayesian Information Criterion (BIC). BIC is an approximation to the posterior distribution on model classes. While based on the assumption that proper priors have been assigned to each model, this approximation effectively eliminates any explicit dependence on prior choice. The resulting solution takes the form of a penalized log likelihood. The Bayesian Information Criterion states that the quality of model M to represent data $\{x_1,\ldots,x_N\}$ is given by:

$$BIC(M) = \log P(X|M) - \frac{\lambda}{2}V(M)\log N = \log P(x_1,\ldots,x_N|M) - \frac{\lambda}{2}V(M)\log N \qquad (4.3)$$

where $p(x_1,\ldots,x_N|M)$ denotes the maximum log likelihood of data $X = \{x_1,\cdots,x_N\}$ given model M, $V(M)$ denotes the number of free parameters in M and N denotes the number of observations in X. In theory λ should equal to 1, but it is a tunable parameter in practice. It was shown in [106] that maximizing the BIC value also results in maximizing the integrated likelihood which is the expected value of the likelihood over the set of parameters of M. A model having maximum BIC value is selected using this theory.

BIC was introduced for the case of speech and specifically for acoustic change detection and clustering by Chen and Gopalakrishnan in [17], where the problem was formulated as that of model selection. Since then, BIC has been used in many speech applications and is a state-of-the-art approach for acoustic change detection and clustering.

The problem of determining if there is a speaker change at point i in data $X = \{x_1,\ldots,x_N\}$ can be converted into a model selection problem. The two alternative models are: (1) model M_1 assumes that X is generated by a multi-Gaussian process, that is $\{x_1,\ldots,x_N\} \sim N(\mu,\Sigma)$, or

(2) model M_2 assumes that X is generated by two multi-Gaussian processes, that is

$$\{x_1,\ldots,x_i\} \sim N(\mu_1,\Sigma_1)$$
$$\{x_{i+1},\ldots,x_N\} \sim N(\mu_2,\Sigma_2)$$

The BIC values for the two models are

$$\begin{aligned} BIC(M_1) &= \log P(x_1,\ldots,x_N|\mu,\Sigma) - \frac{\lambda}{2}V(M_1)\log N \\ BIC(M_2) &= \log P(x_1,\ldots,x_i|\mu_1,\Sigma_1) + \log P(x_{i+1},\ldots,x_N|\mu_2,\Sigma_2) - \frac{\lambda}{2}V(M_2)\log N \end{aligned}$$

The difference between the two BIC values is

$$\begin{aligned} \Delta BIC &= BIC(M_1) - BIC(M_2) \\ &= \log \frac{P(x_1,\ldots,x_N|\mu,\Sigma)}{P(x_1,\ldots,x_i|\mu_1,\Sigma_1)P(x_{i+1},\ldots,x_N|\mu_2,\Sigma_2)} + \frac{\lambda}{2}[V(M_2)-V(M_1)]\log N \end{aligned}$$

A negative value of ΔBIC means that model M_2 provides a better fit to the data, that is there is a speaker change at point i. Therefore, we continue merging segments until the value of ΔBIC for the two closest segments (candidates for merging) is negative.

It is also interesting to note that BIC formally coincides with other information theoretic criteria like Minimum Description Length (MDL) [97] and Akaike Information Criterion (AIC) [2]. These information theoretic measures have a completely different motivation and derivation to BIC. The motivation for MDL, for example, is to select a model that provides the shortest description of the data, where describing data can be regarded as coding. The term depending on the number of free parameters in BIC (right hand side of 4.3) is explained in the MDL framework as the extra cost incurred by transmitting the parameters of the model. Rissanen [97] demonstrates that for a regular parametric family of dimension d, this amounts to transmitting at least $\frac{d}{2}\log N$ bits, where N is the length of the data.

4.2.3 Performance Measurement

A good speaker segmentation algorithm should provide only the correct speaker changes. So each segment should contain exactly one speaker. There are two types of errors related to speaker change detection: insertion errors (when a speaker change is detected but it does not exist in reference) and deletion errors (an existing speaker change is not detected). These two types of errors have a different impact depending upon the application. In our system, the segmentation stage is followed by a clustering stage. Therefore, insertion errors (resulting in over segmentation) are less critical than deletion errors, since the clustering procedure has the opportunity to correct the insertion errors by grouping the segments related to the same speaker. On the other hand, deletion errors cannot be corrected in the clustering stage.

For evaluation, the reference was generated from a manual transcription. However, the exact speaker change point is not very accurate in the reference, since the perception of speaker change is very subjective. Therefore, we define an accuracy window around the reference speaker change point; following [118], it is set to one second. For example, if N_r and N_h are sample indexes of reference and hypothesized speaker change points respectively. We call the hypothesis N_h a hit if

- N_h is the hypothesized change point closest to N_r, and
- N_r is the reference change point closest to N_h, and
- the distance between N_r and N_h is less than one second.

From the alignment between reference and hypothesis, we can determine the precision (percentage of correct hypothesized speaker change points among all the hypothesized change points) and recall (percentage of correct hypothesized speaker change points among all the true change points). Deletion errors will directly lower the recall. Insertion errors will reduce the precision. Generally we seek systems that exhibit both high recall and high precision. However, as men-

tioned before, insertion errors can be overcome by following clustering procedure, therefore deletion errors are more critical than insertion errors; we are more concerned about the recall value.

Speaker diarization error is the standard measurement of the overall performance for speaker segmentation and clustering used in the NIST evaluations [115]. The overall speaker segmentation and clustering performance can be expressed in terms of the miss rate (speaker in reference but not in system hypothesis), false alarm rate (speaker in system hypothesis but not in reference), and speaker error rate (mapped reference speaker is not the same as the hypothesized speaker). The speaker diarization score is the sum of these three components and can be calculated using

$$DiaErr = \frac{\sum_{allS}\{dur(S) * (max(N_{ref}(S), N_{sys}(S)) - N_{correct}(S))\}}{\sum_{allS}\{dur(S) * N_{ref}(S)\}} \tag{4.4}$$

where *DiaErr* is the overall speaker diarization error, $dur(S)$ is the duration of the segment, $N_{ref}(S)$ is number of reference speakers in the segment, $N_{sys}(S)$ is the number of system speakers in the segment, and $N_{correct}(S)$ is the number of reference speakers in the segment which are also hypothesized by the system. This formula allows the entire audio to be evaluated, including regions of overlapping speech. In tables in this chapter, we use abbreviations "Miss", "False Alarm", "Spkr Err", and "Diarization Err" to represent miss rate, false alarm rate, speaker error rate, and diarization error rate, respectively.

4.3 Related Work

This section presents a literature review of most of the significant work that addressed the issues related to speaker segmentation and clustering.

4.3.1 Speaker Segmentation

Speaker segmentation is also called ”speaker change detection” in the literature. Although speaker change detection also belongs to the family of pattern classification problems, and thus has a feature extraction module followed by classification/segmentation framework, no significant work has been reported on the feature extraction module. Traditionally, MFCC features are extracted every 10ms and fed to the segmentation algorithm. Various segmentation algorithms have been proposed in the literature, which can be categorized as follows:

- **Decoder-guided segmentation:** The input stream is first decoded; then the desired segments are produced by cutting the input at the silence locations generated from the decoder ([61] [124]). Other information from the decoder, such as gender information could also be utilized in the segmentation.

- **Model-based segmentation:** This involves making different models e.g. GMMs, for a fixed set of acoustic classes, such as telephone speech, pure music, etc. from a training corpus (e.g. [7] [57]); the incoming audio stream is classified by ML selection over a sliding window; segmentation is made at the locations where there is a change in the acoustic class.

- **Metric-based segmentation:** A distance-like metric is calculated between two neighboring windows placed at each sample; metrics such as Kullback-Liebler (KL) divergence ([108]), LLR ([10] [24] [23] [76]) or BIC ([16] [24] [23] [76] [68] [128] [117] [118]) can be used. The local maxima or minima of these metrics are considered to be the change points.

All of these methods have limitations. The decoder guided segmentation only places boundaries at silence locations, which in general has no connection with the acoustic changes in the data. The model based segmentation approaches may not generalize to unseen data conditions

as the models are no longer compatible with the new data conditions. The metric based approaches generally require a threshold/penalty term to make decisions. These thresholds are set empirically and require additional development data.

4.3.2 Speaker Clustering

Most of the state-of-the-art solutions to the speaker clustering problem use a bottom-up hierarchical clustering approach i.e. starting from a large number of segments (clusters), some of the clusters are sequentially merged following a "suitable" strategy. This strategy mostly consists of computing a distance metric between any two clusters and then merging the two clusters with the smallest distance. Since most of the popular distance metrics are monotonic functions of the number of clusters, an external method of controlling the number of clusters (or merging process) is a necessary part of the problem. Thus, most of the previous work on this topic revolves around employing a suitable distance metric and corresponding stopping criterion.

One of the earliest pieces of work on speaker clustering from the point of view of speaker adaptation in ASR systems was proposed in [51]. In this work, the Gish-distance proposed in [40] was used as a distance metric, which is based on Gaussian models of the acoustic segments. Hierarchical clustering was performed based on this distance metric through selecting the best clustering solution automatically by minimizing the within-cluster dispersion with some penalty against too many clusters. The penalty term is needed because the within-cluster dispersion will keep monotonically decreasing, which will lead to the unwanted clustering of one segment per cluster. Although, a study of the number of clusters obtained with different penalty terms was done, no systematic way was proposed to deduce the optimal value of this term. It was also shown in this work that the automatic speaker clustering contributed significantly to reduction of Word Error Rate (WER).

Siegler et al. in [108] used Kullback-Leibler (KL) divergence (relative cross entropy) as

the distance metric for speaker clustering. The KL distance between the distributions of two random variables A and B is an information theoretic measure equal to the additional bit rate accrued by encoding random variable B with a code that was designed for optimal encoding of A [21]. In an agglomerative clustering approach, the KL distance was also compared with the Mahalanobis distance. In this framework, an utterance was clustered with an existing cluster if it was within a threshold distance, otherwise it was used as the seed of a new cluster. Different threshold values were tried in this work but no systematic way of finding this threshold was proposed.

Solomonoff et al. in [109] used LLR and KL distance metrics for speaker clustering. An agglomerative dendogram clustering framework was proposed in this work. Thus, a closest pair of clusters was picked using these distance matrices and merged until there was only one cluster. In order to obtain the appropriate number of clusters, dendogram cutting was used. This was done using "cluster purity" as a measure, which was also proposed in this work.

A top-down split-and-merge clustering framework was proposed in [55] [56]. This work was based upon the idea that the output of this clustering was to be used for Maximum Likelihood Linear Regression (MLLR) adaptation, and so a natural evaluation metric for clustering is the increase in the data likelihood from adaptation. The distance metrics used for splitting and merging were Arithmetic Harmonic Sphericity (AHS) [8] and Gaussian Divergence. The clustering scheme works top-down with each node being split into up to four child nodes at each stage. The splitting was done until no more segments could be moved or the maximum number of iterations was reached. At each stage of splitting, some of the nodes were merged. The merging criterion was based on simple distance from the center of the node to its segments. The decisions were made by comparing this distance against a threshold value. It was also found to be necessary to define when a split is allowable to prevent data being split back into its constituent segments. Thus, a heuristic based on minimum occupancy count was used to ensure robust speaker adaptation. This clustering algorithm was applied to the HTK broadcast

news transcription system [124] [42].

The most commonly used distance metric for speaker clustering is BIC [16]. In this work, starting from each segment (hand segmented) as a cluster, hierarchical bottom-up clustering was performed by calculating the BIC measure for every pair of clusters and merging the two clusters with the highest BIC measure (see equation 4.3). The clustering was stopped when the merging of two clusters resulted into no increase in BIC measure. Although, in this work the authors used a theoretically motivated penalty value, subsequent work on speaker clustering using BIC ([117] [118] [64]) found that adjusting this penalty on the training data not only produces better results, but is also necessary for the system to run on unseen data.

Recently, in the framework of the DARPA-EARS program, NIST started the speaker diarization evaluation. This task is very similar to speaker clustering, except that the motivation is a little different. Speaker diarization is intended to make the transcription of ASR systems richer by determining who said what. Whereas, the motivation of speaker clustering work for speech recognition is to create speaker clusters from the point of view of speaker adaptation. Thus, in the later case, it is possible to make a single cluster for two speakers if the two individual clusters do not have enough data and the two speakers are acoustically very similar. Most of the approaches presented in this evaluation used BIC as the distance metric ([78] [3] [74]). Another penalized LLR distance metric was proposed by Gauvain and Barras [37], where the penalty parameters were tuned to get the best performance. The Cambridge diarization system [116] used the framework defined in [55]. It is clear that BIC is the state-of-the-art approach toward speaker clustering.

4.4 Speaker Segmentation and Clustering Scenarios

There are mainly three scenarios in speaker segmentation and clustering evaluations, Broadcast News (BN), Conversational Telephone Speech (CTS), and Meetings (MT). These three types

of scenarios show different characteristics in terms of number of speakers and number of turn changes. Generally, there are a large number of speakers in broadcast news shows (average 14), fewer number of speakers in meetings (average 6), and usually only two speakers in conversational telephone speech. The speaker's talking style is different for these three scenarios. The talking style is very spontaneous for conversational telephone speech and meetings, while for broadcast news the talking style is relative formal and similar to read speech. The different talking style results in different number of turn changes for different scenarios. As shown in Figure 4.1, conversational telephone speech and meetings have much faster speaker turn changes. The fast turn change is particularly crucial as reported in [24] [98]. The very spontaneous speaker's talking style also results in the presence of many short and non-verbal sounds (e.g. huh, laughter etc.), and the existence of cross talking (speech conveys multiple speakers speaking at the same time). The factors including fast speaker turn changes, spontaneous talking style, and relative large amount of speakers make the speaker segmentation and clustering in meetings the most challenge task. In the rest of this chapter, we present our speaker segmentation and clustering systems for conversational telephone speech and meetings.

