Spectral Features for Automatic Text-Independent Speaker Recognition

Tomi Kinnunen
University of Eastern Finland

Spectral Features for Automatic Text-Independent Speaker Recognition

Tomi Kinnunen

LICENTIATE'S THESIS

University of Joensuu
Department of Computer Science
P.O. Box 111, FIN-80101 Joensuu, Finland

December 21, 2003

JOENSUUN
YLIOPISTO

Abstract

Front-end or *feature extractor* is the first component in an automatic speaker recognition system. Feature extraction transforms the raw speech signal into a compact but effective representation that is more stable and discriminative than the original signal. Since the front-end is the first component in the chain, the quality of the later components (*speaker modeling* and *pattern matching*) is strongly determined by the quality of the front-end. In other words, classification can be at most as accurate as the features.

Several feature extraction methods have been proposed, and successfully exploited in the speaker recognition task. However, almost exclusively, the methods are adopted directly from the *speech recognition* task. This is somewhat ironical, considering the opposite nature of the two tasks. In speech recognition, speaker variability is one of the major error sources, whereas in speaker recognition it is the information that we wish to extract. The mel-frequency cepstral coefficients (MFCC) is the most evident example of a feature set that is extensively used in speaker recognition, but originally developed for speech recognition purposes. When MFCC front-end is used in speaker recognition system, one makes an implicit assumption that the human hearing meachanism is the optimal speaker recognizer. However, this has not been confirmed, and in fact opposite results exist.

Although several methods adopted from speech recognition have shown to work well in practise, they are often used as "black boxes" with fixed parameters. It is not understood what kind of information the features capture from the speech signal. Understanding the features *at some level* requires experience from specific areas such as speech physiology, acoustic phonetics, digital signal processing and statistical pattern recognition. According to the author's general impression of literature, it seems more and more that currently, at the best we are *guessing* what is the code in the signal that carries our individuality.

This thesis has two main purposes. On the one hand, we attempt to see the feature extraction as a whole, starting from understanding the speech production process, what is known about speaker individuality, and then going into the details of the feature extraction methods. Although prosodics and other high-level features have been recently exploited successfully in speaker recognition, our attention is on the low-level spectral features due to their widespread use, easy computation and modeling, and the "black box" effect associated with these. Particularly, attention is paid on subband processing, LPC parameters, cepstral processing and spectral dynamics (delta-features).

On the other hand, the second purpose of the thesis is to find out which of the several spectral features are best suited for automatic speaker recognition systems in terms of their reliability and computational efficiency. We aim to find what are the critical parameters that affect the performance and try to give some general guidelines about the analysis parameters. We conduct experiments on two speech corpora using vector quantization (VQ) speaker modeling. The corpora are a 100 speaker subset of the American English TIMIT corpus, and a Finnish corpus consisting of 110 speakers. Although noise robustness is an important issue in real applications, it is outside the scope of this thesis. Our main attempt is to gain at least some understanding what is individual in the speech spectrum.

Keywords: speaker individuality, feature extraction, spectral features, automatic speaker recognition

Preface

Writing a long monograph is not easy. Four years ago I finished my master's thesis, and although I had made a lot of work for that, later I felt that after all, it was quite fun. After that, however, I have worked myself on this research topic, and reviewing of other methods proposed by other authors is, after all, much more boring than inventing your own methods :-). During writing of this thesis, the topic has changed one time, my text editor has switched from MS Word to LaTeX, and I have rewritten most chapters 2-3 times. By the time of my writing, I have heard the following question N times: "Why do you write licentiate's thesis, it is not compulsory, why don't you do PhD thesis directly?".

That was the anti-acknowledgments part of this preface. Now comes the acknowledgments. Several person have contributed to my thesis either directly or undirectly, and I want to say "paljon kiitoksia!" ("thanks a lot!") to many people. Greatest thanks goes to my supervisor, prof. Pasi Fränti, who patiently read my text and gave me useful and constructive criticism. Prof. Stefan Werner has inspired me a lot with his positive attitude towards my research. Researcher Ismo Kärkkäinen has been an invaluable help in debugging my programs and helping with scripting, as well as pointing out me weird but good movies. Thanks for Ville Hautamäki for helping with my LaTeXproblems and long philosophical talks in cafeteria Houkutus and other places. Thanks for Meeri Parkkinen for giving useful feedback about my language, and thanks for Jussi Parkkinen for pointing out Meeri, and introducing me to the field of pattern recognition originally. Thanks for library assistant Irmeli Sajaniemi for kindly helping me in finding bibliography. Thanks for other CS department people as well, especially the office staff (Merja Hyttinen, Marja-Liisa Makkonen, Eeva Saukkonen), Juha Hakkarainen, and the rest of the speaker recognition research group. Thank for all of you, really! If I have forgotten you from the list and you should be there, I'm sorry about that.

The main funder of my work is Itä-Suomen tietotekniikan tutkijakoulu (East Finland Graduate School in Computer Science and Engineering, ECSE).

This thesis was also partially funded by a grant from Itä-Suomen Korkean Teknologian Säätiö. I would also like to thank prof. Antti Iivonen and his research group at the University of Helsinki for allowing me to use their speech data. Thanks for the inspectors of this thesis. Unfortunately, I am not yet sure who they will be because they first have to be ordered by the Faculty of Science ;-).

That was the official part of the acknowledgments. But what would be this world be without friends? I would like to thank all of my old and new friends as well as my parents. Thanks for being there, and thanks for understanding.

Merry Christmas and Happy New Year!

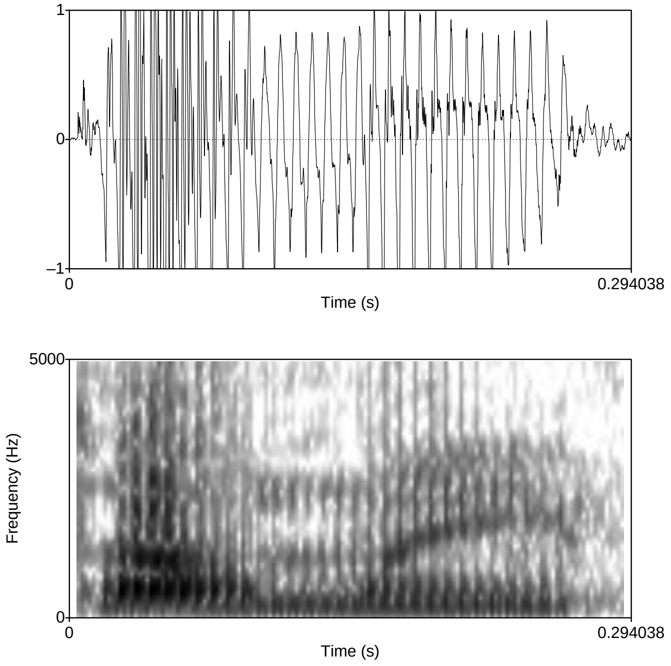

Figure 1: “Tomi”

Contents

Chapter 1

INTRODUCTION

It is not uncommon that you receive a phone call where the caller starts by saying “Hello, it is me”, and you reply immediately “Well, hi!”. You have recognized the caller from his/her voice only (supposing your telephone does not show the caller’s name). This is an example of naive speaker recognition that we perform in our everyday life.

There is an increasing need for person authentication in the world of information, applications ranging from credit card payments to border control and forensics. In general, a person can be authenticated in three different ways [128]:

1. Something the person *has*, e.g. a key or a credit card,

2. Something the person *knows*, e.g. a PIN number or a password,

3. Something the person *is*, e.g. signature, fingerprints, voice, facial features

The first two are traditional authentication methods that have been used several centuries. However, they have the shortcoming that the key or credit card can be stolen or lost, and the PIN number or password can be easily misused or forgotten. For the last class of authentication methods, known as *biometric person authentication* [15, 128, 78], these problems are lesser. Each person has unique anatomy, physiology and learned habits that familiar persons use in everyday life to recognize the person.

Increased computing power and decreased microchip size has given impetus for implementing realistic biometric authentication methods. The interest in biometric authentication has been increasing rapidly in the past few years. The topic of this thesis deals with our most natural way of communicating with each other, speech. *Speaker recognition* refers to task of recognizing peoples by their voices.

1.1 Applications

The main applications of speaker recognition technology include the following:

- Person authentication
- Forensics
- Speech recognition
- Multi-speaker environments
- Personalized user interfaces

Person authentication is the most obvious application of any biometric authentication technique. Speaker recognition could be used in credit card transactions as an authentication method combined with some others like face recognition. Alternatively, it could be used in computer login, a "key" to a physical facility, or in border control.

Forensics is an important application of speaker recognition. If there is a speech sample that was recorded during the commitment of a crime, the suspect's voice can be compared with this in order to give an indication of the similarity of the two voices. This topic is covered in detail in [139]. In Finland, about 50 requests related to forensic audio research are sent each year to the Crime Laboratory of the National Bureau of Investigation [111]. A considerable number of these are related to speaker recognition (see Table 1.1).

Table 1.1: Research requests sent to the Crime Laboratory of the National Bureau of Investigation related to forensic audio [111].

	1999	2000	2001	2002
Total # requests	38	40	45	51
Speaker recognition	24 (63 %)	13 (32 %)	13 (28 %)	32 (62 %)

Speech recognition, i.e. conversion from speech to text, has been actively studied since the 1950's, but there is not yet a universal speech recognition system that would work for unlimited vocabulary and for all speakers. Speech and speaker recognition are dual research areas in the sense that speaker variability is one of the major problems in speech recognition, whereas in speaker recognition it is an advantage. Speaker recognition technology could be used to reduce the speaker variability in speech recognition systems by *speaker adaptation* [81]. For instance, speech recognition system could have a "speaker gating" unit that recognizes who is speaking (see Fig. 1.1). Then, the system could adapt its speech recognizer parameters to suit better for the current speaker, or to select a speaker-dependent speech recognizer from its database.

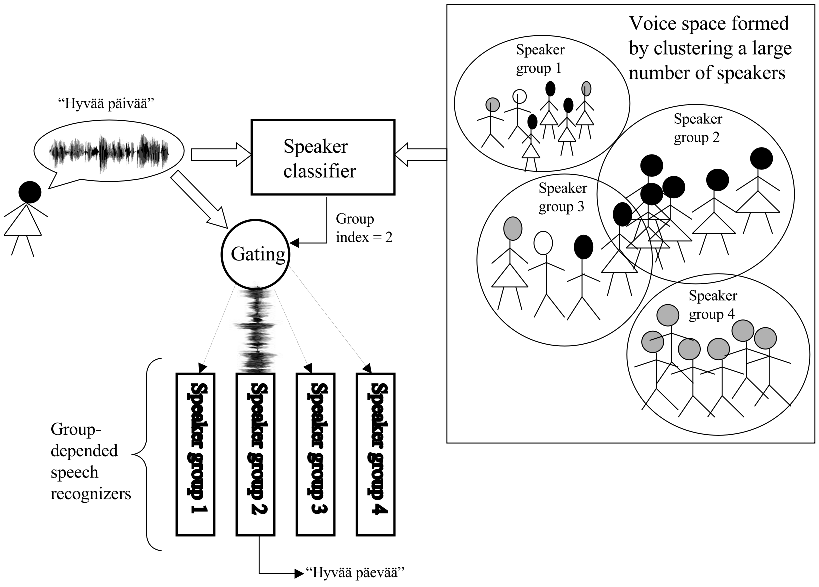

Figure 1.1: Applying speaker recognition in speech recognition.

In a *multi-speaker environment*, several speakers are included in the audio recording. Examples can be found everywhere: panel discussions, court room conversations, teleconferencing, TV and radio broadcasts. Speaker recognition technology might be very useful in application designed for multi-speaker environments. Three different multi-speaker tasks are recognized [102]: *speaker detection*, *speaker tracking*, and *speaker segmentation*. The detection task consist of deciding whether a known speaker is present in a multi-speaker recording. In the tracking task, a given speaker's speaking intervals are located in the recording. The segmentation task consists of locating the speech intervals of each different speaker. In the most general case, there might be no prior knowledge of the speakers or their number (see an example of a such system in [84]). Applications of speaker segmentation have been proposed for segmentation of news broadcasts [94, 82].

Finally, *speech user interfaces* such as voice-mail are becoming more and more popular due to the developments in speech technology in general. By recognizing the speaker, the system could adapt to his/her needs and preferences.

1.2 Pros and Cons of Speaker Recognition

The main advantage of speaker recognition is its naturalness. Speaking is our main communication matter, and embedding speaker recognition technology into applications is non-intrusive from the user's viewpoint. Another strong advantage are cheap costs; no special equipment is needed. In order to capture a speech signal, only a microphone is needed, as contrasted to fingerprint and retinal scanners, for instance. Signal processing and pattern matching algorithms for speaker recognition are low-cost and memory-efficient, and thus applicable for mobile devices. Last but not least, performance of automatic speaker recognition is considerably high in right conditions.

It has been demonstrated that integration of speech with other biometric authentication methods (*multi-modal person authentication*) improves overall recognition performance [19, 153]. In the latest AVBPA conference [78], speech was the third most popular biometric after face and fingerprint. In the same conference, nine different multimodal biometric authentication systems were introduced, and speaker recognition was included in six of them.

A common belief is that speaker recognition is an unreliable authentication method, and this is true to a certain extent. Persons "voice signature" simply is not as unique as, for instance, a fingerprint. The difference between fingerprint and voice is that the former is a *physical* biometric, which is directly measured from person's body. The voice, on the other hand, is more a *behavioral* biometric, which is a result of body part movements. The resulting speech wave is merely a reflection of the physical properties of the speech production organs. The articulatory movements and consequently the acoustic speech wave, are never exactly the same even when the same speaker produces the same utterance two times successively.

The most common argument that the author hears is referred via impersonators: person's voice can be easily imitated. However, this is based on our subjective impression only. Often the impersonator exaggerates certain person characteristics (and uses possibly also visual information to make a better impersonation). If the impersonated speaker has a "personal sounding" voice, human listeners tend to pay attention more to the exaggerated voice characteristics such as accent [166]. Speaker individuality consists of several different parameters that supplement each other, and human listeners probably use only a small subset of available cues. Based on our own subjective impression, we tend to think that speaker recognition technology is not reliable.

1.3 Elementary Concepts and Terminology

The most common characterization of automatic speaker recognition is the division into two different tasks: *speaker identification* and *speaker verification* tasks [20, 43].

1.3.1 Identification and Verification Tasks

In the identification task, or *1:N matching*, an unknown speaker is compared against a database of N known speakers, and the best matching speaker is returned as the recognition decision. The verification task, or *1:1 matching*, consists of making a decision whether a given voice sample is produced by a claimed speaker. An *identity claim* (e.g., a PIN code) is given to the system, and the unknown speaker's voice sample is compared against the claimed speaker's voice template. If the similarity degree between the voice sample and the template exceeds a predefined *decision threshold*, the speaker is accepted, and otherwise rejected.

Of the identification and verification tasks, identification task is generally considered more difficult. This is intuitive: when the number of registered speakers increases, the probability of an incorrect decision increases [28, 44, 128]. The performance of the verification task is not, at least in theory, affected by the population size since only two speakers are compared.

Open and Closed-Set Identification

Speaker identification task is further classified into *open*- and *closed-set* tasks. If the target speaker is assumed to be one of the registered speakers, the recognition task is a closed-set problem. If there is a possibility that the target speaker is none of the registered speakers, the task is called an *open-set* problem. In general, the open-set problem is much more challenging. In the closed-set task, the system makes a forced decision simply by choosing the best matching speaker from the speaker database - no matter how poor this speaker matches. However, in the case of open-set identification, the system must have a predefined tolerance level so that the similarity degree between the unknown speaker and the best matching speaker is within this tolerance. In this way, the verification task can be seen as a special case of the open-set identification task, with only one speaker in the database ($N = 1$).

Text-Independent and Text-Dependent Tasks

Speaker recognition tasks can be further classified into *text-dependent* or *text-independent* tasks. In the former case, the utterance presented to the recognizer is known beforehand. In the latter case, no assumptions about the text being spoken is made, but the system must model the general underlying properties of the speaker's vocal space.

In general, text-dependent systems are more accurate, since both the content and voice can be compared. For instance, a speech recognizer can be used in recognizing whether the user utters the sentence that the system prompted to the the user. This is known as *utterance verification*, and it can be efficiently combined with speaker verification [86].

In text-dependent speaker verification, the pass phrase presented to the system is either the same always, or alternatively, it can be different for every verification session. In the latter case, the system selects randomly a pass phrase from its database and the user is prompted to utter this phrase. In principle, the pass phrase can be stored as a whole word/utterance template, but a more flexible way is to form it online by concatenating different words (or other units such as diphones). This task is called *text-prompted speaker verification.* The advantage of text prompting is that a possible intruder cannot know beforehand what the phrase will be, and playback of pre-recorded speech becomes difficult. Furthermore, the system can be made the user to utter the pass phrase within a short time interval, which makes the intruder harder to use a device or software that synthesizes the customer's voice.

1.3.2 Types of Speaker Recognition

The discussion in the previous subsection was mainly from the viewpoint of automatic speaker recognition. From a more general viewpoint, we can consider also speaker recognition by humans (*auditory recognition*) and a compromise between the automatic and auditory recognition (*semi-automatic recognition*).

Auditory Speaker Recognition

We perform auditory speaker recognition in our everyday life. Intuitively we know that when we have heard a lot of speech from a close friend or relative, we can easily recognize his/her voice. Even if we have not had enough "training speech" we can still guess quite well some other attributes of the speaker (gender and age).

In forensics, auditory speaker recognition might have usage, if there is an *earwitness*, i.e. a person who heard the voice of the criminal during the crime. However, it is been observed that there are considerable differences between individuals in the auditory speaker recognition task [143, 139]. Moreover, as the time between listening the two voices increases, human performance decreases [70]. These are the arguments why the earwitness method is not generally considered a reliable from a forensic viewpoint.

Several studies have been conducted to compare human and machine performance in speaker recognition [95, 143, 152]. Schmidt-Nielsen and Crystal [143] have conducted a large-scale comparison in which nearly 50,000 listening judgments were performed by 65 listeners grouped in panels of 8 listeners. The results were compared with the state-of-the-art computer algorithms. It was observed that individual human listeners vary significantly in their ability to recognize speakers. More interestingly, different listeners seem to use different decision thresholds. In other words, the balance between false acceptance (FA) errors and false rejection (FR) errors depends on the listener. Regarding the comparison with the computer algorithms, Schmidt-Nielsen and Crystal [143] observed that human performs better when the quality of the speech samples is degraded by background noise, crosstalk, channel mismatch, and other sources of noise. With matched acoustic conditions and clean speech, the performance of the best algorithms was comparable with the human listeners.

The study by Schmidt-Nielsen and Crystal was conducted on the NIST 1998 speaker evaluation data. However, since then, performance of computer algorithms has been improved significantly, and the comparison may be already outdated. In recent years, the higher level cues have begun to interest more and more researchers in automatic speaker recognition [29, 163, 135, 21]. For instance, recently automatic systems that use several low- and high-level speaker cues have been introduced [135, 21]. The systems included, for instance, prosodic statistics, phone N-grams, idiolectical features and pronunciation modeling. Significant improvements in the recognition performance over the baseline method was observed when different cues were used in combition.