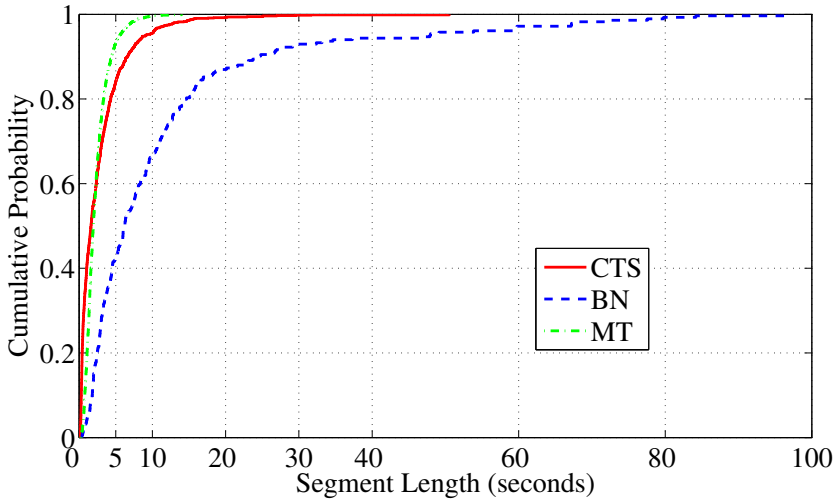

Figure 4.1: *CDF of Speaker Segment Length*

Figure 4.1 compares the cumulative density function (CDF) of segment length for Broadcast News (BN), conversational telephony speech (CTS) and Meetings (MT). We can see that the speaker segment durations are much shorter in conversations and meetings than in broadcast news, which also indicates that speaker-changing rate is much higher in conversations and meetings. The median and mean segment lengths measured in seconds are 1.63 and 2.80 in conversations, 2.36 and 1.93 in meetings, while in broadcast news they are 6.19 and 11.27.

4.5 Speaker Segmentation and Clustering on CTS

This section presents our speaker segmentation and clustering system on the conversation telephone speech, which is one of the evaluation tasks in the NIST 2003 Spring Rich Transcription evaluation (RT-03S).

4.5.1 Data Description

The English Conversational Telephone Speech (CTS) set was used for the NIST RT-03S evaluation [115]. The CTS set was selected from the Switchboard database, a corpus of spontaneous telephony conversations. It is a standard publicly available database that is suitable for speaker identification, verification and segmentation evaluations. Each call is about 5-6 minutes in duration and contains English conversations between two speakers via landline or cellular telephone. The participants speak in a spontaneous and unscripted way, with frequent pauses and spontaneous effects such as non-verbal sounds. Originally each side (speaker) in a call has a separate channel, which is the condition used in RT-03S evaluation. We also mix the two channels to form a single channel two-speaker conversation. We used both the dry run and evaluation test set from RT-03S evaluation. The dry run test set contains 12 calls and the evaluation test set contains 36 calls, about 4 hours total. The whole data set is well diversified in speaker genders and telephony networks (landline and cellular).

4.5.2 System Overview

For the CTS data, it is very unusual to find more than one speaker on the same conversational side. When the data for the channels is provided separately, the speaker segmentation and clustering task reduces to a speech activity detection problem, that is to detect whether the (single) speaker is talking or not.

We use a hybrid segmentation approach, which consists of four steps. This approach utilizes the advantages of metric-based and model-based approaches. It does not need any prior training. Therefore it has the potential of portability across different do-mains. The four steps in the implementation of this algorithm are:

- Initial segmentation
- GMMs generation
- Segmentation with GMMs
- Resegmentation with Tied GMM

Operations of each step are described in detail in the following subsections.

Initial Segmentation

In this step, each frame of the audio stream is first classified into one of the three classes: highly confident speech, highly confident non-speech, unsure. The frame window size is 30ms (240 samples) and it shifts every 10ms along the audio stream. The classification decision is based on the features of energy, zero-crossing rate and FFT magnitude variance. The feature of FFT magnitude variance is chosen based on the statistics that a speech frame has higher FFT magnitude variance than a non-speech frame. The goal of the initial segmentation is to

gather as many highly accurate speech and non-speech segments as possible to bootstrap the generation of GMM models for speech and non-speech events.

GMMs Generation

Based on the highly confident speech and non-speech segments found in the initial segmentation step, we build two GMMs, θ_{sp} and θ_{nsp} , for speech and non-speech respectively. Feature vectors use 13-dimensional Mel-frequency cepstral coefficients extracted with the same frame and window size as in the initial segmentation. 32 Gaussian mixture components and diagonal variance matrices are used.

Segmentation with GMMs

After building the GMMs (θ_{sp} and θ_{nsp}) for speech and non-speech, we use these models to classify the unsure parts in the audio produced by the initial segmentation. A three-way classification is done according to formula 4.5. θ_{sp} and θ_{nsp} are updated after each classification step by adding the new speech and non-speech data. The classification step iterates with the model-updating step (in our experiments, these two steps usually iterate 3-5 times) until all the unsure parts are labeled as either speech or non-speech. The constant TH in formula 4.5 is set to 0.5 in our experiments. Our goal in this step is to produce as pure speech segments as possible while not missing too many short speech segments at the same time.

$$\begin{array}{llll} P(x|\theta_{sp}) > P(x|\theta_{nsp}) & and & \dfrac{P(x|\theta_{sp}) - P(x|\theta_{nsp})}{P(x|\theta_{nsp})} > TH & speech \\ P(x|\theta_{nsp}) > P(x|\theta_{sp}) & and & \dfrac{P(x|\theta_{nsp}) - P(x|\theta_{sp})}{P(x|\theta_{sp})} > TH & nonspeech \\ & & otherwise & unsure \end{array} \tag{4.5}$$

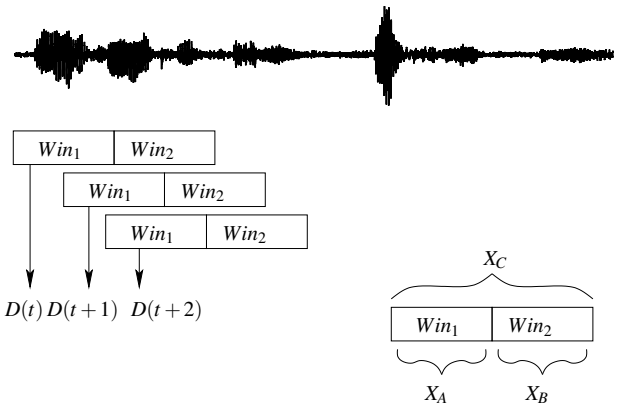

Figure 4.2: *Speaker Change Detection*

Resegmentation with Tied GMMs

If there is a clear pause between a speaker turn-change, it will be detected by the previous three steps via non-speech segment. However, if no pause occurs between a speaker turn-change, the previous three steps will fail to detect it. Therefore, in this step, we aim at detecting those seamless speaker turn-changes by the metric-based BIC segmentation. We suspect such speaker turn-changes only happen in long segments. So for segments obtained from the previous three steps that are longer than a certain length (5 seconds is used in our implementation since the majority of segments is shorter than 5 seconds as shown in Figure 4.1), we use the metric-based BIC segmentation approach to find out whether there still exist speaker turns. This may cause over segmentation, but the following clustering procedure can recover from this by merging homogeneous segments. However, if a speaker turn is missed, which means that a segment contains speech from more than one speaker, it can never be recovered by clustering. The procedure is shown in Figure 4.2.

We first compute the distance between two neighboring windows. The window duration is one second and windows are shifted by 10ms. The distance between Win_1 and Win_2 is defined

as

$$D(Win_1, Win_2) = -\log \frac{P(X_C|\theta_C)}{P(X_A|\theta_A)\, P(X_B|\theta_B)} \tag{4.6}$$

where X_A, X_B, and X_C are feature vectors in Win_1, in Win_2, and in the concatenation of Win_1 and Win_2, respectively. θ_A, θ_B, and θ_C are GMM models built on X_A, X_B, and X_C, respectively. We can see from (4.6) that the larger the distance, the more likely a speaker turn change exists at the boundary between Win_1 and Win_2.

We assume a speaker turn change exists if the local maximum of distances satisfies

$$\begin{aligned} D_{max} - D^L_{min} &> \alpha \\ D_{max} - D^R_{min} &> \alpha \\ \min\{(I_{max} - I^L_{min})(I^R_{min} - I_{max})\} &> \beta \end{aligned} \tag{4.7}$$

where D_{max} refers to the local maximum distance value and D^L_{min} and D^R_{min} refer to the left and right local minimum distance values around the local maximum. I_{max} refers to the index of the local minimum. The third inequality in (4.7) considers not only the value of the local maximum but also its shape. α and β are constant thresholds, for which we found optimal values via cross-validation on the development set. α is equal to the variance of all the distance values times a factor of 0.5. β is set to 5. Our approach differs from other approaches, such as [17][24], because in our implementation we build a Tied GMM (TGMM) using all speech segments and generate a GMM for each segment by adapting the TGMM. The advantage is that a more reliable model can be estimated with a TGMM.

Speaker Clustering

Originally each side (speaker) in a telephone conversation has a separate channel. On the separate channel, no speaker clustering is needed since only one speaker talks on this channel most of the time. We also mix the two channels to form a single mixed channel two-speaker conversation. In this case, speaker clustering is required. For speaker clustering, we use a hierarchical,

agglomerative clustering technique called TGMM-GLR. We first train a TGMM, θ, based on all speech segments. Adapting θ to each segment generates a GMM for that segment. The definition of the GLR distance between two segments is the same as in (4.6). A symmetric distance matrix is built by computing the pairwise distances between all segments. At each clustering step, the two segments which have the smallest distance are merged, and the distance matrix is updated. We use the Bayesian Information Criterion as a stopping criterion.

After clustering we merge any two segments that belong to the same speaker and have less than a 0.3 second gap between them.

4.5.3 Experimental Results

We use "Purity" to measure the performance of speech and non-speech classification in the initial segmentation and model-based resegmentation. Purity of a speech segment is defined as the percentage of the frames in the segment that are true speech frames. Purity of non-speech is defined similarly.

We use the standard diarization error to measure the overall speaker segmentation and clustering performance. For the CTS data, since the data for the channels is provided separately, Therefore the performance under the separate channel condition can be explained by a simple graphical representation to allow a quick visual inspection of the entire side in question, which can not be provided by the numerical diarization scores. Figure 4.3 gives an example of such graphical representation. For the example illustrated in Figure 4.3, the two hypotheses get the same overall diarization errors, but the graph clearly shows the differences between the systems.

In initial segmentation, 42% of the entire conversation stream is labeled as unsure. Table 4.1 shows the speech and non-speech purity after the initial segmentation step and GMM-based resegmentation step. The results show that initial segmentation is very effective and has the po-

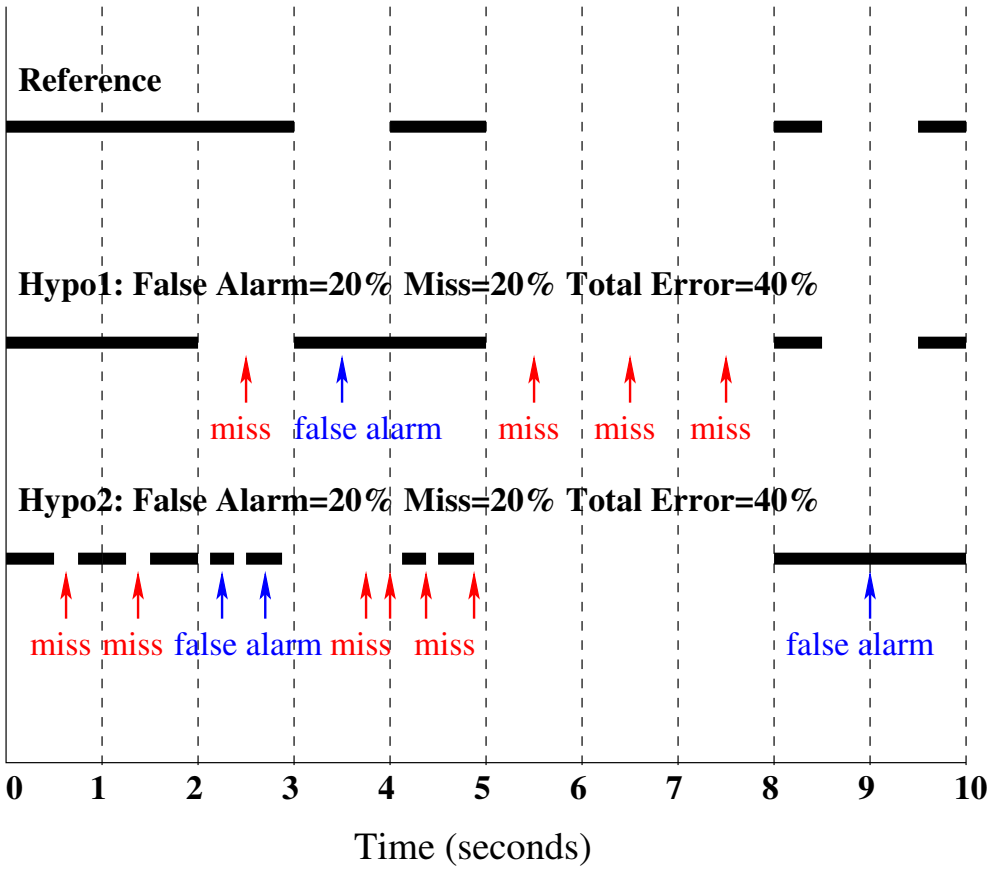

Figure 4.3: *Graphical representation of system performance for the separate channel CTS diarization problem*

tential of domain independence since it does not need any prior domain-dependent information. Table 4.2 shows the decomposed diarization errors on separate channels and single mixed channel respectively. The single mixed channel task is more challenging than the separate channel task, since there are more severe cross talkings and seamless speaker turn changes. While in the separate channel task, it is always true that each channel contains only one speaker. Therefore, speaker diarization task on separate channels is simplified as a speech activity detection task. In our current implementation, we did not specifically deal with cross talkings, just label them as one speaker by clustering.

Table 4.1: *Purity performance on dry run set*

Purity	Initial segmentation (58% of the data)	Segmentation with GMM (100% of the data)
Speech	99.5%	95.3%
Non-Speech	97.2%	95.5%

Table 4.2: *Diarization error on separate vs. single mixed channel on dry set*

Diarization Err		Miss	False Alarm	Spkr Err
separate channel	Dry run	4.5%	4.7%	0.0%
	Evaluation	6.5%	4.0%	0.0%
Single channel	Dry run	10.8%	7.0%	0.9%
	Evaluation	15.5%	8.4%	1.4%

We also found that there is no significant difference in the performance for seperate channel speech activity detection on landline telephony speech and cellular telephony speech, as shown in table 4.3, which indicates that the techniques are robust to different background noises and telephony networks.

Table 4.3: *Diarization error for landline vs. cellular on dry run set*

Tele Network	Miss	False Alarm	Diarization Err
Landline	4.8%	4.4%	9.22%
Cellular	4.0%	5.3%	9.29%

Table 4.4 compares the performance across different participant systems in the RT03s evaluation for this seperate channel activity detection task including Cambridge University system

Table 4.4: *Performance comparison for seperate channel speec activity detection across systems in RT03s evaluation*

System	CU	ISL	LL
Diarization Err	11.63%	11.41%	11.52%

(CU), our system (ISL) and Lincoln Laboratory system (LL). There is no significant difference in performance across systems. Our results can represent the state-of-the-art performance in the NIST RT-03S evaluation for seperate channel speech activity detection.

4.6 Speaker Segmentation and Clustering on Meetings

The full automatic transcription of meetings is considered an AI-complete, as well as an ASR-complete, problem [75]. It includes transcription, meta-data extraction, summarization, etc. In recent years, the study of multi-speaker meeting audio has seen a surge of activity at many levels of speech processing, as exemplified by the appearance of large meeting speech corpora from several groups, important observations available in the literature [11][107], and the new evaluation paradigm launched by NIST, the Rich Transcription Evaluation on Meetings.

4.6.1 Data Description

The experiments throughout this section were conducted on the RT-04S meeting data. Each meeting was recorded with personal microphones for each participant (close-talking microphones), as well as multiple room microphones (distant microphones) placed on the conference table. In this section we focus on the task of automatic speaker segmentation and clustering based on multiple distant microphone (MDM) channels.

Both the development and the evaluation datasets from the NIST RT-04S evaluation were

Table 4.5: *RT04s Development dataset*

MeetingID (abbreviation)	#Spkrs	cMic	#dMic
CMU_20020319-1400 (CMU1)	6	L	1
CMU_20020320-1500 (CMU2)	4	L	1
ICSI_20010208-1430 (ICSI1)	7	H	4
ICSI_20010322-1450 (ICSI2)	7	H	4
LDC_20011116-1400 (LDC1)	3	L	8
LDC_20011116-1500 (LDC2)	3	L	8
NIST_20020214-1148 (NIST1)	6	H	7
NIST_20020305-1007 (NIST2)	7	H	6

used. The data were collected at four different sites, including CMU, ICSI, LDC, and NIST [12][49][111][110]. The development dataset consists of 8 meetings, two per site. Ten minute excerpts of each meeting were transcribed. The evaluation dataset also consists of 8 meetings, two per site. Eleven minute excerpts of each meeting were selected for testing. All of the acoustic data used in this work is of 16kHz, 16-bit quality. Table 4.5 gives a detailed description of the RT-04S development dataset, on which we subsequently report detailed performance numbers. "cMic" refers to close-talking microphone used. "L" stands for lapel and "H" stands for headset. "#dMic" is the number of distant microphones provided for each meeting.

4.6.2 System Overview

The MDM system consists of following steps:

- initial speech/non-speech segmentation for each channel
- unification of the initial segmentations across multiple channels

- best channel selection for each segments
- speaker change detection in long segments
- speaker clustering on all segments
- smoothing.

Initial speech/non-speech segmentation is generated using the acoustic segmentation software CMUseg_0.5. We removed the classification and clustering components and used it as a segmenter. A detailed description of the algorithms used in this software can be found in [108].

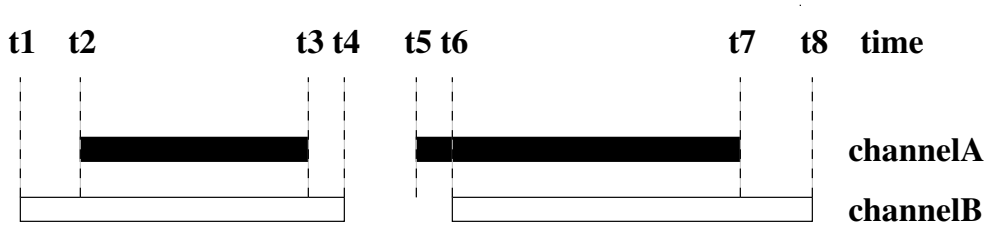

Figure 4.4: *Multiple Channel Unification*

In the **multiple channel unification** step, the segment boundaries are unified across multiple channels. Figure 4.4 shows an example for two distant microphone channels. The initial segmentation produces two speech segments on channel A, (t2, t3) and (t5, t7); and two segments, (t1, t4) and (t6, t8), on channel B. After unification, the segments across the two channels are (t1, t2), (t2, t3), (t3, t4), (t5, t6), (t6, t7) and (t7, t8).

We then conduct a **best channel selection** for each of the segments produced during the unification step. We compute the minimum energy ($MinE_i$), maximum energy ($MaxE_i$), and the signal-to-noise ratio (SNR_i) within each segment on all channels. We select the best channel for each segment according to following criterion,

$$i^{\star} = \text{argmin}_i \left(\frac{MinE_i}{MaxE_i} \times \frac{1}{SNR_i} \right) \tag{4.8}$$

Speaker change detection is applied to any segment that is longer than 5 seconds to check whether there exist speaker turn changes within such long segments. We choose 5 seconds because the majority of segments in meeting is shorter than 5 seconds as shown in figure 4.1 and this was found to give optimal segmentation accuracy via cross-validation on the development set. The procedure is the same as shown in Figure 4.2. **Speaker clustering** is then performed on all segments. The same hierarchical, agglomerative clustering technique as described in section 4.5 is applied here.

In the final **smoothing** step, we merge any two segments that belong to the same speaker and have less than a 0.3 second gap between them. This is based on our experience in the RT-03S evaluation.

4.6.3 Experimental Results

Speaker Segmentation Performance

Table 4.6: *Speaker Segmentation Performance (in %) on dev set*

System Stage	Precision	Recall
Initial	86.83	11.60
Unification	87.74	19.00
Change Detection	85.17	76.41

Table 4.6 shows the speaker segmentation performance at different system stages. Not surprisingly, the low recall of the initial segmentation indicates high deletion errors, which means that a lot of speaker changes are missed. Multiple channel unification compensates a little for the deletion errors. Speaker change detection leads to a big improvement in recall while suffering only a small decrease in precision.

Speaker Diarization Performance

Table 4.7: *Speaker Diarization Performance (in %)*

Error	Development Set		Evaluation Set	
	Include	Exclude	Include	Exclude
Miss	8.7	0.0	19.8	0.4
False Alarm	3.3	2.9	2.6	4.1
Spkr Err	25.1	26.7	17.8	23.4
Diarization Err	**37.11**	**29.59**	**40.19**	**28.17**

Table 4.7 shows the overall speaker diarization performance on the development set and on the evaluation set, both when including regions of overlapping speech and when excluding the regions of overlapping speech. Comparable results are achieved on both datasets. The dominant error among the three error components is speaker error.

In Table 4.8 we show the speaker diarization performance on individual meetings of the development set. The results exhibit large variability over meetings collected at different sites. We think that this variability may be due to unquantified meeting characteristics such as overall degree of crosstalk, general meeting geometry including room acoustics and microphone variability within a meeting. However, we noticed that our system often underestimates the number of speakers involved in a meeting. Although on meetings CMU2 and NIST1 the system underestimates the number of speakers, it still achieves better performance compared to most other meetings. This is due to the fact that both these two meetings have a dominant speaker who talks for more than 70% of the time. We compute the speaker speaking time

Table 4.8: *Speaker Diarization Performance on individual meeting in dev set including overlapping speech (in %)*

Meeting	Miss	False Alarm	Spkr Err	Diarization Err	#ref	#sys
CMU1	12.6	4.3	30.3	47.12	6	4
CMU2	3.4	5.0	16.3	24.72	4	2
ICSI1	4.7	2.9	35.0	42.62	7	4
ICSI2	9.8	1.1	37.0	47.92	7	3
LDC1	6.2	2.6	9.0	17.78	3	3
LDC2	17.3	1.1	11.0	29.41	3	3
NIST1	7.2	7.1	11.7	26.01	6	2
NIST2	6.5	3.1	49.5	59.04	7	2

entropy $H(Meeting)$ for each meeting,

$$\begin{aligned} H(Meeting) &= -\sum_{i=1}^{M} P(S_i) * \log P(S_i) \\ P(S_i) &= \frac{T(S_i)}{\sum_{i=1}^{M} T(S_i)} \end{aligned}$$

where M is the number speakers involved in the meeting. $T(S_i)$ is the total time that speaker S_i speaks. $P(S_i)$ is the percentage of time (ie. probability) that speaker S_i speaks. The lower the entropy, the more biased is the distribution of the speaker speaking time in the meeting. As $H(Meeting) \rightarrow 0$, it becomes more likely that there is only one dominant speaker in the meeting.

Figure 4.5 shows the speaker diarization error on each individual meeting in the development set versus its speaker speaking time entropy. We can see from the figure that our system tends to produce lower speaker diarization error on meetings that have lower speaker speaking time entropy. We think the reason that the two CMU meetings do not follow this trend is that there is

only one distant microphone channel provided. This makes it harder in general to segment and cluster relative to other meetings, for which multiple distant microphone channels are provided.

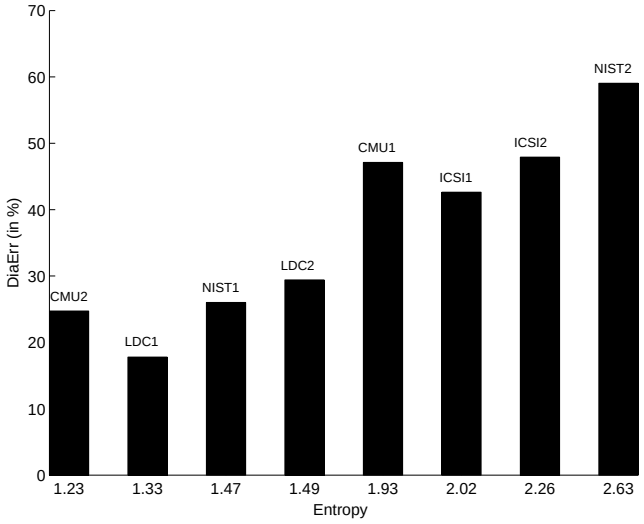

Figure 4.5: *Speaker speaking time entropy vs. diarization error.*

We also conducted an experiment as follows. We assume a one-to-one mapping between channel and speaker. We use the best channel information only, which was provided in the channel selection step described in section 4.6.2. We do not perform speaker clustering. For any two segments, if the channel selection process produces the same best channel for them, we assume these two segments belong to the same speaker. This yields 55.45% and 52.23% speaker diarization error when including and excluding overlapping speech, respectively. It indicates that there is rich information that can be used to aid in speaker segmentation and clustering from the multi-channel recordings. Our current system utilizes such information implicitly by doing best channel selection.

Table 4.9 compares the performance across different participant systems in the RT04s meeting evaluation including joint systems from CLIPS and LIA laboratories (LIA+CLIPS), our system (ISL) and Macquarie University system (Macquarie). Our system was ranked number 2 among the three systems.

Table 4.9: *Performance comparison across systems in RT04s evaluation*

System	LIA+CLIPS	ISL	Macquarie
Diarization Err exclude overlapping	23.54%	28.17%	62.0%
Diarization Err include overlapping	37.53%	40.19%	69.1%

4.7 Impact on Speech Recognition

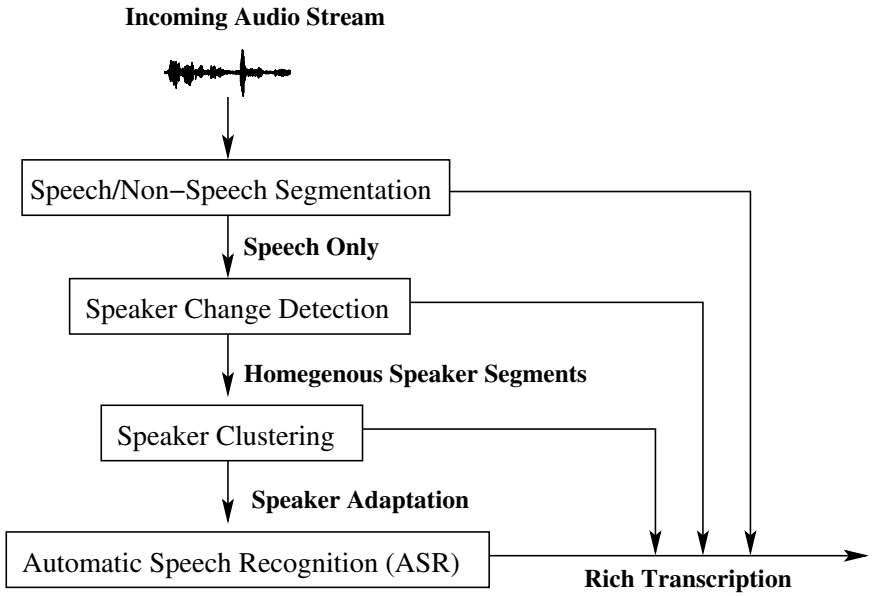

Figure 4.6: *Speaker Segmentation and Clustering impact on Speech Recognition*

Figure 4.6 shows a typical ASR system work-flow for Rich Transcription task and how components in the speaker segmentation and clustering system impact on ASR system. The speech/non-speech segmentation allows only speech segments to be passed to the recognizer, saving computation time as well as improving recognition accuracy. The speech segments are further segmented in terms of speakers, which is useful for ASR as decoding of well segmented and manageable speech chunks is always more easier and accurate. These homogeneous speaker segments can be clustered together in terms of speakers to facilitate speaker

adaptation of an ASR system, which has been shown to significantly improve ASR accuracy [16] [55] [89]. Finally, the output of all these modules can also be combined with the output of the ASR system, resulting in Rich Transcription.