Semi-Automatic Speaker Recognition

The main conclusion of Rose [139] is that in forensic cases, different techniques must be used jointly in voice comparisons, and specifically, the person who does speech analysis should have a linguistic background. Two speech samples that are compared must be comparable in respect to their linguistic parameters. The selection of the units to be compared, e.g. phonemes or words, must be carefully carried out. This requires an expert phonetician to segment the speech samples by hand. This includes using both aural (listening) and visual information (spectrogram, waveform).

Examples of spectrograms are shown in Figures 1.2 and 1.3. At this stage, the reader is encouraged to try recognizing which two of the spectrograms in Fig. 1.2 belong to the same speaker. This example gives an idea about the complexity associated with voice comparison. The correct answer is given in the last page of this thesis.

Auditory comparison definitely helps in the segmentation and speech unit selection process carried out by a linguistics expert. It is quite obvious that semi-automatic tools must be used in forensic speech comparisons (if you were the accused in the court of law, would you let a computer program to give the judgment?). However, the final analysis should be carried out by careful objective measurements, followed by appropriate statistical analysis.

1.4 Dimension of Difficulty

There are different sources of error in speaker recognition. Some of them are related to the speaker itself, and some to the technical conditions.

1.4.1 Intra-Individual Variation

It is well known that physical (e.g. head cold) and mental states (e.g. depression) of health affect the speaker's voice [110]. Stimulants and drugs also affect the voice. For instance, long-term smokers have often perceptually "rougher" voice quality than non-smokers [110]. If the speaker is under stress, several acoustic parameters of the speaker are different from those under relaxed state.

It is also known that speaker's voice changes in long term due to aging, weight changes, and other physiological changes. Actually, *inter-session variability* is probably the largest source of intra-speaker variation. According to the authors personal experience, voice recorded even during the same day with the same technical conditions might not be matched correctly! Some speakers are also more difficult to model than others [30], especially if the training material is poorly designed. In general, training data should be *phonetically balanced* so that it contains instances of all sounds of the language in different contexts. This ensures that an arbitrary input can be presented to the recognition system.

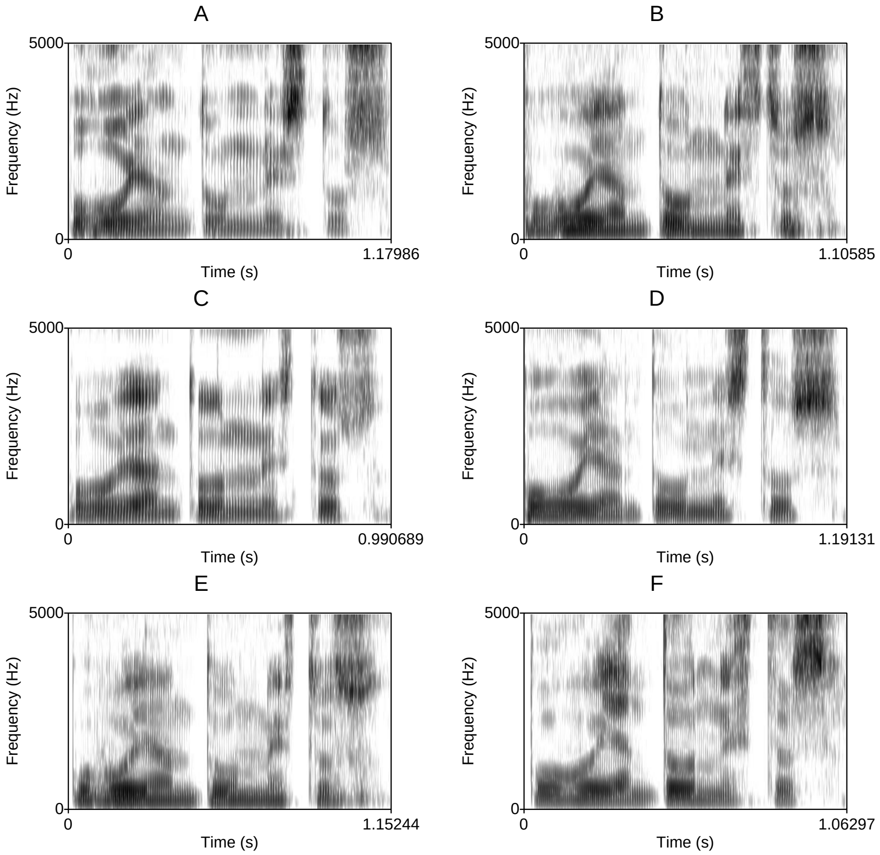

Figure 1.2: Five different male speakers uttering "puhujantunnistus" (speaker recognition). Two of the utterances are produced by the same speaker. Can you detect which two? (the correct answer is given in the last page of the thesis).

1.4.2 Voice Disguise and Mimicry

Voice disguise means deliberately changing one's voice so that it could not be matched with another sample produced by the same speaker. Disguise may be common in forensic cases. For instance, when making a blackmail call, the criminal may keep his nostrils closed, or he might alter his voice in police investigations. Some amount of research has been conducted on voice disguise. For instance, three different parameter sets were studied in [96] in order to find out which of these are the most robust against different types of disguise. The disguise modes included, for instance, talking with a pencil between the front teeth and talking by whispering. However, they studied only variation *within* speakers so they did not take into account the

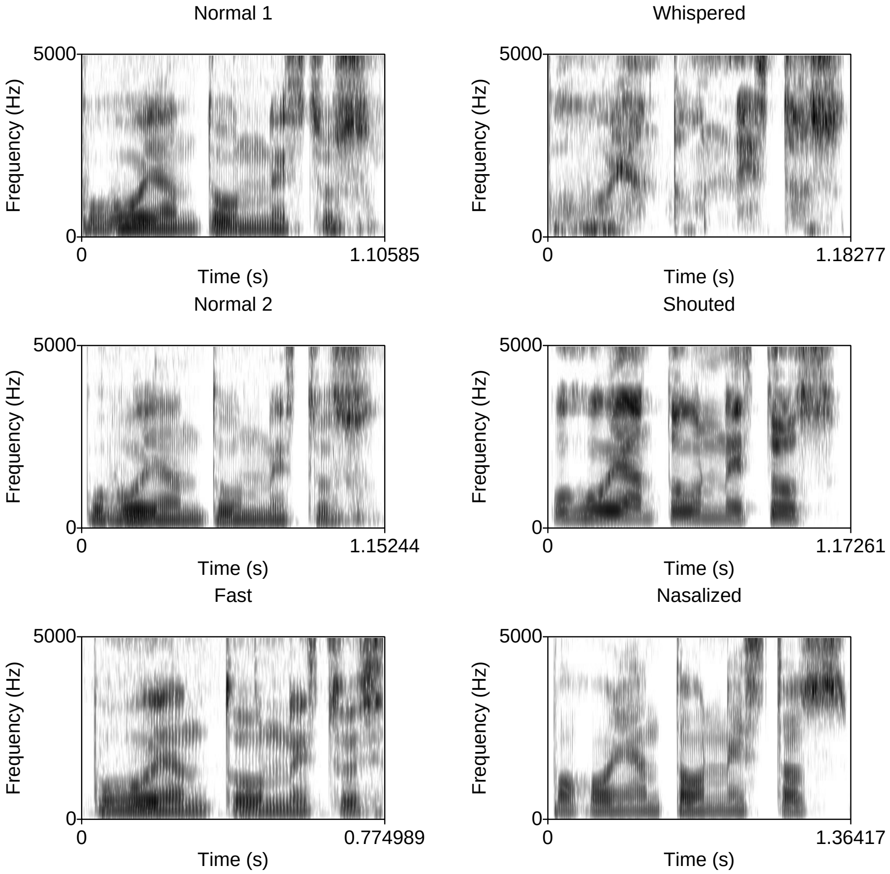

Figure 1.3: Six repetitions of the utterance "puhujantunnistus" (speaker recognition) by the same male speaker with different styles.

inter-speaker variation. Even though a certain parameter would give small intra-speaker variation in respect to disguise, it might be a poor parameter otherwise for speaker recognition. It seems that there is room for studying disguise-resilient parameters. *Imitation* (*impersonation, mimicry*) is a special type of voice disguise where the speaker tends to map his voice to sound like another speaker.

Disguise and imitation definitely degrade the performance of speaker recognition systems. It has been demonstrated that *voice mapping* that converts voice of another speaker to another speaker's voice, can degrade the performance of speaker recognition [45]. General discussion about security concerns in speaker recognition is given in [155].

1.4.3 Technical Error Sources

Several *technical error sources* may degrade the performance of speaker recognition (both auditory and automatic). Some of the most commonly recognized error sources are summarized in Fig. 1.4. Typical assumptions made in noise suppression algorithms and noise-robust feature extraction are the following [67, 58]: (1) noise is stationary in short term, (2) noise has zero mean, and (3) noise is uncorrelated with the speech signal. In general, the type and amount of the noise is unknown.