Table 4.10: *Word error rate on RT04s dev set*

Acoustic Models	Manual Segmentation	MDM Segmentation
PLAIN	53.4%	54.4%
SAT/CAT	46.6%	48.5%
SAT/CAT-VTLN	43.3%	45.5%
Multi-pass CNC	42.8%	45.0%

Table 4.10 compares the automatic speech recognition performance in word error rate based on manual segmentation vs. on segmentation provided by our MDM speaker segmentation and clustering system [73]. The first column refers to different acoustic models in the ASR system:

- **PLAIN** Merge-and-Split training followed by Viterbi (2 iteration) on the close-talking data only, no VTLN
- **SAT/CAT** Extra 4 iteration Viterbi training on the distant data, no VTLN (speaker adaptive training (SAT); cluster adaptive training (CAT))
- **SAT/CAT-VTLN** $\equiv$ SAT/CAT, but trained with VTLN
- **Multi-pass CNC** confusion network combination

We lose 1% to 2.2% absolute in word error rate based on automatic segmentation compared to manual segmentation. It is clear that speaker segmentation and clustering plays a vital role in improving the performance of adaptation. We have noticed that speech recognition has a different requirement for speaker segmentation and clustering. In speech recognition, the goal

of speaker segmentation and clustering is to provide clean single speaker segments for speaker adaptation. Speaker adaptation is concerned more with the regression of speakers, than with the strict classification of speakers. So if two speakers sound similar, they can be considered as equal and grouped into one cluster. It actually would be rather desirable for speech recognition to group similar speakers together, so that more data is available for adaptation. Therefore, a specific speaker segmentation and clustering system tuned for speech recognition may achieve better word error rate even if speaker diarization performance is worse.

Our system has been used widely in many other evaluations, such as in the TC-STAR Evaluation [112], in the NIST RT04 Mandarin Broadcast News Evaluation [127] and in the GALE Evaluations (2006).

4.8 Chapter Summary

In this chapter we presented techniques for speaker segmentation and clustering (TGMM-GLR) that do not require a prior training. These techniques are evaluated in the NIST RT-03S evaluation on the conversational telephony speech. We achieve state-of-the-art performance with a diarization error of 11.4%. The performance analysis also shows that these techniques are effective and robust against different background noises and telephony networks. They show the potential of domain independence. We also presented our automatic speaker segmentation and clustering system for natural, multi-speaker meeting conversations based on multiple distant microphones. The performed experiments show that the system is capable of providing useful speaker information on a wide range of meetings. The system achieved a 28.17% speaker diarization error in the NIST RT-04S evaluation. The speaker segmentation and clustering techniques also play significant roles in our automatic speech recognition systems, which break the continuous audio stream into manageable chunks applicable to the configuration of the ASR system and provide speaker information that is used for efficient speaker adaptation.

Chapter 5

Person Identification System

5.1 Introduction

Person identification consists of determining the identity of a person from a data segment, such as a speech, video segment, etc. Currently, there is a high interest in developing person identification applications in the framework of smart room environments. Person identification in smart environments is very important in many aspects. For instance, customization of the environment according to the person's identity is one of the most useful applications. In a smart room, the typical situation is to have one or more cameras and several microphones as shown in figure 5.1. Perceptually aware interfaces can gather relevant information to model and interpret human activity, behavior and actions. Such applications face an assortment of problems such a mismatched training and testing conditions or the limited amount of training data.

The objective of the CHIL (Computers in Human Interaction Loop) project [20] is to create environments in which computers serve humans who focus on interacting with other humans as opposed to having to attend to and being preoccupied with the machines themselves. Instead of computers operating in an isolated manner, and humans thrust in the loop of computers, we put computers in the human interaction loop (CHIL) and design computer services that

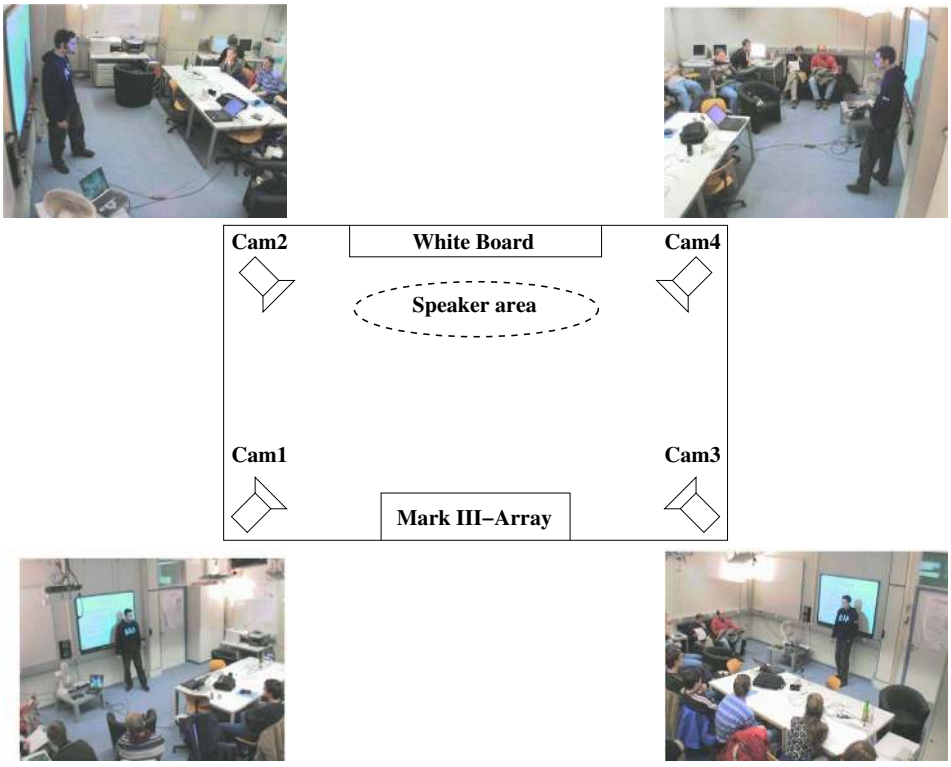

Figure 5.1: *Audio and Video sensors setup in a typical smart-room environment*

model humans and the state of their activities and intentions. Based on the understanding of the human perceptual context, CHIL computers are enabled to provide helpful assistance implicitly, requiring a minimum of human attention or interruptions

A data corpus and evaluation procedure has been provided in the CHIL project to encourage the research efforts for person identification in smart environments. Following the two successful uni-modal identification (audio-only and video-only) evaluations, this year multi-modal identification is also included to the person identification task.

5.2 Multimodal Person Identification

Multimodal recognition involves the combination of two or more human traits like voice, face, fingerprints, iris, hand geometry, etc. to achieve better performance than using unimodal recognition. In our Person ID system, audio-based identification (speaker identification) and video-based identification (face recognition) are combined.

5.2.1 Audio-based Identification

Our audio-based identification (speaker identification) system is a GMM based system. Reverberation compensation and feature warping as described in detail in Chapter 2 are applied in the feature processing stage. In our system, we use 13-dimensional MFCC as speaker features and 128 Gaussians and 32 Gaussians as speaker models under the 30-second training condition and 15-second training condition respectively. We will show why we choose these numbers of Gaussians in the experimental results section.

5.2.2 Video-based Identification

Our video-based identification system is a face recognition system [31]. The face recognition system processes multi-view, multi-frame visual information to obtain an identity estimate. The system consists of the following building blocks:

- Image alignment
- Feature Extraction
- Camera-wise classification
- Score normalization

- Fusion over camera-views

- Fusion over image sequence

The system receives an input image and the eye-coordinates of the face in the input image. The face image is cropped and aligned according to the eye coordinates. If in the image only one eye is visible, it is not processed. The aligned image is then divided into non-overlapping 8x8 pixels resolution image blocks. Discrete cosine transform (DCT) is applied on each local block. The obtained DCT coefficients are ordered using zig-zag scan pattern. From the ordered coefficients, the first one is removed since it only represents the average value of the image block. The first M coefficients are selected from the remaining ones [120]. To remove the effect of intensity level variations among the corresponding blocks of the face images, the extracted coefficients are normalized to unit norm.

Classification is performed by comparing the extracted feature vectors of the test image, with the ones in the database. Each camera-view is handled separately. That is, the feature vectors that are extracted from the face images acquired by Camera 1 are compared with the ones that are also extracted from the face images acquired by Camera 1 during training. This approach speeds up the system significantly. That is, if we have N images from each camera for training, and if we have R images from each camera for testing, and we have C cameras, it requires $(C*N)*(C*R)$ similarity calculations between the training and testing images. However, when we do camera-wise image comparison, then we only need to do $C*(N*R)$ comparisons between the training and testing images. Apparently, this reduces the amount of required computation by a factor of C. In addition to the improvement in system's speed, it also provides a kind of view-based approach that separates the comparison of different views, which was shown to perform better than doing matching between all the face images without taking into consideration their view angles [84]. Distance values obtained from each camera-view are

normalized using Min-Max rule, which is defined as:

$$ns = 1 - \frac{s - min(S)}{max(S) - min(S)}$$

where, s corresponds to a distance value of the test image to one of the training images in the database, and S corresponds to a vector that contains the distance values of the test image to all of the training images. The division is subtracted from one, since the lower the distance is, the higher the probability that the test image belongs to that identity class. This way, the score is normalized to the value range of [0,1], best match having the score "1", and the worst match having the score "0". These scores are then normalized by dividing them to the sum of the confidence scores. The obtained confidence scores are summed over camera-views and over image-sequence. The identity of the face image is assigned as the person who has the highest accumulated score. This face recognition system is developed by our colleague Hazim Ekenel [31].

5.2.3 Multimodal Person Identification

Multimodal identification is performed by fusing the match scores of both modalities (audio and video). In a multimodal biometric system that uses several characteristics, fusion is possible at three different levels: feature extraction level, matching score level or decision level. Fusion at the feature extraction level combines different biometric features in the recognition process, while decision level fusion performs logical operations upon the unimodal system decisions to reach a final resolution. Score level fusion matches the individual scores of different recognition systems to obtain a multimodal score. Fusion at the matching score level is usually preferred by most of the systems. Matching score level fusion is a two-step process: normalization and fusion itself [32] [48] [69] [121]. Since unimodal scores are usually non-homogeneous, the normalization process transforms the different scores of each unimodal system into a comparable range of values. One of the most conventional normalization methods is z-score (ZS) [69]

[121], which normalizes the global mean and variance of the scores of a unimodal biometric. Denoting a raw matching score as a from the set A of all the original unimodal biometric scores, the z-score normalized biometric x is calculated according to the formula as follows:

$$x_{zs} = \frac{a - mean(A)}{std(A)} \tag{5.1}$$

where $mean(A)$ is the statistical mean of A and $std(A)$ is the standard deviation. After normalization, the converted scores are combined in the fusion process in order to obtain a single multimodal score. Product and sum are the most straightforward fusion methods. Other fusion methods are min-score and max-score that choose the minimum and the maximum of the unimodal scores as the multimodal score. "Matcher Weighting" is a fusion method where each biometric is weighted by a different factor proportional to the recognition result of the biometric, and in the user weighting method different weighting methods are applied for every user.

In our person identification system, the scores from each of the two unimodal systems (speaker identification and face recognition) are normalized using z-score techniques, then they are fused via sum rule.

5.3 Data Setup and Experimental Results

5.3.1 Experimental Setup

Table 5.1: *CLEAR 2006 Evaluation Test Dataset (%)*

Segment Duration	Num of segments
1	613
5	411
10	289
20	178

Classification of Events, Activities and Relationships (CLEAR) is a international technology evaluation supported by CHIL, NIST and the US ARDA VACE program. A set of audiovisual recordings of seminars and of highly-interactive small working groups seminars have been used. These recordings were collected by the CHIL consortium for the CLEAR 06 Evaluation. The recordings were done according to the "CHIL Room Setup" specification [15]. Data segments are short video sequences and matching far-field audio recordings taken from the above seminars.

To evaluate how the duration of the training signals impacts the performance of the system, two training durations have been considered: 15 and 30 seconds. Test segments of different durations (1, 5, 10 and 20 seconds) have been used during the algorithm development and testing phases. A total of 26 personal identities have been used in the recognition experiments. Each seminar has one audio signal from the microphone number 4 of the Mark III array. Each audio signal has been divided into segments which contain information of a unique speaker. These segments have been merged to form the final testing segments of 1, 5, 10 and 20 seconds (see Table 5.1) and training segments of 15 and 30 seconds. Video is recorded in compressed

JPEG format, with different frame-rates and resolutions for the various recordings.

Far-field conditions have been used for both modalities, i.e. corner cameras for video and Mark III microphone array for audio as shown in Figure 5.1. In the audio task only one array microphone has been considered for both development and testing phases. In the video task, we have four fixed position cameras that are continuously monitoring the scene. All frames in the 1, 5, 10, 20 sec segments and all synchronous camera views can be used and the information can be fused to find the identity of the concerned person. To find the faces to be identified, a set of labels is available with the position of the bounding box for each person's face in the scene. These labels are provided each one second. The face bounding boxes are linearly interpolated to estimate their position in intermediate frames. To help this process, an extra set of labels is provided, giving the position of both eyes of each individual every 200 ms.

The metric used to benchmark the quality of the algorithms in the CLEAR evaluation is the Miss Classification Rate (MCR) in percentage.

5.3.2 Experimental Results

Audio-based Identification Results

To find an optimal number of Gaussians for a speaker model, we conducted several speaker identification experiments with different number of Gaussians in a speaker model. We use the evaluation data in CHIL 2005 Spring Evaluation [18] to decide the number of Gaussians for speaker models. This data set has been carried out on the union of the UKA-ISL_ Seminar_2003 and UKA-ISL_Seminar_2004 databases. Non-speech segments have been manually removed both from the training and the testing segments. There are two microphone conditions: Closed-Talking-Microphone (CTM) and Microphone Array (ARR). The duration and number of segments selected for the training and testing as improving our system is described in Table 5.2

Table 5.2: *CHIL 2005 Spring Evaluation Dataset (%)*

Segment ID	Duration	Num of CTM segments	Num of ARR segments
Train A (15 sec)	30	11	11
Train B (30 sec)	15	11	11
Test	5	1100	682

Table 5.3 and 5.4 show that speaker identification error rate changes while the number of Gaussians changes in a speaker model. According Table 5.3 and 5.4, we choose to use 128 Gaussians for the 30-second training condition and 32 Gaussians for the 15-second training condition.

Table 5.3: *Performance with different number of Gaussians for Train B (30-sec) training duration (%)*

Num Gaussians	64	128	256
MCR	0.36	0.27	0.36

Table 5.4: *Performance with different number of Gaussians for Train A (15-sec) training duration (%)*

Num Gaussians	16	32	64
MCR	2.82	2.00	2.23

The final results of audio identification on the CLEAR 2006 evaluation dataset are shown in Table 5.5. We can see from the table that more training data and longer test gets help to reduce the false identification rate.

Table 5.5: *CLEAR 2006 Audio Person ID in Error Rate (%)*

Duration	Segments	Train A (15-sec)	Train B (30-sec)
1	613	23.7	14.4
5	411	7.8	2.2
10	289	7.3	1.4
20	178	3.9	0.0

Video-based Identification Results

In face recognition experiments, face images are aligned according to eye-center coordinates and scaled to 40x32 pixels resolution. Only every five frame that has the eye coordinate labels is used for training and testing. The aligned image is then divided into 8x8 pixels resolution non-overlapping blocks making 20 local image blocks. From each image block 10 unit norm DCT-0 coefficients are extracted and they are concatenated to construct the 200-dimensional final feature vector. The classification is performed using nearest neighbor classifier. L1 norm is selected as the distance metric, since it has been observed that, it consistently gives the best correct recognition rates when unit norm DCT-0 coefficients are used. The distance values are converted to the matching scores by using the Min-Max rule. The normalized matching scores are accumulated over different camera views and over image sequence. The identity candidate that has the highest score is assigned as the identity of the person.

The miss classification rates for different training and testing durations can be seen in Table 5.6. As can be observed from the table, the increase in the training segments' duration or in the testing segments' duration decreases the miss classification rate.

Table 5.6: *CLEAR 2006 Video Person ID in Error Rate (%)*

Duration	Segments	Train A (15-sec)	Train B (30-sec)
1	613	46.8	40.1
5	411	33.6	23.1
10	289	28.0	20.4
20	178	23.0	16.3

Multimodal Person Identification Results

In this section we summarize the results for the evaluation of different modalities and the result improvement with the multimodal technique. In the following tables, we show the identification error rate for both audio and video unimodal modalities and multimodal fusion. The first column shows the duration of test segments in seconds. The second column shows the number of tested segments. Train A and B are the training sets of 15 seconds and 30 seconds.

Table 5.7: *Multimodal Person ID in Error Rate (%) with Equal Fusion Weights*

Duration	Segments	Train A (15-sec)			Train B (30-sec)		
		Audio	Video	Fusion	Audio	Video	Fusion
1	613	23.7	46.8	29.2	14.4	40.1	19.3
5	411	7.8	33.6	17.7	2.2	23.1	10.0
10	289	7.3	28.0	17.5	1.4	20.4	10.5
20	178	3.9	23.0	13.5	0.0	16.3	7.3

Table 5.8: *Multimodal Person ID in Error Rate (%) with Unequal Fusion Weights*

Duration	Segments	Train A (15-sec)			Train B (30-sec)		
		Audio	Video	Fusion	Audio	Video	Fusion
1	613	23.7	46.8	**19.4**	14.4	40.1	**12.9**
5	411	7.8	33.6	**7.6**	2.2	23.1	**1.7**
10	289	7.3	28.0	**6.8**	1.4	20.4	**2.9**
20	178	3.9	23.0	**4.5**	0.0	16.3	**1.1**
All	1491	13.8	36.7	**11.9**	6.8	28.8	**6.5**

Table 5.7 summarizes the person ID results when audio ID system and video ID system are fused together using equal weights. Again, it can be observed that the increasing in training segments' duration or in test segments' duration decreases the false identification rate. Due to equal weighting of each modality, the multimodal identification results are better than the video-only results and worse than the audio-only results. To better combine the audio and video modalities, we weight the two modalities differently according to the identification performance of the uni-modality, which means we gave higher weights to audio-only scores for fusion. Table 5.8 shows the performance improvement of the multimodal system by fusion over the audio-only and video-only systems. From the results we see that fusion of multi modalities can significantly improve the performance over each of the uni-modalities. Although there is other sophisticated fusion logic, in this thesis we used simply linear fusion strategy.

Table 5.9 compares the performance of uni-modal and multimodal systems from different participants. Our audio-based and video-based unimodal systems achieved the best performance among three participants. However, notice that although UPC system had worse audio-based performance and much worse video-based performance, it achieved fusion performance close to our performance. This indicates fusion strategies is very crucial in a multimodal system.

Table 5.9: *Multimodal Person ID in Error Rate (%) accross different systems*

	Duration	Segments	Train A			Train B		
			Audio	Video	Fusion	Audio	Video	Fusion
UPC	1	613	25.0	79.8	**23.2**	16.0	80.4	**13.8**
	5	411	10.7	78.6	**8.0**	2.9	77.1	**2.9**
	10	289	10.7	77.5	**5.9**	3.8	74.4	**2.0**
	20	178	11.8	76.4	**4.0**	2.8	73.0	**1.1**
	All	1491	16.7	78.7	**13.3**	8.4	77.5	**6.8**
ISL	1	613	23.7	46.8	**19.4**	14.4	40.1	**12.9**
	5	411	7.8	33.6	**7.6**	2.2	23.1	**1.7**
	10	289	7.3	28.0	**6.8**	1.4	20.4	**2.9**
	20	178	3.9	23.0	**4.5**	0.0	16.3	**1.1**
	All	1491	13.8	36.7	**11.9**	6.8	28.8	**6.5**
AIT	1	613	26.9	50.6	**23.7**	15.2	47.3	**13.7**
	5	411	9.7	29.7	**6.8**	2.9	31.1	**2.2**
	10	289	8.0	23.2	**6.6**	1.7	26.6	**1.7**
	20	178	4.5	20.2	**2.8**	0.6	24.7	**0.6**
	All	1491	15.8	35.9	**13.2**	7.4	36.1	**6.6**

5.4 Chapter Summary

In this chapter, we presented our multimodal person identification system, which combines speaker identification modality and face recognition modality. We show the evaluation results of our multimodal person identification results in the CLEAR 2006 evaluation. We achieved best performance for both unimodal systems. Although the results of the person identification

system at the CLEAR evaluation are far superior for audio than video recognition, fusion the two systems brings additional gains. Although the results show that video only provides minor improvement of the audio recognition, this is not generally true. One obvious reason is that speech is usually much sparser than face images in a multi-camera setup. In the CLEAR evaluations, care has been taken to have segments with speech available. More balanced contribution from both modalities is expected in real unsupervised scenarios. Also more sophisticated fusion strategies may be deployed to more efficiently fuse multi modalities.

Chapter 6

Conclusion

6.1 Summary of Results and Thesis Contributions

In this thesis, we conducted research work to improve speaker recognition robustness on far-field microphones from two directions: to improve robustness of traditional speaker recognition system based on low-level features and to improve robustness by using high-level features. We also implemented sytems that support robust speaker recognition, including a speaker segmentation and clustering system aiming at robust speaker recognition in multi-speaker scenarios and a person identification system by integrating audio and visual modalities.

- We first investigated approaches to improve speaker recognition robustness on far-field distant microphones, which is a research area has not received much attention. We introduced a reverberation compensation approach and applied feature warping in the feature processing. These approaches bring significant gain. We proposed four multiple channel combination approaches to utilize information from multiple sources. These approaches achieve significant improvement over baseline performance, especially in the case that test condition can not be covered in the training.

- We introduced a new approach to model a speaker's pronunciation idiosyncrasy from two complementary dimensions: time dimension and cross-stream dimension. Each dimension contains useful information for distinguishing speakers pronunciation characteristics. Combining both dimensions achieves significant better performance than that of each single dimension. The proposed approach has the potential of language independence. This research along with work from other researchers in phonetic speaker recognition inspires other researchers to exploit high-level features for speaker recognition. In addition, the proposed approach was applied to other classification tasks, such as language identification and accent identification, and achieved good performance as well.

- We studied speaker segmentation and clustering across domains such as telephone conversations and meetings. We implemented a speaker segmentation and clustering system which was tested within the NIST Rich Transcription evaluations. It is also a very important module in a complete ASR system, such as BN system, meeting system, and lecture recognition system etc. It provides crucial information for speaker adaptation.

- We integrated speaker recognition modality with face recognition modality and built a robust person identification system which was tested in the NIST CLEAR06 evaluation.

6.2 Future Research Directions

The speaker recognition problem can be formulated mathematically as follows [44]:

$$S^* = \arg\max_S P(S|O) \qquad O = \{X, W, F, C, \cdots\} \tag{6.1}$$

where S is speaker identity and O is the observation. Observation can take many forms including low-level features such as Mel-cepstrum X, word or phone or phrase information W, prosodic information F, and channel information C (including handheld/handsfree landline,

wireless, PC microphones, conference room microphones etc) and so on. We can factorize the problem formula 6.1 into speaker knowledge components with the assumption that W and F are both independent of C:

$$\begin{aligned} S^* &= \arg\max_S P(S|O) \\ &= \arg\max_S P(S|X,W,F,C) \\ &= \arg\max_S \frac{P(X|S,W,F,C)}{P(X|W,F,C)} * \frac{P(F|W,S)}{P(F|W)} * \frac{P(W|S)}{P(W)} * \frac{P(C|S)}{P(C)} * P(S) \end{aligned} \tag{6.2}$$

$\frac{P(X|S,W,F,C)}{P(X|W,F,C)}$ can be considered as text-dependent speaker recognition; $\frac{P(F|W,S)}{P(F|W)}$ can be the expression of speaker-dependent prosodic modeling; $\frac{P(W|S)}{P(W)}$ can be explained as speaker dependent word/phonetic modeling; $\frac{P(C|S)}{P(C)}$ can be described as speaker's channel profile; and $P(S)$ can be considered as a speaker's profile (prior).

As mentioned before, human listeners are aware of multiple information such as prosody, word choice, pronunciation, accent, and other speech habits (for example laughs). when recognizing speakers, while traditional systems only rely on seemingly one source of information. As seen in the high-level speaker recognition research area highlighted by the SuperSID project, researchers have already started to exploit different types of high-level information for speaker recognition. However it is still an unfinished research, there are still more high-level features that have not been investigated such as how a person interact in a multi-party conversations and a person's emotion status etc. On the other hand, humans rely on different level of information under different contexts. Current systems do not aware the contexts and use the different levels of information equally. More research effort is needed to explore how to use different levels of information more efficiently.

As shown in Chapter 4, multiple channel information is useful for speaker segmentation and clustering in the meeting scenarios with multiple microphone setups. It is beneficial to investigate this issue. We have studied how to use information from multiple close-talking microphones in meeting scenarios for speaker segmentation and clustering as shown in [65].

Other researchers studied this topic too as shown in [85] [26] [119]. However, only the multiple close-talking microphone conditions are studied. There has been limited effort in studying the multiple distant microphone condition.

To solve the sparse data problem for new speakers is important because we will face such problem in real applications. Approaches are desired to incrementally learn and adapt speaker models. Other modalities are helpful if available to reliably detect new speakers and identify previously enrolled speakers.

Appendix A

Open-Set Speaker Identification

A.1 Introduction

In this section we present how our system performs in the open-set stituation. As mentioned in the first chapter, speaker identification can be divided into two categories: closed-set speaker identification and open-set speaker identification [34] [35]. Given a set of enrolled speakers and a test utterance, open-set speaker identification is defined as a twofold problem. Firstly, it is required to identify the speaker model in the set, which best matches the test utterance. Secondly, it must be determined whether the test utterance has actually been produced by the speaker associated with the best-matched model, or by some unknown speaker outside the enrolled set. As shown in figure A.1, the open-set speaker identification can be considered as closed-set speaker identification plus speaker verification.