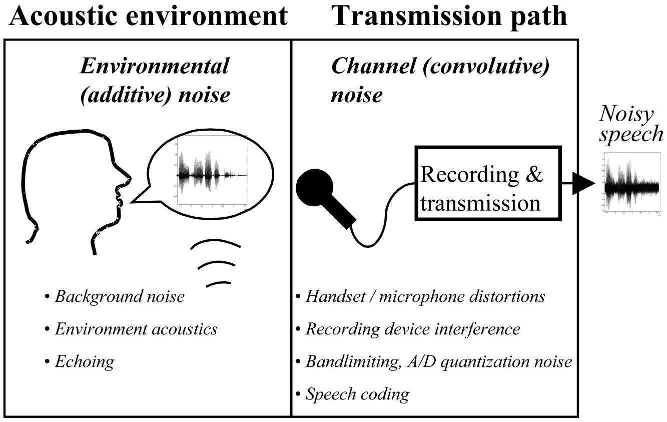
Acoustic environment
Transmission path
Environmental (additive) noise
Channel (convolutive) noise
Noisy speech
Recording & transmission
• Background noise
• Environment acoustics
• Echoing
• Handset / microphone distortions
• Recording device interference
• Bandlimiting, A/D quantization noise
• Speech coding

Figure 1.4: Technical error sources in speaker recognition.

First, speech is recorded with a microphone or telephone handset, and environmental noise (computer hum, car engine, door slams, keyboard clicks, traffic noise, background babble, music) adds to the speech wave. Reverbation adds delayed versions of the original signal to the recorded signal [58].

Poor-quality microphones introduce nonlinear distortion to the true speech spectrum. Quatieri & al. [130] demonstrate, by comparing pairs of same speech segment recorded with good- and poor-quality microphones, that poor-quality microphones introduce several spectral artefacts, such as *phantom formants* that occur at the sums, multiples and differences of the true formants. Formant bandwidths are also widened and the overall spectral shape is flattened.

The A/D converter adds its own distortion, and the recording device might interfere with a mobile phone radio-waves. If the speech is transmitted through a telephone network, it is compressed using lossy techniques which might have added noise into the signal. Speech coding can degrade speaker recognition performance significantly [123, 12].

To sum up, the speech wave seen by the recognition algorithm or human ear is *not* the same wave that was transmitted from the speaker's lips and nostrils, but it has gone through several transformations degrading its quality.

Mismatched conditions is recognized as the most serious error source in speaker recognition [87, 100, 170, 116, 134, 149]. It means that the circumstances of the training and recognition phases are different. In addition to intra-individual variation (such as pitch mismatch), technical mismatches arise from one or several of the following factors:

- Environmental acoustics mismatch
- Mismatch in the type and amount of background noise
- Microphone type mismatch
- Recording quality mismatch

It is easy to imagine a situation where the user speaks training utterances in a clean environment (home) but uses the recognition system in a noisy environment (car, pub, street, office).

1.5 Motivation and Outline of the Thesis

Feature extraction is the first component in an automatic speaker recognition system [20, 43]. Feature extraction transforms the raw speech signal into a compact but effective representation that is more stable and discriminative than the original signal. Since the front-end is the first component in the chain, the quality of the later components (*speaker modeling* and *pattern matching*) is strongly determined by the quality of the front-end. In other words, classification can be at most as accurate as the features.

Several feature extraction methods have been proposed, and successfully exploited in the speaker recognition task. Very often, the methods are adopted directly from the *speech recognition* task, which is somewhat ironical considering the opposite nature of the two tasks. Also, sometimes the feature extraction methods exploit directly psychoacoustical models, i.e. how *humans* process auditory stimuli. By doing so, one implicitely assumes that human listener is the optimal speaker discriminator. However, speech signal might contain features that are not captured by the human ear, but which are important for speaker discrimination.

Quite often the feature extraction methods adopted from speech recognition are used as "black boxes" with fixed parameters. In other words, no effort is put on optimizing the front end of the system. According to the

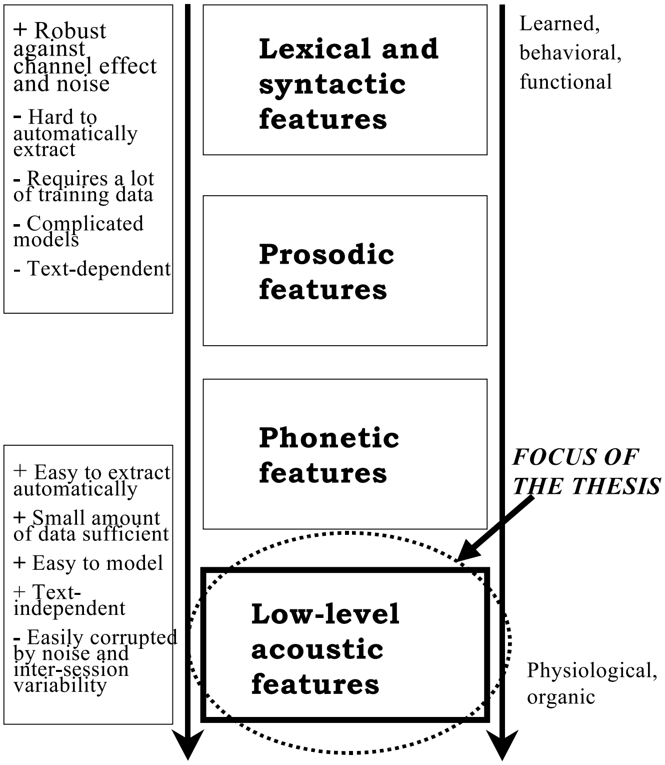
+ Robust against channel effect and noise
- Hard to automatically extract
- Requires a lot of training data
- Complicated models
- Text-dependent
Lexical and syntactic features
Learned, behavioral, functional
Prosodic features
Phonetic features
FOCUS OF THE THESIS
+ Easy to extract automatically
+ Small amount of data sufficient
+ Easy to model
+ Text-independent
- Easily corrupted by noise and inter-session variability
Low-level acoustic features
Physiological, organic

Figure 1.5: Hierarchy of features for automatic speaker recognition.

author's general impression of literature, it seems that currently, at the best we are *guessing* what is the code in the signal that carries our individuality.

This thesis has two main purposes. On the one hand, we attempt to see the feature extraction as a whole, starting from understanding the speech production process, what is known about speaker individuality, and then going into the details of the feature extraction methods. Although prosodics and other high-level features have been recently exploited successfully in speaker recognition (see Fig. 1.5), our attention is on the low-level spectral features due to their text-independence, easy computation/modeling, widespread use and the "black box" effect associated with these.

On the other hand, the second purpose of the thesis is to find out which of the several spectral features are best suited for automatic speaker recognition systems in terms of their reliability and computational efficiency. We aim to find what are the critical parameters that affect the performance and try to give some general guidelines about the analysis parameters. We conduct experiments on two speech corpora using vector quantization (VQ) speaker modeling. The corpora are an American English TIMIT corpus subset consisting of 100 speakers, and a Finnish corpus consisting of 110 speakers. Noise robustness is definitely an important issue in realistic application, but it is outside the scope of this thesis.

The rest of the thesis is organized as follows. Chapter 2 reviews shortly the techniques used in automatic text-independent speaker recognition, with a special focus on speaker modeling and pattern matching. Chapter 3 gives a background of the speech production mechanism, and phonetic aspects of speaker recognition. Chapter 4 reviews the signal processing background needed in feature extraction. Chapter 5 gives an overview of the spectral features used in automatic speaker recognition. Chapters 6 and 7 include the description of the speech material, and the experimental results, respectively. Finally, conclusions are drawn in Chapter 8.

Chapter 2

AUTOMATIC SPEAKER RECOGNITION

Figure 2.1 shows the abstraction of an automatic speaker recognition system. Regardless of the type of the task (identification or verification), system operates in two modes: *training* and *recognition* modes. In the training mode, a new speaker (with known identity) is enrolled into the system's database. In the recognition mode, an unknown speaker gives a speech input and the system makes a decision about the speaker's identity.

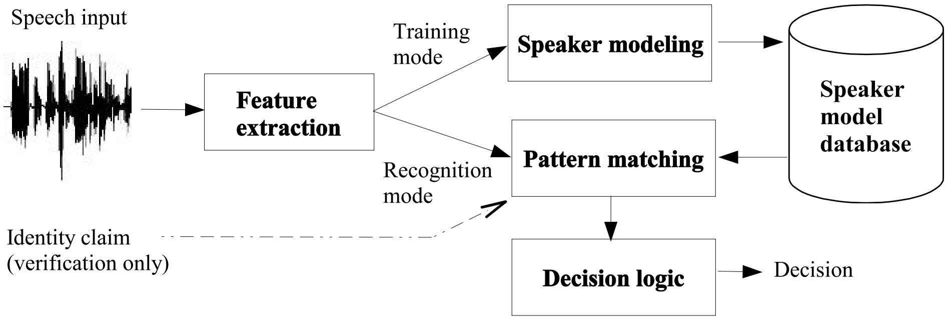

Figure 2.1: Components of automatic speaker identification system.

Both the training and the recognition modes include *feature extraction*, sometimes called the *front-end* of the system. The feature extractor converts the digital speech signal into a sequence of numerical descriptors, called *feature vectors*. The features[1] provide a more stable, robust, and compact

[1] Elements of feature vector are called *features*. Alternative terms for feature are *measurement*, *attribute*, *quantity* and *parameter*.

representation than the raw input signal. Feature extraction can be considered as a data reduction process that attempts to capture the essential characteristics of the speaker with a small data rate.

In the training phase, a *speaker model* is created from the feature vectors. The aim is to model the speaker's voice so that it *generalizes* beyond the training material. In other words, unseen vectors can be classified correctly. A recent overview of various modeling techniques is given in [132].

In the recognition phase, features are extracted from the unknown speaker's voice sample. *Pattern matching* refers to an algorithm, or several algorithms, that compute a match score between the unknown speaker's feature vectors and the models stored in the database. The output of the pattern matching module is a *similarity score.*

The last phase in the recognition chain is *decision making.* The decision module takes the match scores as its input, and makes the final decision of the speaker identity, possibly with a confidence value [47, 59]. The type of the decision depends on the task. For the verification task, the binary decision is either acceptance or rejection of the speaker. In the case of identification, there are two possibilities. In the closed-set identification task, the decision is the ID number of the most similar speaker to the unknown speaker. In the open-set task, there is an additional decision that the speaker is *none* of the registered speakers ("no decision").