The potential errors and difficulties in open-set speaker identification can be analysed as follows. Suppose that M speakers are enrolled in the system and their statistical model descriptions are $\Theta_1, \Theta_2, \ldots, \Theta_M$. If O denotes the feature vector sequence extracted from the test

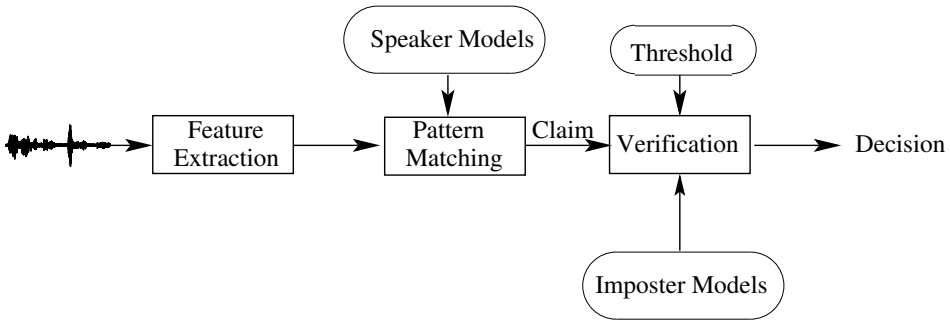

Figure A.1: *Block diagram of open-set speaker identification system*

utterance, then the open-set identification can be stated as:

$$\max_{1\leq i\leq M}\{p(O|\Theta_i)\} \gtrless \zeta \quad \rightarrow \quad O \in \begin{cases} \Theta_k,\ k = \arg\max\limits_{1\leq i\leq M}\{p(O|\Theta_i)\} \\ \textit{unknown speaker model} \end{cases} \tag{A.1}$$

where ζ is a pre-determined threshold. In other words, O is assigned to the speaker model that yields the maximum likelihood over all other speaker models in the system, if this maximum likelihood score itself is greater than the threshold ζ. Otherwise, it is declared as originated from an unknown speaker. It is evident from the above description that, for a given O, three types of error are possible:

- **False Acceptance(FA):** the system accepts an impostor speaker as one of the enrolled speakers.
- **False Rejection(FR):** the system rejects a true speaker.
- **Speaker Confusion(SC):** the system correctly accepts a true speaker but confuses him/her with another enrolled speaker.

These types of errors are referred to as FA, FR, and SC respectively. Open-set identification is a two-stage process. For a given O, the first stage determines the speaker model that yields

the maximum likelihood, and the second stage makes the decision to assign O to the speaker model determined in the first stage or to declare it as originated from an unknown speaker. The first stage is responsible for generating SC error whereas, both FA and FR are the consequences of the decision made in the second stage.

An important point to note about this two-stage process is that the latter stage is far more susceptible to distortions in the characteristics of the test utterance than is the former stage. This is because, in the former stage, since the same test utterance is used to compute all the likelihood scores, the distortions in the test utterance are likely to be similarly reflected in all the likelihood scores. As a consequence, the selection of the model that yields the maximum likelihood is likely to be unaffected. On the other hand, in the second stage, the absolute maximum likelihood score is compared against a threshold determined a priori and without any knowledge about the characteristics of the distortion in the test utterance.

It should be pointed out that a task similar to that described above (in the second stage of open-set identification) is also encountered in speaker verification. However, in speaker verification, the problem is not as challenging. To be more specific, the challenge in open-set identification can be viewed as a special (but unlikely) scenario in speaker verification in which each impostor targets the speaker model in the system for which he/she can achieve the highest score.

Although more sophisticated decision logic may be deployed, our focus is not on this issue. Our goal is simply to evaluate how our system works under the open-set situation.

A.1.1 Data Description and Experimental Setup

For the set of open-set speaker identification experiments, we use both 3D Distant Microphone Database and 2D Distant Microphone Database. Each of these databases have 8 microphone channels recording. There are in total 24 speakers in 3D Distant Microphone Database and 30

speakers in 2D Distant Microphone Database. We randomly select 27 speakers out of the total 54 speakers as target speakers. The female and male speakers are balanced in the selection. In the following experimental results report, we use CH1 to CH8 to refer to the 8 microphone channels. The naming keeps the same as for the 3D Distant Microphone Database, while for the 2D Distant Microphone Database, CH1 corresponds to Dis0, CH2 corresponds to Dis1, CH3 corresponds to Dis2, CH4 corresponds to Dis4, CH5 corresponds to Dis5, CH6 corresponds to Dis6, CH7 corresponds to Dis8, CH8 corresponds to DisL. For each of the target speakers, we randomly select 60 seconds from his/her training speech to train a 256-mixture GMMs as the target speaker model. For the impostor model training, we use leave one out strategy. We exclude one impostor speaker from the entire impostor group. We randomly select 60 seconds training speech from each of the impostor speakers' training data in the group and pool them together to train one 512-mixture GMMs as the impostor model. The remaining speech for each of the target and impostor speakers are divided into 20 seconds segments and are used as test trials. There are in total 964 test trials, 451 target speaker test trials and 513 impostor speaker test trials.

A.1.2 Experimental Results

We plot the tradeoff between the three types of errors as a function of a decision threshold. Figure A.2 shows such tradeoff for the average performance under mismatched conditions as a function of threshold values. The equal error rate is about 3%.

Figure A.3 summarizes the system performance in total errors with different threshold value setup under mismatched conditions. Among the three threshold values, 0.8 achieves the best performance under both matched and mismatched conditions on all the channels. Although we see different performance on different test channel, our focus is not to discover the reason for this matter. Our expect is to see the same trend for one threshold setting on all the channels. Table A.1 shows the detailed average performance under mismatched conditions with threshold

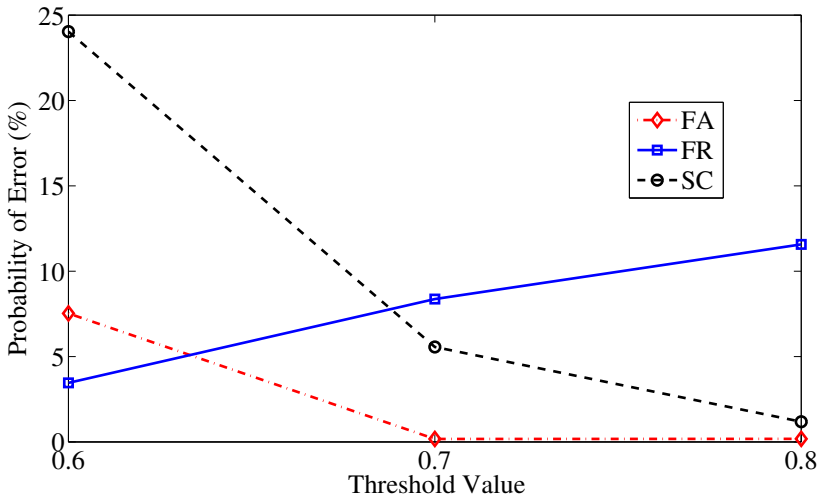

Figure A.2: *Tradeoff of FA, FR and SC errors with different threshold values*

equals to 0.8 on all the test channels. This will be our baseline to be compared with later after multi channel combination approaches are applied.

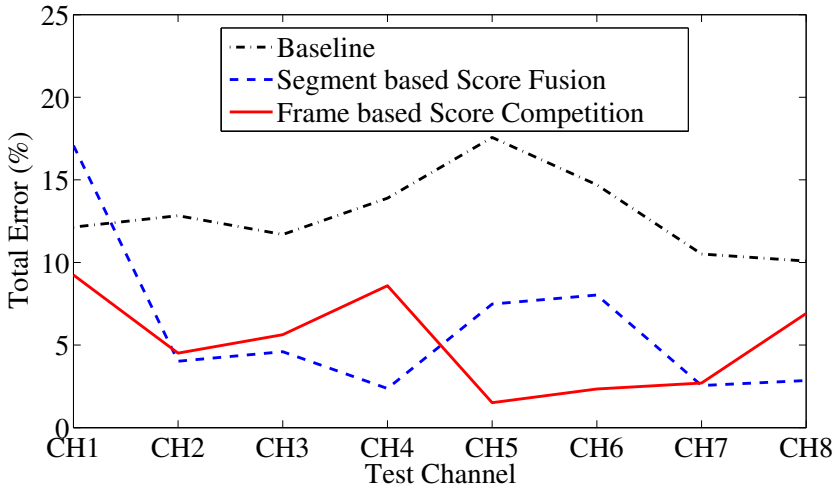

Figure A.3: *Total errors with different threshold under mismatched condition*

Table A.1: *Average performance under mismatched condition with threshold=0.8*

Test Channel	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8
FA	0.63	0.19	0.06	0.43	0.00	0.03	0.00	0.06
FR	9.41	12.34	10.67	11.94	16.95	12.07	9.35	9.83
SC	2.09	0.30	0.97	1.52	0.62	2.61	1.16	0.20
Total Error	**12.13**	**12.83**	**11.70**	**13.89**	**17.57**	**14.71**	**10.51**	**10.09**

A.1.3 Multiple Channel Combination

We applied the multiple channel combination approaches "Segment based Score Fusion" and "Frame based Score Competition" as described in chapter 2 in our open-set speaker identification experiment under mismatched conditions with our best threshold value, which equals to 0.8. Table A.2 and A.3 present the detailed open-set speaker identification performance with "Segment based Score Fusion" and "Frame based Score Competition" approaches applied on the baseline system respectively. Figure A.4 compares the total errors of the baseline system with the systems where each of the two multiple channel combination approaches is applied. No significant improvement is seen on most of the channels by the "Segment based Score Fusion" approach. On average "Frame based Score Competition" approach gains although it loses on one of the channels. We can see both approaches significantly reduces the SC error. This matches our expectation because we already see significant improvement by these approaches for closed-set speaker identification as presented in chapter 2, while the first stage of our open-set speaker identification, which can be considered as closed-set speaker identification, is responsible for generating the SC errors.

Table A.2: *Open-set speaker identification performance with Segment based Score Fusion*

Test Channel	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8
FA	7.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FR	3.01	4.02	4.60	2.37	7.48	8.03	2.55	2.85
SC	6.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Total Error	**17.08**	**4.02**	**4.60**	**2.37**	**7.48**	**8.03**	**2.55**	**2.85**

Table A.3: *Open-set speaker identification performance with Frame based Score Competition*

Test Channel	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8
FA	9.24	4.51	5.63	7.35	0.37	2.34	2.70	6.91
FR	0.00	0.00	0.00	1.24	0.76	0.00	0.00	0.00
SC	0.00	0.00	0.00	0.00	0.38	0.00	0.00	0.00
Total Error	**9.24**	**4.51**	**5.63**	**8.59**	**1.51**	**2.34**	**2.70**	**6.91**

A.2 Summary

In this section, we show how our system performs in the open-set scenario. Our focus is not to explore new strategies for open-set speaker recognition, but to test our system in the open-set situation. We observe that there is trade off between the three types of errors in the open-set speaker identification. The equal error rate for our system is about 3%. "Segment based Score Fusion" and "Frame based Score Competition" multi channel combination approaches help the performance differently. They both reduce the "Speaker Confusion (SC)" error significantly, the former increases the "False Rejection (FR)" error rate and the later increases the "False

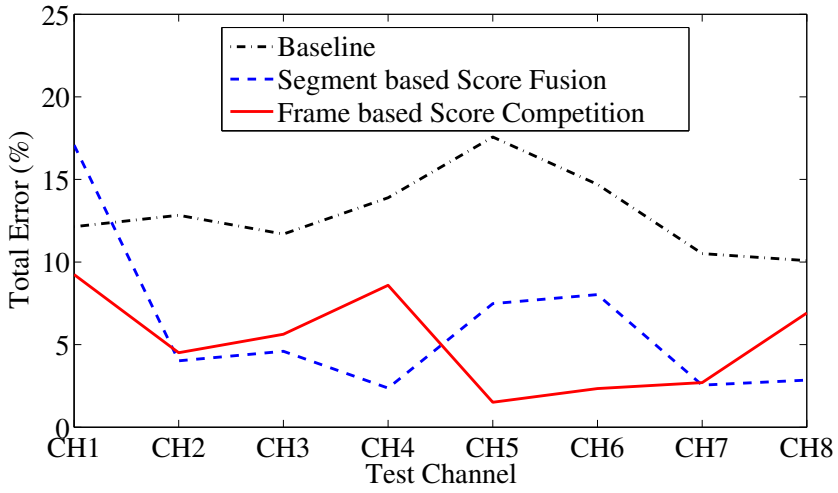

Figure A.4: *Performance comparison of multi channel combination vs baseline*

Alarm (FA)" error rate. The "Frame based Score Competition" approach reduce the total error rate on most of the channels.

Appendix B

Application of PSR to Other Tasks

We apply the phonetic speaker identification approaches on other classification tasks as well. In this section, we present the work of accent identification and language identification with phonetic speaker recognition approaches.

B.1 Accent Identification

In this section we apply our LSPM-pp approach to accent identification. In the first experiment, we use the LSPM-pp approach to differentiate between native and non-native speakers of English. The non-native English speaker set contains native speakers of Japanese with varying English proficiency levels. Each speaker was recorded reading several news articles aloud. Training and test sets are disjoint with respect to articles as well as speakers. The data used for this experiment is shown in B.1

We used 6 of the GlobalPhone phone recognizers in language $\{CH, DE, FR, JA, KR, PO, SP\}$. On the test set of 303 utterances, this approach achieves an accuracy of 97.7%.

In the second experiment, we attempt to further classify non-native utterances according to

Table B.1: *Number of speakers, total number of utterances, total length of audio for native and non-native classes*

		native	non-native
n_{spk}	training	3	7
	testing	2	5
$\sum n_{\text{utt}}$	training	318	680
	testing	93	210
$\sum \tau_{\text{utt}}$	training	23.1 min	83.9 min
	testing	7.1 min	33.8 min

proficiency level. The original non-native data was labeled with the proficiency of each speaker on the basis of a standardized evaluation procedure conducted by trained proficiency raters [2]. All speakers received a floating point grade between 0 and 3, with a grade of 4 reserved for native speakers. The distribution of non-native training speaker proficiencies shows that they fall into roughly three groups and we create three corresponding classes for our new discrimination task. Class 1 represents the lowest proficiency speakers, class 2 contains intermediate speakers, and class 3 contains the high proficiency speakers. Table B.2 shows our division of data.

With the LSPM-pp approach, we achieve accuracy of 61% for this 3-way proficiency classification task. This result indicates that discriminating among proficiency levels is a more difficult problem than discriminating between native and non-native speakers. The proficiency classification task attempts to determine the class of an utterance in a space that varies continuously according to the English proficiency of its speaker. While classification of native and non-native speakers can be described as identifying speakers who are clustered at the far ends of this proficiency axis.

Overall, the phonetic approach worked well for classifying utterances from speaker proficiency classes that were sufficiently separable. Like the other applications of this approach,

Table B.2: *Number of speakers, total number of utterances, total length of audio and average speaker proficiency score per proficiency class*

		class 1	class 2	class 3
n_{spk}	training	3	12	4
	testing	1	5	1
$\sum n_{\text{utt}}$	training	146	564	373
	testing	78	477	124
$\sum \tau_{\text{utt}}$	training	23.9 min	82.5 min	40.4 min
	testing	13.8 min	59.0 min	13.5 min
ave. prof	training	1.33	2.00	2.89
	testing	1.33	2.00	2.89

accent identification requires no hand-transcription and could easily be ported to test languages other than English/Japanese.

B.2 Language Identification

In this section, we apply the LSPM-pp approach to the problem of classification of four languages: Japanese (JA), Russian (RU), Spanish (SP) and Turkish (TU).

We employed a small number of phone recognizers in languages other than the four classification languages to demonstrate a degree of language independence which holds even in the language identification domain. Phone recognizers in Chinese (CH), German (DE) and French (FR), with phone vocabulary sizes of 145, 47 and 42 respectively, were borrowed from the GlobalPhone project as discussed in [104].

The data for this classification experiment, also borrowed from the GlobalPhone project but

not used in training the phone recognizers, was divided up as shown in Table B.3. Data set 1 was used for training the phonetic models, while data set 4 was completely held-out during training and used to evaluate the end-to-end performance of the complete classifier. Data sets 2 and 3 were used as development sets while experimenting with different decision strategies.

Table B.3: *Number of speakers per data set, total number of utterances and total length of audio per language*

	Set	JA	RU	SP	TU
n_{spk}	1	20	20	20	20
	2	5	10	9	10
	3	3	5	5	5
	4	3	5	4	5
$\sum n_{\text{utt}}$	all	2294	4923	2724	2924
$\sum \tau_{\text{utt}}$	all	6 hrs	9 hrs	8 hrs	7 hrs

We achieve 94.01%, 97.57%, 98.96% and 99.31% accuracy on 5s, 10s, 20s and 30s test durations respectively.

The phonetic language identification technique was also applied in our Mandarin Broadcast News system for the RT-04f (Rich Transcription) evaluation [127]. We have observed a number of foreign language segments, mostly English, in several Chinese news shows. As they cause high insertion errors for our Mandarin ASR system, it is beneficial to detect and discard them. The phonetic language identification technique is used to classify English from Chinese.

Table B.4 shows the effect of language identification on speech recognition performance on the RT04 evaluation development data set. We can clearly see big gains by rejecting English segments from the ASR output.

Table B.4: *Character Error Rate (CER) on development data set*

	RT03	Dev04
before Language Identification	5.9%	18.4%
after Language Identification	5.2%	16.6%

B.3 Summary

We applied the same techniques used in phonetic speaker recognition to other non-verbal cues recognition tasks including accent identification and language identification. Our classification framework performed equally well in the domains of accent and language identification. We achieved 97.7% discrimination accuracy between native and non-native English speakers. For language identification, we obtained 95.5% classification accuracy for utterances 5 seconds in length and up to 99.89% on longer utterances. The phonetic language identification technique was one component in our Mandarin Broadcast News system for the RT-04f Rich Transcription evaluation. It brought significant gains to the over all system performance.

Bibliography

[1] A. Adami, R. Mihaescu, D. A. Reynolds, and J. Godfrey. Modeling Prosodic Dynamics for Speaker Recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, China, 2003.

[2] H. Akaike. A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, 19:716–723, 1974.

[3] D. Reynolds amd P. Torres and R. Roy. EARS RT-03 Diarization. In *NIST RT-03S Workshop*, Boston, MA, 2003. http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/index.htm/EARSRT03SDiarization.pdf.

[4] W. Andrews, M. Kohler, J. Campbell, J. Godfrey, and J. Hernandez-Cordero. Gender-Dependent Phonetic Refraction for Speaker Recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 149–152, Orlando, USA, May 2002.

[5] B. S. Atal. Effectiveness of Linear Prediction Characteristics of the Speech Wave for Automatic Speaker Identification. *Journal of The American Statistical Association*, 55:1304–1312, 1974.

[6] B. S. Atal. Automatic Recognition of Speakers from Their Voices. *Proceedings of the IEEE*, 64:460–475, 1976.

[7] R. Bakis, S. Chen, P. Gopalakrishnan, R. Gopinath, S. Maes, and L. Polymenakos. Transcription of Broadcast News Shows with the IBM Large Vocabulary Speech Recognition System. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Munich, Germany, April 1997.

[8] F. Bimbot and L. Mathan. Text-free Speaker Recognition Using an Arithmetic Harmonic Sphericity Measure. In *Proceedings of Eurospeech*, Berlin, Germany, 1993.

[9] S. F. Boll. Suppression of Acoustic Noise in Speech Using Spectral Subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27:113–120, 1979.

[10] J. F. Bonastre, P. Delacourt, C. Fredouille, T. Merlin, and C. Wellekens. A Speaker Tracking System based on Speaker Turn Detection for NIST Evaluations. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Istanbul, Turkey, June 2000.

[11] S. Burger, V. MacLaren, and H. Yu. The ISL Meeting Corpus: The Impact of Meeting Type on Speech Style. In *Proceedings of International Conference on Spoken Language Processing*, Denver, USA, 2002.

[12] S. Burger and Z. Sloane. The ISL Meeting Corpus: Categorical Features of Communicative Group Interactions. In *Proceedings of NIST Meeting Recognition Workshop*, Montreal, Canada, May 2004.

[13] J. P. Campbell. Speaker Recognition: A Tutorial. *Proceeding of the IEEE*, 85:1437–1462, September 1997.

[14] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. T. Leek. Phonetic Speaker Recognition with Support Vector Machines. In *Neural Informatio Processing Systems (NIPS)*, Vancouver, Canada, 2003.

[15] J. Casas, R. Stiefelhagen, and et. al. Multi-camera/Multi-microphone System Design for Continuous room Monitoring. In *CHIL-WP4-D4.1-V2.1-2004-07-08-CO, CHIL Consortium Deliverable D4.1*, July 2004.

[16] S. Chen and P.S. Gopalakrishnan. Speaker, Environment and Channel Change Detection and Clustering via Bayesian Information Criterion. In *DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA, February 1998.

[17] S. S. Chen and P. S. Gopalakrishnan. Clustering via the Bayesian Information Criterion with Applications in Speech Recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Seattle, USA, 1998.

[18] CHIL Technology Evaluation January 2005. http://chil.server.de/servlet/is/1334/CHIL-EvalData-Overview-v04-2004-10-29.pdf.

[19] P. Clarkson and R. Rosenfeld. Statistical Language Modeling Using the CMU-Cambridge Toolkit. In *Proceedings of Eurospeech*, Rhodes, Greece, September 1997.

[20] Computers in the Human Interaction Loop - CHIL. http://chil.server.de .

[21] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, New York, 1991.

[22] S. B. Davis and P. Mermelstein. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Utterances. *IEEE Transactions on Speech and Audio Processing*, 28:357–366, 1980.

[23] P. Delacourt and C. Wellekens. Audio Data Indexing: Use of Second-order Statistics for Speaker-based Segmentation. In *Proceedings of International Conference of Multimedia Computing and Systems*, Florence, Italy, June 1999.

[24] P. Delacourt and C. J. Wellekens. DISTBIC: A Speaker-based Segmentation for Audio Data Indexing. *Speech Communications*, 32:111–126, 2000.

[25] T. Dietterich. Machine Learning Research: Four Current Directions. *AI Magazine*, 18:97–136, 1998.

[26] J. Dines, J. Vepa, and T. Hain. The Segmentation of Multi-channel Meeting Recordings for Automatic Speech Recognition. In *Proceedings of International Conference on Spoken Language Processing*, Pittsburgh, USA, 2006.

[27] G. Doddington, M. Przybycki, A. Martin, and D. Reynolds. The NIST Speaker Recognition Evaluation—Overview, Methodology, Systems, Results, Perspective. *Speech Communication*, 31:225–254, 2000.

[28] George Doddington. Speaker Recognition Based on Idiolectal Differences between Speakers. In *Proceedings of Eurospeech*, Aalborg, Denmark, September 2001.

[29] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley Interscience, 2nd edition, 2000.

[30] A.W.F. Edwards. *Likelihood*. Cambridge University Press, 1972.

[31] H. K. Ekenel and Q. Jin. ISL Person Identification Systems in the CLEAR Evaluations. In *CLEAR Evaluation Workshop*, Southampton, UK, 2006.

[32] N. Fox, R. Gross, P. Chazal, J. Cohn, and R. Reilly. Person Identification Using Automatic Integration of Speech, Lip and Face Experts. In *ACM SIGMM 2003 Multimedia Biometrics Methods and Applications Workshop*, Berkeley, CA, 2003.

[33] S. Furui. Cepstral Analysis Technique for Automatic Speaker Verification. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29:254–272, 1981.

[34] S. Furui. An Overview of Speaker Recognition Technology. In *Proceeding of Workshop on Automatic Speaker Recognition, Identification and Verification*, Martigny, Switzerland, 1994.

[35] S. Furui. Recent Advances in Speaker Recognition. *Pattern Recognition Letters*, 18:859–872, 1997.

[36] M. Gales and S. Young. Robust Speech Recognition in Additive and Convolutional Noise using Parallel Model Combination. *Computer Speech and Language*, 9:289–307, 1995.

[37] J. Gauvain and C. Barras. Speaker Diarization. In *NIST RT-03S Workshop*, Boston, MA, 2003.
http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/limsi_rt0503spkrvg.pdf.

[38] H. Gish, M. Krasner, W. Russell, and J. Wolf. Methods and Experiments for Text-independent Speaker Recognition over Telephone Channels. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 865–868, Tokyo, Japan, 1986.

[39] H. Gish and M. Schmit. Text-Independent Speaker Identification. In *IEEE Signal Processing Magazine*, October 1994.

[40] H. Gish, M. Siu, and R. Rohlicek. Segmentation of Speech for Speech Recognition and Speaker Identification. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toronto, Canada, April 1991.

[41] J. Gonzalez-Rodriguez, J. Ortega-Garcia, C. Martin, and L. Hernandez. Increasing Robustness in GMM Speaker Recognition Systems for Noisy and Reverberant Speech with Low Complexity Microphone Arrays. In *Proceedings of International Conference on Spoken Language Processing*, Philadelphia, USA, 1996.

[42] T. Hain, S. Johnson, A. Tuerk, P. Woodland, and S. Young. Segment Generation and Clustering in the HTK Broadcast News Transcription System. In *DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA, February 1998.

[43] A. O. Hatch, B. Peskin, and A. Stolcke. Improved Phonetic Speaker Recognition Using Lattice Decoding. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, USA, 2005.

[44] L. Heck. Integrating High Level Information. In *SuperSID project at JHU Summer Workshop*, 2002. http://www.clsp.jhu.edu/ws2002/groups/supersid/.

[45] H. Hermansky. Perceptual Linear Predictive (PLP) Analysis of Speech. *Journal of the Acoustical Society of America*, 87:1738–1752, 1990.

[46] H. Hermansky and N. Morgan. RASTA Processing of Speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589, 1994.

[47] M. Hunt. Further Experiments in Text-independent Speaker Recognition over Communications Channels. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 563–566, Boston, USA, 1983.

[48] M. Indovina, U. Uludag, R. Snelik, A. Mink, and A. Jain. Multimodal Biometric Authentication Methods: A COTS Approach. In *ACM SIGMM 2003 Multimedia Biometrics Methods and Applications Workshop*, Berkeley, CA, 2003.

[49] A. Janin, J. Ang, S. Bhagat, R. Dhillon, J. Edwards, N. Morgan, B. Peskin, E. Shriberg, A. Stolcke, C. Wooters, and B. Wrede. The ICSI Meeting Project: Resources and Research. In *Proceedings of NIST Meeting Recognition Workshop*, Montreal, Canada, May 2004.

[50] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. The ICSI Meeting Corpus. In *Proceedings*

of IEEE International Conference on Acoustics, Speech, and Signal Processing, Hong Kong, China, 2003.

[51] H. Jin, F. Kubala, and R. Schwartz. Automatic Speaker Clustering. In *DARPA Broadcast News Transcription and Understanding Workshop*, pages 108–111, Chantilly, VA, 1997.

[52] Q. Jin, J. Navratil, D. Reynolds, W. Andrews, J. Campbell, and J. Abramson. Combining Cross-stream and Time Dimensions in Phonetic Speaker Recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, China, 2003.

[53] Q. Jin, Y. Pan, and T. Schultz. Far-field Speaker Recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, 2006.

[54] Q. Jin, T. Schultz, and A. Waibel. Speaker Identification Using Multilingual Phone Strings. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Orlando, USA, May 2002.

[55] S. Johnson and P. Woodloand. Speaker Clustering using Direct Maximization of the MLLR Adapted Likelihood. In *Proceedings of International Conference on Spoken Language Processing*, Sydney, Australia, 1998.

[56] S. E. Johnson. Who Spoke When? - Automatic Segmentation and Clustering for Determining Speaker Turns. In *Proceedings of Eurospeech*, Budapest, Hungary, September 1999.

[57] T. Kemp, M. Schmidt, M. Westphal, and A. Waibel. Strategies for Automatic Segmentation of Audio Data. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Istanbul, Turkey, June 2000.

[58] L. G. Kersta. Voiceprint Identification. *Nature*, 196:1253–1257, 1962.

[59] D. Klusacek, J. Navratil, D. Reynolds, and J. Campbell. Conditional Pronunciation Modeling in Speaker Detection. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, China, 2003.

[60] M. Kohler, W. Andrews, J. Compbell, and J. Hernandez-Cordero. Phonetic Refraction for Speaker Recognition. In *Proceedings of Workshop on Multilingual Speech and Language Processing*, Aalborg, Denmark, 2001.