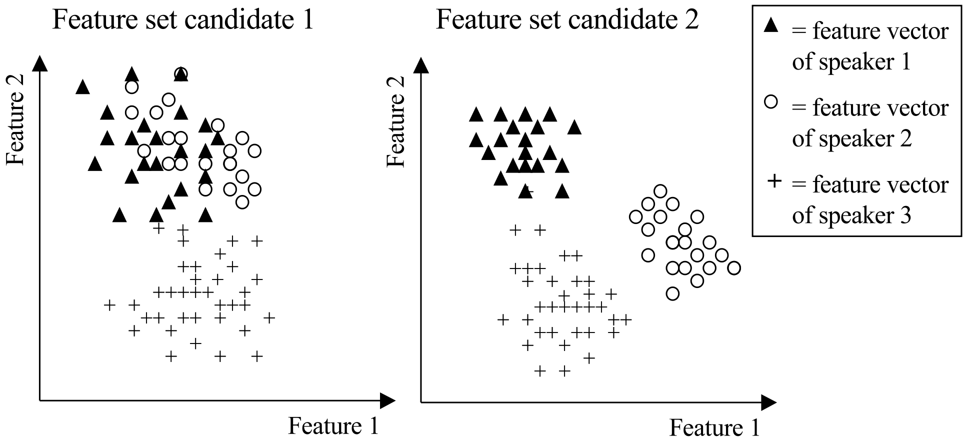

Figure 2.2: Examples of two-dimensional feature sets with poor and good discrimination.

2.1 Feature Extraction

Formally, feature extraction is understood as a process which transforms originally high-dimensional vectors into lower dimensional vectors. So essentially, it is a mapping $f : \mathbb{R}^N \rightarrow \mathbb{R}^d$, where $d \ll N$. Feature extraction is a necessary operation for two main reasons. First, in order the statistical speaker models to be robust, the number of training samples must be large enough compared to the dimensionality of the measurements. The amount of needed training vectors grows exponentially with the dimensionality. This phenomenon is known as *curse of dimensionality* [16, 64, 65]. The second reason for performing feature extraction is the reduced computational complexity.

For speaker recognition purposes, optimal feature has the following properties [139]:

1. high inter-speaker variation,

2. low intra-speaker variation,

3. easy to measure,

4. robust against disguise and mimicry,

5. robust against distortion and noise,

6. maximally independent of the other features.

The first two requirement require that features are as discriminative as possible. Examples of two-dimensional feature sets are shown in Fig. 2.2. From the two candidate feature sets, the set 2 obviously discriminates better the speakers. Notice, however, that even in the case of the feature set 1, the feature 2 discriminates the speaker 3 from the two other speakers. Notice also that neither one feature alone discriminates the speakers perfectly, but both features are needed.

The features should be *easily measurable.* This includes two factors. Firstly, the feature such occur frequently and naturally in speech so that it could be extracted from short speech samples. Secondly, the feature extraction itself should be easy. In automatic recognition, the features must be measurable without the aid of a human expert.

A good feature is *robust* against several factors like voice disguise and distortion/noise. Finally, different features extracted from the speech signal should be *maximally independent of each other.* If two correlated features are combined, nothing is gained, and in fact, this may even degrade recognition results.

No feature has all the requirements listed above, and we can relax some of the requirements in automatic speaker recognition. Robustness against disguise and mimicry is clearly beyond the scope of this thesis. By the nature of our task, we can forget features that require a human expert involved. In practise, the signal processing methods used in the feature extraction are computationally efficient.

Some of the widely used feature sets, such as mel-cepstrum (MFCC) and line spectrum pairs (LSP), have already rather uncorrelated features. We would like to point out that there exists several general-purpose feature transformation methods that can be used to transform original features into a new space where they are more discriminative and/or have smaller inter-feature correlations. Some of these include *linear discriminant analysis* (LDA) [41, 31], *Karhunen-Loeve transform* (KLT) [41], and *independent component analysis* (ICA) [61].

Finally, we would like to point out the difference between feature *extraction* and feature *selection*. In feature extraction, the new features are a function of all of the original features. In contrast, in feature *selection*, a subset of the original features are selected in a way that attempts to maximize some separability criterion. A good review of feature selection methods is given in [65].

2.2 Speaker Modeling and Matching

There are two main approaches for estimating the class-conditional (speaker-dependent) feature distributions: *parametric* (stochastic) and *non-parametric* (template) approaches [31, 41, 20]. In the parametric approach, a certain type of distribution is fitted to the training data by searching the parameters of the distribution that maximize some criterion. The non-parametric approach, on the other hand, makes minimal assumptions about the distribution of the features.

The pattern matching phase consists of computing a similarity score for the unknown speaker's feature vectors and all speaker models. The similarity (or dissimilarity) measure depends on the type of the speaker models. In the next two subsections, we consider the two most popular approaches to text-independent speaker recognition, *vector quantization* (VQ) approach and *Gaussian mixture model* (GMM). VQ is a non-parametric method whereas GMM is a parametric method. Figure 2.3 shows an example of VQ and GMM-based modeling of the same data set with two different model sizes.

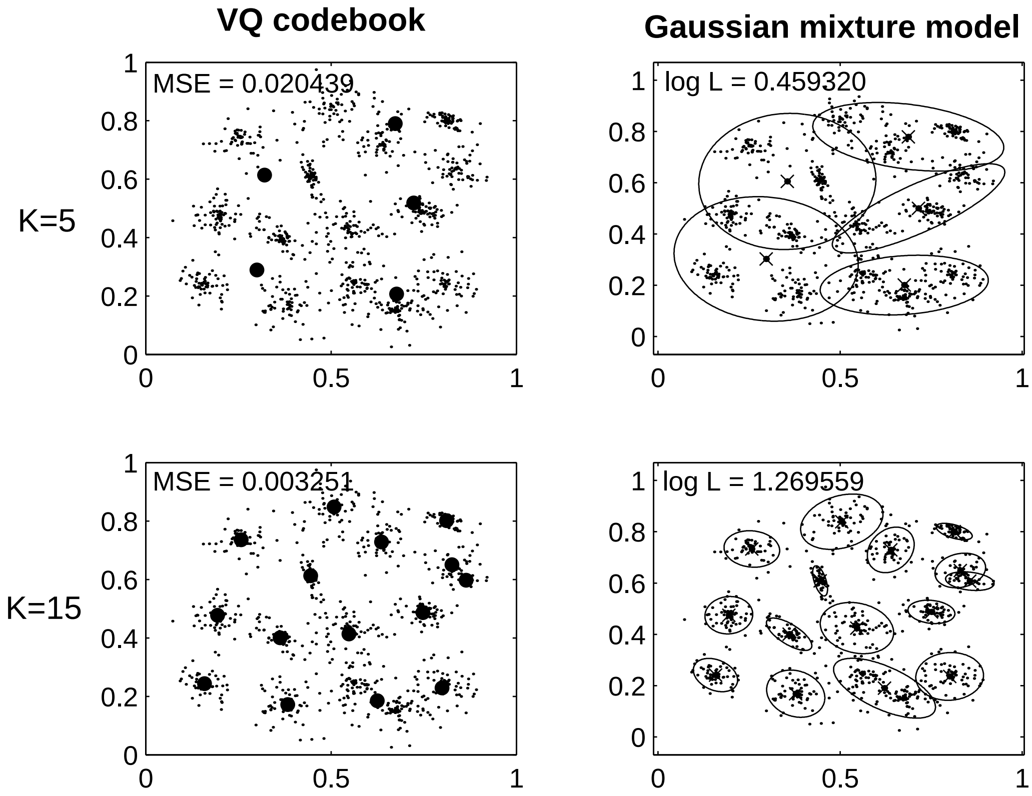

Figure 2.3: Examples of VQ- and GMM-based modeling for different model sizes ($K = 5, 15$).

2.2.1 VQ-Based Speaker Recognition

In VQ-based approach [148], the speaker models are formed by clustering the speaker's feature vectors in K non-overlapping clusters. Each cluster is represented by a *code vector* $\boldsymbol{c}_i$, which is the centroid (average vector) of the cluster. The resulting set of code vectors $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_K\}$ is called a *codebook*, and it serves as the model of the speaker. The model size (number of code vectors) is significantly smaller than the training set. The distribution of the code vectors follows the same underlying distribution as the training vectors [46]. Thus, the codebook effectively reduces the amount of data by preserving the essential information of the original distribution.

There are two design issues in the codebook generation: (1) the method for generating the codebook, and (2) the size of the codebook. Regarding the codebook size, a general observation has been that increasing the codebook size reduces recognition error rates [148, 38, 37, 54, 74, 75]. However, if the codebook size is set too high, it learns the training samples but not the general distribution (this is called *overfitting*). The claim that the best speaker model is the data itself [27], is *not* true in general according to author's experience.

Two classes of methods for codebook generation exist: *unsupervised* and *supervised* learning algorithms. In unsupervised methods, each speaker's codebook is trained independent of each other, whereas in supervised training, the intercorrelations between the codebooks are taken into account so that the codebooks have minimal overlap. Usually the unsupervised methods are used since they have less user-adjustable control parameters. One supervised codebook training approach termed *group vector quantization* (GVQ) has been proposed by He & al. [54]. The idea is to first train the codebook individually, and then fine-tune them so that inter-speaker differences are emphasized.

Of the unsupervised codebook training algorithms, the most popular and one of the most simplest one is the *generalized Lloyd algorithm* (GLA) [89]. The algorithm is also known as *Linde-Buzo-Gray algorithm* (LBG) according to its inventors[2]. The user must require the desired codebook size K. GLA then starts from an initial codebook of size K (usually, randomly selected vectors from the training set), which it iteratively refines in two successive steps until the codebook does not change.

We studied the effect of unsupervised codebook training algorithms on the speaker identification task in [74]. We observed that the choice of the algorithm does not have much effect to the recognition performance. A possible explanation is that the feature vectors obtained from overlapping speech frames may not have a clustering structure, but they form less or more a continuous density [76]. Therefore, the codebook training serves as sub-sampling of the training data rather than finding a clustering structure. The selection of the clustering algorithm is therefore not a vital issue. The matching function in VQ-based speaker identification is typically defined as the *quantization distortion* between two vector sets $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$ and $C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\}$. Consider a feature vector $\boldsymbol{x}_i$ generated by the unknown speaker, and a codebook C. The quantization distortion d_q of the vector $\boldsymbol{x}_i$ with respect to C is given by

$$d_q(\boldsymbol{x}_i, C) = \min_{\boldsymbol{c}_j \in C} d(\boldsymbol{x}_i, \boldsymbol{c}_j), \tag{2.1}$$

where $d(\cdot, \cdot)$ is a distance measure defined for the feature vectors. The code vector $\boldsymbol{c}_{j^*}$ for which $d(\boldsymbol{x}_i, \boldsymbol{c}_{j^*})$ is minimum, is the *nearest neighbor* of x_i in

[2] Although some authors make a difference between the *K-means* algorithm and the GLA/LBG, we do not. Essentially they are the same algorithm.

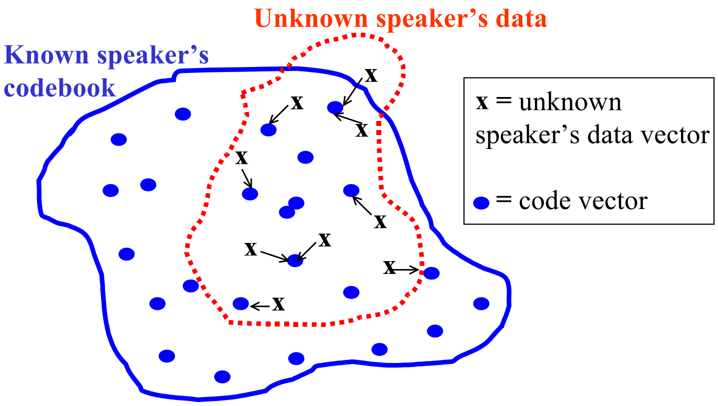

Figure 2.4: Illustration of VQ match score computation.

the codebook C. Most often, *Euclidean* or *Euclidean squared* distance measure is used due to the straightforward implementation and intuitive notion (for instance, it can be shown that Euclidean distance between two cepstral vectors measures the squared distance between the corresponding short-term log spectra [131]). Sometimes *Manhattan distance* and *Mahalanobis distance* [31] are used. Also, tailored distance measures for a certain feature set can be used (see [131, 108]).

The *average quantization distortion* D_Q is defined as the average of the individual distortions:

$$D_Q(X, C) = \frac{1}{T} \sum_{i=1}^{T} d_q(\boldsymbol{x}_i, C), \tag{2.2}$$

Obviously, for $X \subseteq C$, $D_Q(X, C) = 0$. The better the sets X and C match to each other, the smaller the distortion is. The computation of the distortion is illustrated in Fig. 2.4. Notice that (2.2) is not symmetrical, i.e. $D_Q(X, C) \neq D_Q(C, X)$. We will assume that the first argument of D_Q is the sequence of the unknown speaker's feature vectors, and the second argument is a known speaker's codebook. It is also worth noticing that multiplication of $D_Q(X, C_i)$ by a constant factor does not affect the ordering of distortions $\{D_Q(X, C_1), \ldots, D_Q(X, C_N)\}$. It does not matter whether (2.2) is normalized by T, since it is the same for all speakers.

Several modifications have been proposed to the baseline VQ distortion matching [158, 104, 57, 72, 75, 35]. For instance, in [72, 75], we assign to each VQ code vector a discriminative weight so that a code vector that is close to some other speaker's code vector is given a small contribution to the overall distance. Some of the various improved VQ methods are compared in [35]. It was found out that the discriminative training [54] and partition-normalized distance measure [158] were the most efficient methods, and they could be efficiently combined.

2.2.2 GMM-Based Speaker Recognition

In GMM-based speaker recognition [137, 136], the speaker model consists of K Gaussian distributions parametrized by their *a priori* probabilities P_j, mean vectors $\boldsymbol{\mu}_j$ and covariance matrices $\boldsymbol{\Sigma}_j$. We denote this model by $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_K\}$, where $\lambda_j = (P_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ are the parameters of the jth component. The parameters of the model are typically estimated by maximum likelihood estimation, using the *Expectation-Maximization* (EM) algorithm [26, 16].

The matching function in GMM is defined in terms of *likelihood*. Assuming that the observations $\{\boldsymbol{x}_i\}_{i=1}^{T}$ are statistically independent, the *log-likelihood* of the GMM is given by

$$\begin{aligned} L &= \log p(X|\lambda) \\ &= \log \prod_{i=1}^{T} p(\boldsymbol{x}_i|\lambda) \\ &= \sum_{i=1}^{T} \log p(\boldsymbol{x}_i|\lambda), \end{aligned} \tag{2.3}$$

where $p(\boldsymbol{x}_i|\lambda)$ is the Gaussian mixture density:

$$p(\boldsymbol{x}_i|\lambda) = \sum_{j=1}^{K} P_j \, \mathcal{N}_j(\boldsymbol{x}_i), \tag{2.4}$$

with the mixing weights constrained by

$$\sum_{j=1}^{K} P_j = 1. \tag{2.5}$$

The component densities $\mathcal{N}_j(\boldsymbol{x}_i)$ are given by the multivariate Gaussian density [31]:

$$\mathcal{N}_j(\boldsymbol{x}_i) = (\sqrt{2\pi})^{-\frac{d}{2}} |\boldsymbol{\Sigma}_j|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T \, {\boldsymbol{\Sigma}_j}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\}. \tag{2.6}$$

The determinant $|\boldsymbol{\Sigma}_j|$ and the inverse covariance matrix ${\boldsymbol{\Sigma}_j}^{-1}$ can be precomputed in the training phase and stored. Depending on the type of the covariance matrix, the number of multiplications in (2.6) is approximately d^2+d and d for the full and diagonal matrices, respectively. If the feature vector components can be assumed uncorrelated, diagonal covariance matrices can be used. The covariance matrix inversion in this case consist of simply taking inverse values of the diagonal elements, and thus it is also numerically more stable than the full covariance case. As an example, typically diagonal covariance matrices are used with the mel-frequency cepstral coefficients (MFCC) [67] since the computation of these features includes as a last step a decorrelating transform, the discrete cosine transform (DCT).

2.2.3 Discussion and Other Approaches

Although the concept of GMM is intuitive, the EM-algorithm is pretty complex from the implementation point of view. It requires, for instance, setting of the minimum allowed values for the components variances to avoid numerical problems [137]. The computation of the multivariate Gaussian density easily produces an overflow if the feature space has high dimensionality. This means that in practise there is an upper limit for the number of features before numerical problems arise. The VQ approach does not have these problems, but its basic deficiency is that the clusters cannot overlap, and thus the density function the code vectors represent is not continuous. It is useful to notice that GMM is an extension of the VQ in which clusters are allowed to overlap.

The advantages of both methods are exploited in [79, 118, 144][3]. First, the feature space is partitioned into K disjoint clusters using the LBG algorithm. Then, the covariance matrices of each cluster are computed from the vectors that belong to that cluster. The mixing weight of each cluster is computed as the proportion of vectors belonging to that cluster. All studies [79, 118, 144] show that this simple algorithm gives comparable results with the GMM-based speaker recognition with much simpler implementation.

Several other approaches to speaker modeling in text-independent speaker recognition have been proposed, including for instance neural nets, monogaussian models, support vector machines, and decision trees. Overview and comparisons of some of the methods are given in [37, 132]. So-called *classifier ensembles* (*committee classifiers*) have become also popular in the past few years. The basic idea is to model each feature set with the modeling technique best suited for it, and to combine the sub-scores of the classifiers into the final score. An overview and comparison of several combinations strategies for speaker recognition is given in [138].

[3] In all papers, essentially the same algorithm is presented.

2.3 Decision Making

The final step in speaker recognition process is the decision. The feature extraction and pattern matching are same for different speaker recognition tasks, but the decision depends on the task. Let us denote generally a speaker model of speaker i by S_i, and let $S = \{S_1, \ldots, S_N\}$ be the speaker database of N known speakers. Without assuming a specific speaker model/classifier, let $\mathit{score}(X, S_i)$ be the match score between the unknown speaker's feature vectors $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$ and the speaker model S_i. We will assume that a larger match score corresponds to better match. In the case of distance-based classifiers, we can put a minus sign in front of the match score without loosing generality.

In closed-set speaker identification task, the decision is simply the speaker index i^* that yields the maximum score:

$$i^* = \arg\max_j \mathit{score}(X, S_j), \tag{2.7}$$

where the maximum is taken over the speaker database S. In the verification task, the decision is given as follows:

$$\mathit{score}(X, S_j) \begin{cases} \geq \Theta_j, & \text{accept speaker } j \\ < \Theta_j, & \text{reject speaker } j, \end{cases} \tag{2.8}$$

where Θ_j is the *verification threshold*. The verification threshold can be set the same for all speakers, or it can be speaker-dependent. The threshold(s) are determined so that a desired balance between the *false acceptances* (FA) and *false rejections* is obtained (FR). The former means accepting an impostor speaker, and the latter means rejecting a true speaker. There is a trade-off between the two errors: when the decision thresholds Θ_i are increased, false acceptance error decreases but false rejection error increases, and vice versa. The balance between the error depends on the application. Different threshold setting methods can be found in [13].