[61] F. Kubala, H. Jin, S. Matsoukas, L. Nguyen, R. Schwartz, and J. Makhoul. The 1996 BBN Byblos Hub-4 Transcription System. In *DARPA Broadcast News Transcription and Understanding Workshop*, pages 90–93, Chantilly, VA, 1997.

[62] D. V. Lancker, J. Kreiman, and T. D. Wickens. Familiar Voice Recognition: Patterns and parameters, Part I: Recognition of Backward Voices. *Journal of Phonetics*, 13:19–38, 1985.

[63] D. V. Lancker, J. Kreiman, and T. D. Wickens. Familiar Voice Recognition: Patterns and Parameters, Part II: Recognition of Rate-altered Voices. *Journal of Phonetics*, 13:39–52, 1985.

[64] I. Lapidot. SOM as Likelihood Estimator for Speaker Clustering. In *Proceedings of Eurospeech*, Geneva, Switzerland, September 2003.

[65] K. Laskowski, Q. Jin, and T. Schultz. Crosscorrelation-based Multispeaker Speech Activity Detection. In *Proceedings of International Conference on Spoken Language Processing*, Jeju Island, South Korea, 2004.

[66] Y. Lavner, I. Gath, and J. Rosenhouse. The Effects of Acoustic Modifications on the Identification of Familiar Voices Speaking Isolated Vowels. *Speech Communication*, 30:9–26, 2000.

[67] Q. Lin, E. Jan, and J. Flanagan. Microphone Arrays and Speaker Identification. *IEEE Transactions on Speech and Audio Processing*, 2:622–629, 1994.

[68] J. Lopez and D. Ellis. Using Acoustic Condition Clustering to Improve Acoustic Change Detection on Broadcast News. In *Proceedings of International Conference on Spoken Language Processing*, Beijing, China, October 2000.

[69] S. Lucey and T. Chen. Improved Audio-visual Speaker Recognition via the Use of a Hybrid Combination Strategy. In *The 4th International Conference on Audio and Video Based Biometric Person Authentication*, Guilford, U.K., 2003.

[70] J. Makhoul. Linear Prediction: A Tutorial Review. *Proceeding of the IEEE*, 63:561–580, 1975.

[71] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET Curve in Assesment of Detection Task Performance. In *Proceedings of Eurospeech*, Rhodes, Greece, 1997.

[72] S. Matsoukas, R. Iyer, O. Kimball, J. Ma, and T. Colthurst. BBN CTS English System. In *NIST RT-03 Workshop*, Boston, MA, 2003.

[73] F. Metze, Q. Jin, C. Fúgen, K. Laskowski, Y. Pan, and T. Schultz. Issues in Meeting Transcription — The ISL Meeting Transcription System. In *Proceedings of NIST Meeting Recognition Workshop*, Montreal, Canada, May 2004.

[74] D. Moraru, S. Meifnier, C. Fredouille, and J. Bonastre. ELISA, CLIPS and LIA NIST 2003 Segmentation. In *NIST RT-03S Workshop*, Boston, MA, 2003. http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/Nist2003-LIA-CLIPS-segv2.pdf.

[75] N. Morgan, D. Baron, J. Edwards, D. Ellis, D. Gelbart, A. Janin T. Pfau, E. Shriberg, and A. Stolcke. The Meeting Project at ICSI. In *Proceedings of Human Language Technology Conference*, San Diego, March 2001.

[76] K. Mori and S. Nakagawa. Speaker Change Detection and Speaker Clustering Using VQ Distortion for Broadcast News Speech Recognition. In *International Conference on Pattern Recognition (ICPR)*, Quebec, Canada, 2002.

[77] J. Navratil, Q. Jin, W. Andrews, and J. Campbell. Phonetic Speaker Recognition Using Maximum-Likelihood Binary-decision Tree Models. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, China, 2003.

[78] P. Nguyen. PSTL's Speaker Diarization. In *NIST RT-03S Workshop*, Boston, MA, 2003. http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/isometspkr.pdf.

[79] NIST 1999 Speaker Recognition Evaluation Plan. http://www.itl.nist.gov/iaui/894.01/spk99/spk99plan.html.

[80] NIST Annual Speaker Recognition Evaluation. http://www.nist.gov/speech/tests/spk/index.htm.

[81] F. Nolan. The Phonetic Bases of Speaker Recognition. In *Cambridge University Press*, 1983.

[82] Y. Pan. *Robust Speech Recognition on Distant Microphones*. Carnegie Mellon University, thesis in submission edition, 2007.

[83] J. Pelecanos and S. Sridharan. Feature Warping for Robust Speaker Verification. In *Proceedings of Speaker Odyssey Conference*, Crete, Greece, 2001.

[84] A. Pentland, B. Moghaddam, T. Starner, and M. Turk. View based and Modular Eigenspaces for Face Recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 84–91, Seattle, USA, 1994.

[85] T. Pfau, D. Ellis, and A. Stolcke. Multispeaker Speech Activity Detection for the ICSI Meeting Recognizer. In *Proceedings of Automatic Speech Recognition and Understanding Workshop*, Madonna di Campiglio, Italy, 2001.

[86] M. Przybocki and A. Martin. The NIST Year 2001 Speaker Recognition Evaluation Plan. In *NIST Speaker Recognition Evaluation*, 2001. http://www.nist.gov/speech/tests/spk/2001/doc/.

[87] T. F. Quatieri, D. A. Reynolds, and G. C. O'Leary. Estimation of Handset Nonlinearity with Application to Speaker Recognition. *IEEE Transactions on Speech and Audio Processing*, pages 567–584, 2000.

[88] L. R. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, NJ, 1993.

[89] B. Ramabhadran, J. Huang, U. Chaudhari, G. Iyengar, and H. Nock. Impact of Audio Segmentation and Segment Clustering on Automated Transcription Accuracy of Large Spoken Archives. In *Proceedings of Eurospeech*, Geneva, Switzerland, September 2003.

[90] D. Reynolds, W. Andrews, J. Campbell, J. Navratil, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Mihaescu, J. Godfrey, D. Jones, and B. Xiang. Exploiting High-level Information for High-accuracy Speaker Recognition. In *SuperSID Project Final Report*, 2002. http://www.clsp.jhu.edu/ws2002/groups/supersid/.

[91] D. A. Reynolds. An Overview of Automatic Speaker Recognition Technology. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Orlando, USA, 2002.

[92] D. A. Reynolds. Automatic Speaker Recognition: Acoustics and Beyond. In *SuperSID project at JHU Summer Workshop*, 2002. http://www.clsp.jhu.edu/ws2002/groups/supersid/.

[93] D. A. Reynolds, W. Andrews, J. Campbell, J. Navratil, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Mihaescu, J. Godfrey, D. Jones, and B. Xiang. The SuperSID Project: Exploiting High-level Information for High-accuracy Speaker Recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, China, 2003.

[94] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker Verification Using Adapted Gaussian Mixture Models. In *Digital Signal Processing Review Journal*, January 2000.

[95] D. A. Reynolds and R. C. Rose. Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models. In *IEEE Transactions on Speech and Audio Processing*, pages 72–83, 1995.

[96] Rich Transcription 2004 Spring Meeting Recognition Evaluation. http://www.itl.nist.gov/iad/894.01/tests/rt/rt2004/spring/.

[97] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific, Singapore, 1989.

[98] A. Rosenberg, A. Gorin, Z. Liu, and S. Parthasarathy. Unsupervised Speaker Segmentation of Telephone Conversations. In *Proceedings of International Conference on Spoken Language Processing*, Denver, USA, 2002.

[99] A. E. Rosenberg, J. DeLong, C. H. Lee, B. H. Juang, and F. K. Soong. The Use of Cohort Normalized Scores for Speaker Verification. In *Proceedings of International Conference on Spoken Language Processing*, Banff, Canada, 1992.

[100] A. E. Rosenberg and S. Parthasarathy. Speaker Background Models for Connected Digit Password Speaker Verification. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, USA, 1996.

[101] A. Schmidt-Nielsen and T. H. Crystal. Human vs. machine speaker identification with telephone speech. In *Proceedings of International Conference on Spoken Language Processing*, Sydney, Australia, October 1998.

[102] A. Schmidt-Nielsen and T. H. Crystal. Speaker Verification by Human Listeners: Experiments Comparing Human and Machine Performance Using the NIST 1998 Speaker Evaluation Data. *Digital Signal Processing*, pages 249–266, 2000.

[103] T. Schultz, Q. Jin, K. Laskowski, A. Tribble, and A. Waibel. Improvements in Non-verbal Cue Identification using Multilingual Phone Strings. In *Proceeding of the Speech-to-Speech Translation Workshop on the 40th Anniversary Meeting of the Association for Computational Linguistics*, Philadelphia, USA, 2002.

[104] T. Schultz and K. Kirchhoff. *Multilingual Speech Processing*. Elsevier, Academic Press, 1st edition, 2006.

[105] T. Schultz and A. Waibel. Language Independent and Language Adaptive Acoustic Modeling for Speech Recognition. *Speech Communication*, 35:31–51, 2001.

[106] G. Schwarz. Estimating the Dimension of a Model. *Annals of Statistics*, 6:461–464, 1978.

[107] E. Shriberg, A. Stolcke, and D. Baron. Observations on Overlap: Findings and Implications for Automatic Processing of Multi-party Conversation. In *Proceedings of Eurospeech*, Aalborg, Denmark, 2001.

[108] M. Siegler, U. Jain, B. Raj, and R. Stern. Automatic Segmentation, Classification and Clustering of Broadcast News Audio. In *DARPA Broadcast News Transcription and Understanding Workshop*, Chantilly, VA, 1997.

[109] A. Solomonoff, A. Mielke, M. Schmidt, and H. Gish. Clustering Speakers by Their Voices. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Seattle, USA, 1998.

[110] V. Stanford and J. Garofolo. Beyond Close-talk — Issues in Distant Speech Acquisition, Conditioning Classification, and Recognition. In *Proceedings of NIST Meeting Recognition Workshop*, Montreal, Canada, May 2004.

[111] S. Strassel and M. Glenn. Shared Linguistic Resources for Human Language Technology in the Meeting Domain. In *Proceedings of NIST Meeting Recognition Workshop*, Montreal, Canada, May 2004.

[112] S. Stüker, C. Fügen, R. Hsiao, S. Ikbal, Q. Jin, F. Kraft, M. Paulik, M. Raab, Y. Tam, and M. Wölfel. The isl tc-star spring 2006 asr evaluation systems. In *TC-STAR Workshop on Speech-to-Speech Translation*, pages 139–144, Barcelona, Spain, June 2006.

[113] M. Sugiyama, J. Murakami, and H. Watanabe. Speech Segmentation and Clustering Based on Speaker Features. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Minneapolis, USA, 1993.

[114] SuperSID Project. http://www.clsp.jhu.edu/ws2002/groups/supersid/.

[115] The Rich Transcription Spring 2003 Evaluation Plan. http://www.nist.gov/speech/tests/rt/rt2003/spring/docs/rt03-spring-eval-plan-v4.pdf.

[116] S. Tranter, K. Yu, and the HTK STT team. Diarization for RT-03s at Cambridge University. In *NIST RT-03S Workshop*, Boston, MA, 2003. http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/tranter_rt03sdiary.2up.pdf.

[117] A. Tritschler and R. Gopinath. Improved Speaker Segmentation and Segments Clustering using the Bayesian Information Criterion. In *Proceedings of Eurospeech*, Budapest, Hungary, September 1999.

[118] A. Vandecatseye and J. Martens. A Fast, Accurate Stream-based Speaker Segmentation and Clustering Algorithm. In *Proceedings of Eurospeech*, Geneva, Switzerland, 2003.

[119] A. Vinciarelli. Sociometry based Multiparty Audio Recordings Segmentation. In *In Proceedings of the International Conference on Multimedia and Expo*, Toronto, Canada, 2006.

[120] A. Waibel, H. Steusloff, R. Stiefelhagen, and et. al. CHIL: Computers in the Human Interaction Loop. In *5th International Workshop on Image Analysis for Multimedia Interactive Services(WIAMIS)*, Lisbon, Portugal, 2004.

[121] Yuan Wang, Yunhong Wang, and T. Tan. Combining Fingerprint and Voiceprint Biometrics for Identity Verification: an Experimental Comparison. In *First International Conference on Biometric Authentication*, Hong Kong, China, 2004.

[122] J. J. Wolf. Efficient Acoustic Parameters for Speaker Recognition. *Journal of The American Statistical Association*, 51:2044–2056, 1972.

[123] P. Woodland, G. Evermann, M. Gales, T. Hain, R. Chan, and B. Jia. CUHTK STT System for RT-03. In *NIST RT-03 Workshop*, Boston, MA, 2003.

[124] P. Woodland, M. Gales, D. Pye, and S. Young. The Development of the 1996 HTK Broadcast News Transcription System. In *DARPA Broadcast News Transcription and Understanding Workshop*, Chantilly, VA, 1997.

[125] B. Xiang, U. Chaudhari, J. Navratil, G. Ramaswamy, and R. Gopinath. Short-time Gaussianization for Robust Speaker Verification. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Orlando, USA, 2002.

[126] K. Yao, K. K. Paliwal, and S. Nakamura. Model-based Noisy Speech Recognition with Environment Parameters Estimated by Noise Adaptive Speech Recognition with Prior. In *Proceedings of Eurospeech*, Geneva, Switzerland, 2003.

[127] H. Yu, Y. Tam, T. Schaaf, S. Stúker, Q. Jin, M. Noamany, and T. Schultz. The ISL RT04 Mandarin Broadcast News Evaluation System. In *EARS Rich Transcription Workshop*, Palisades, NY, November 2004.

[128] B. Zhou and J. Hansen. Unsupervised Audio Stream Segmentation and Clustering via the Bayesian Information Criterion. In *Proceedings of International Conference on Spoken Language Processing*, Beijing, China, October 2000.