In the open-set identification task, the decision is given as follows:

$$\text{decide} \begin{cases} i^*, & \text{if } i^* = \arg\max_j \mathit{score}(X, S_j) \wedge \mathit{score}(X, S_{i^*}) \geq \Theta_{i^*} \\ \text{no one}, & \text{otherwise} \end{cases} \tag{2.9}$$

In other words, we find the best matching speaker, and if the match score of this speaker is above the decision threshold, we accept the speaker. Otherwise we decide that the speaker is no one.

In practise, the score that we compare to threshold is not the raw output score of the classifier, but instead, a *normalized* score is used [43]. The

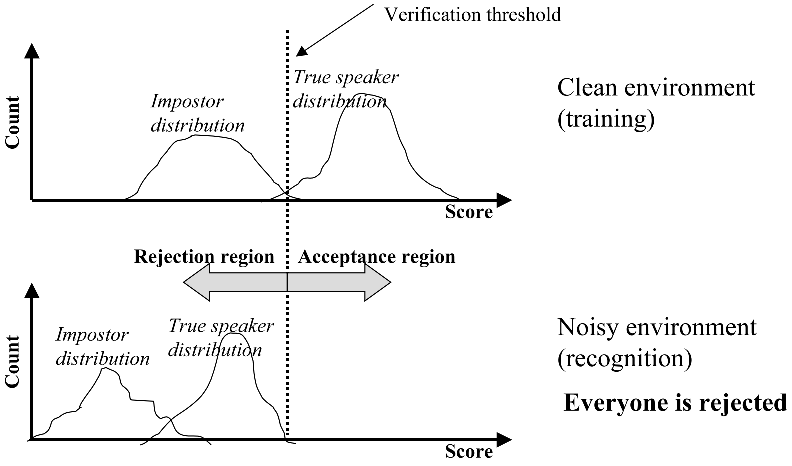

Figure 2.5: The problem without score normalization in the verification task.

motivation for score normalization is the following (see Fig. 2.5). Suppose that the training material was recorded in clean environment, but the recognition happens in noisy conditions (or otherwise acoustically mismatched conditions). The match scores are expected to get worse, since the claimant speaker's feature vectors are different than in the training phase. If the raw score is compared with the threshold, the false rejection rate increases. However, also the other speaker's match match scores get worse, and therefore, the score normalization attempts to transform the client speaker scores relative to the general score distribution. There are several approaches to score normalization, and these can be found e.g. in [13].

Chapter 3

PRODUCTION, ACOUSTICS AND PERCEPTION OF SPEECH

Spoken language is commonly categorized into three different perspectives: *articulatory*, *acoustic* and *perceptual* perspectives. In the articulatory approach, one attempts to describe how humans produce speech sounds by examining the anatomy and physiology of the voice production organs. In the acoustic approach, the acoustic speech signal itself is the object of interest. In the perceptual approach, one examines the anatomy and physiology of the human hearing mechanism and tries to find models which relate the objective acoustic measurements to subjective perceptual attributes.

3.1 Articulatory Approach

Speech production is a complex process which, in a simplified model, consists of the following consecutive tasks [131]:

1. Message formulation,
2. Coding of the message into a language code,
3. Mapping of the language code into neuro-muscular commands,
4. Realization of the neuro-muscular commands

We are not interested in the details of these steps here. The end result of the complex neuro-muscular commands is the physical movements of the voice production organs, whose parts are shown in Fig. 3.1. In a common classification [88] three physiological components of speech production are recognized: (1) *subglottal component*, which consists of the lungs and associated respiratory muscles, (2) the *larynx*, which includes the vocal folds and (3) *supralaryngeal vocal tract*, which consists of the pharyngeal, oral, and nasal cavities. All of the three components, especially the vocal tract, are complex systems with inherently time-varying nature: during speaking, the configuration of these components changes continuously.

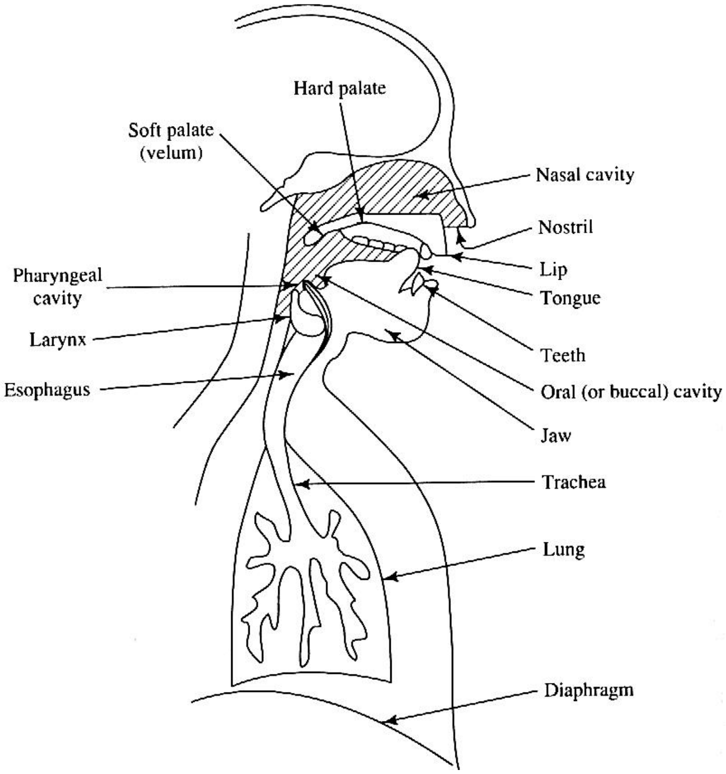

Figure 3.1: Human voice production system [58].

3.1.1 The Subglottal Respiratory System

The subglottal component produces an airstream which powers the speech production process. During inspiration, muscular force is used in filling the lungs. The lungs will expand in their volume in the same way as what happens to a rubber balloon when one blows air into it, and energy is stored in the elastic expansions of each lung. During expiration, this energy will be spontaneously released due to a so-called *elastic recoil force* [85, 88]. The resulting airstream flows through the *trachea* (or *windpipe*) to the larynx.

3.1.2 The Larynx

The larynx is responsible for different *phonation mechanisms* [85, 88]. This refers to producing acoustic energy which serves as an input to the vocal tract. More specifically, the *vocal folds* and *glottis* are the interesting parts from a speech production viewpoint. The larynx has also an important life-supporting function: during swallowing, it will block the trachea and open the way to the esophagus. The glottis is a small, triangular-shaped space between the vocal folds [85, 88]. The egressive airstream from the lungs passes through the glottis to the vocal tract. The action of the vocal folds determines the phonation type, whose major types are *voicelessness*, *whisper* and *voicing* [85].

During whisper and voiceless phonation, the vocal folds are apart from each other, and the airstream from the lungs will pass through the open glottis. The difference between whisper and voiceless phonation is determined by the degree of the glottal opening. In whisper, the glottal area is smaller. This results in a turbulent airstream, generating the characteristic “hissing” sound of whispering [85]. In voiceless phonation, the area of the glottis will be larger and the airstream is only slightly turbulent when it enters the vocal tract. An example of voiceless phonation is the initial [h] in the Finnish word “hattu” (a hat).

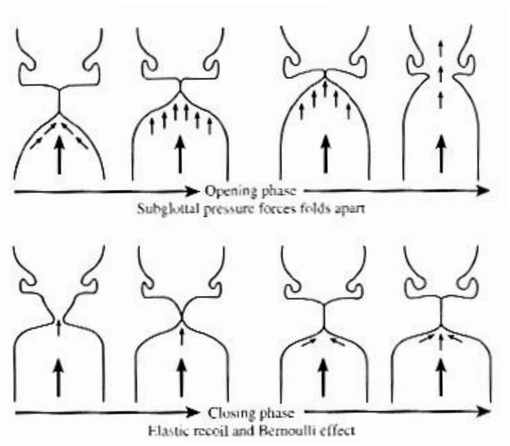
Opening phase
Subglottal pressure forces folds apart
Closing phase
Elastic recoil and Bernoulli effect

Figure 3.2: Illustration of one glottal cycle [85].

Voicing is more complex mechanism than voiceless phonation and whisper. Voicing is a result of periodic repetitions of the vocal folds opening and closing. This is depicted in Fig. 3.2. During the opening phase, the respiratory effort builds up the subglottal pressure until it overcomes the muscular force which keeps the vocal folds together. The glottis opens, and the com-

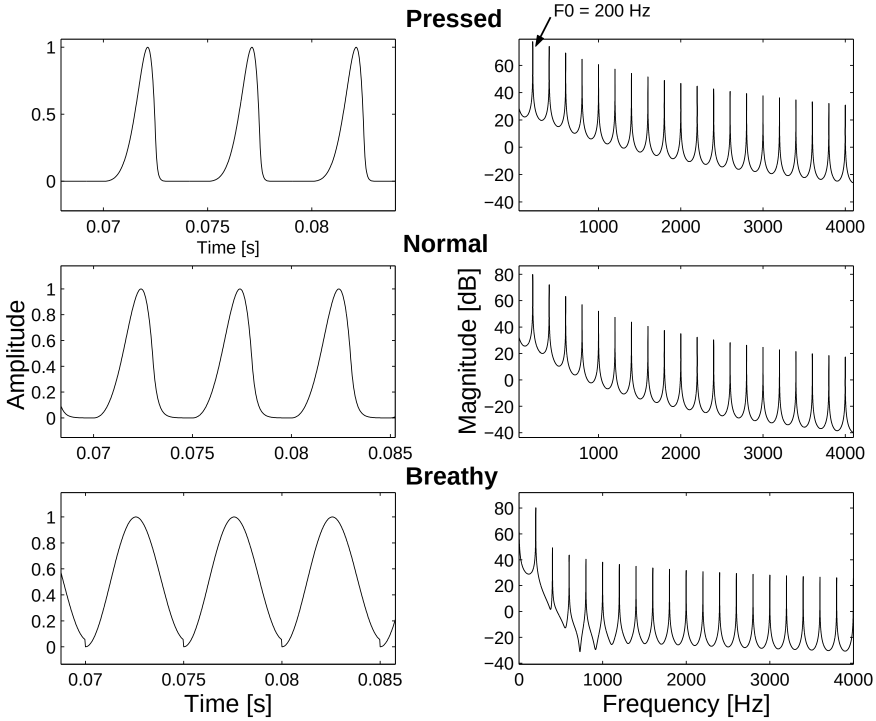

Figure 3.3: Examples of glottal airstreams and their spectra.

pressed airstream bursts into pharynx with a speed of 2-5 m/s [85]. This relatively high speed causes a local drop of air pressure at the glottis, and as a consequence of this so-called *Bernoulli effect*, the vocal folds start to close. The combined effort of the Bernoulli effect and muscular tension overcomes the force of respiratory pressure very quickly, and the vocal folds are pulled together. The coupling of the opening and closing phases continues, and the result is a periodic stream of air puffs which serves as the acoustic source signal for the voiced sounds. The opening-closing is not perfectly periodical in the mathematical sense, and therefore the term *quasi-periodic* is sometimes used in this context.

An illustration of hypothetical glottal flows[1] and their spectra is shown in Fig. 3.3. It can be seen that the shape of the glottal pulse affects the roll-off in the high frequencies. This affects the overall *voice quality*. For instance, in the bottom case, the glottis stays (almost) open. As a result, the spectrum rolls off rapidly and the perceived voice could be verbally characterized perhaps as “breathy”.

The rate at which the vocal folds vibrate is referred to as *fundamental fre-*

[1] The waveforms are generated by so-called *Liljencrants-Fant glottal model*.

quency and abbreviated $F0$. The inverse of $F0$ is referred to as *fundamental period*, and it is the time which a single opening-closing cycle takes. Fundamental frequency differs between females, males and children [88][2]. This results from anatomical differences; usually females have smaller vocal folds compared to males, and due to their higher tension, they vibrate at higher rate. Children have even smaller vocal folds. The average $F0$ values in conversational speech in European languages for males, females and children are approximately 120 Hz, 220 Hz, and 330 Hz, respectively [85]. It is important to keep in mind that $F0$ is defined only for the voiced phonation, and it is undefined for the other phonation types.

3.1.3 The Supralaryngeal Vocal Tract

The *supralaryngeal vocal tract*, or simply *vocal tract*, is the most important, and also most complex system in the speech production process. Vocal tract is a generic term which refers to the voice production organs above the larynx. The main parts of the vocal tract are shown in a schematic drawing of Fig. 3.4. The three main cavities of the vocal tract are the pharyngeal, oral and nasal cavities. The *soft palate* or *velum* controls the amount of airflow to the nasal cavity.

The parts of the vocal tract, especially those of the oral cavity, serve as *articulators*. Each *articulatory gesture*, e.g., a tongue movement, aims at a certain ideal *phonetic target*. The realized *acoustic event* approximates the phonetic target. Articulatory gestures in general overlap in time. In other words, articulation of the preceding phonetic target affects the next target (and therefore, its acoustic parameters also). The phenomenon is known as *coarticulation*. Due to coarticulation, the phonetic targets are not coded in the speech signal as simple linear segments following each other in time, such as letters in written text. Coarticulation is one of the reasons why automatic speech segmentation into phonetic events remains a difficult problem.

The most flexible articulator is the tongue, which can have various positions and orientations. It can be made, for instance, to form a narrow passage (a so-called *stricture*) in the vocal tract, through which the airstream flows. Due to the stricture, the airstream becomes turbulent and makes the characteristic “hiss noise” of certain phonemes. An example of this is the voiceless fricative [s] in the word “sade” (rain), where the stricture is formed by setting the body of the tongue against the hard palate.

In the production of vowels, on the other hand, the airstream flows freely

[2] An interested reader may verify this easily by performing F0 shift using any speech processing system, as Praat [127].

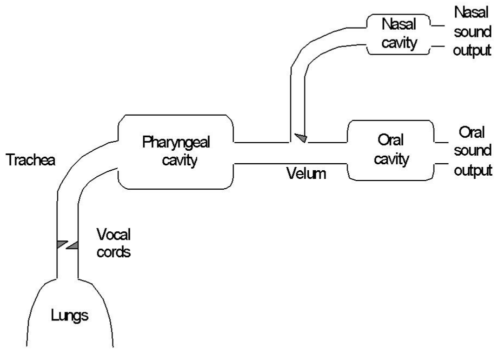

Figure 3.4: A simplified model of the vocal tract.

through the vocal tract. However, also in this case, there is a constriction in the oral cavity. The cross-sectional area of the constriction is significantly larger compared to the fricatives, and therefore turbulence is not built up. Instead, a *standing wave* arises. The place and cross-sectional area of the oral constriction determines (mostly) which vowel is produced [18, 88]. For this reason, vowels are often classified as *front, back* or *mid* vowels based on the place of the oral constriction. For instance, [a] is a back vowel and [i] is a front vowel. The roundness of lips also affects the phonetic quality of certain vowels [85, 88].

Nasal sounds are produced by the nasal cavity with the velum open and a closure in the oral cavity. One can consider the velum as a valve which controls the amount of airstream to the nasal cavity. For instance, during the production of the initial [m] in the word "mahtava" (great) one can notice that one's lips are closed. Consequently, the airstream flows via nasal cavity and egresses from the nostrils. Another example of a nasal sound is the initial [n] in the word "nainen" (woman). This sound is also produced by making a complete closure, this time by the body of the tongue. So, the place of oral constriction plays role in the phonetic quality of different nasal sounds.

3.2 Acoustic Approach

Articulatory phonetics attempts to describe how speech sounds are produced in terms of articulatory gestures, whereas the acoustic approach aims at finding acoustic correlates of the physiology and behavioral aspects of the voice production organs. The acoustic speech signal does not carry an X-ray image of the speaker's vocal tract or a video clip of the lip movements. However, certain acoustic parameters have more or less direct correlates with

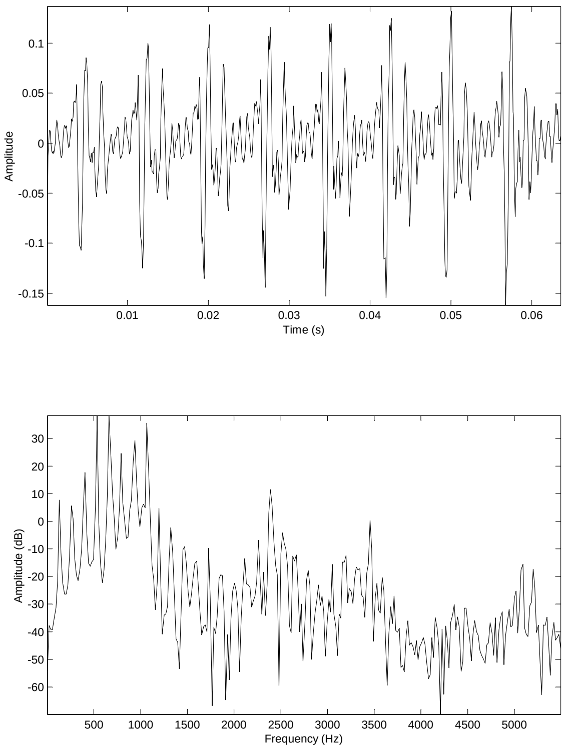

Figure 3.5: The waveform and spectrum of vowel [a] uttered by male speaker.

the anatomy and physiology of the voice production organs.

Speech signals can be analyzed either in the *time domain* or in the *frequency domain*. An example of a time-domain *waveform* and the *short-term spectrum* of the same segment is shown in Fig. 3.5.

3.2.1 Spectrographic Analysis of Speech

In acoustic phonetic research, two useful representations are the waveform and a time-frequency plot called *spectrogram*. The waveform shows the air pressure variations, whereas the spectrogram shows the magnitudes of different frequencies as a function of time. Examples of spectrograms are shown in Figures 3.6 and 3.7. The grey level in the position (t, f) shows the relative magnitude of the frequency f at time t so that darker regions correspond to higher magnitudes.

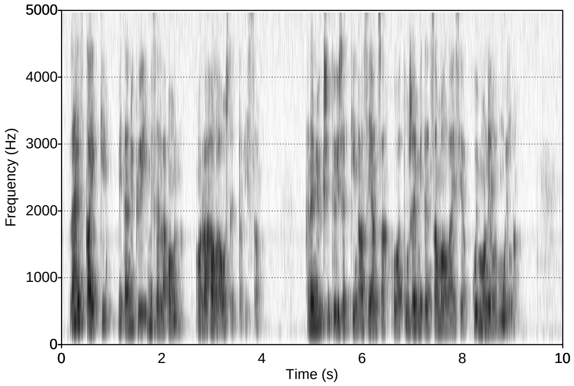

Figure 3.6: An example of a wideband spectrogram.

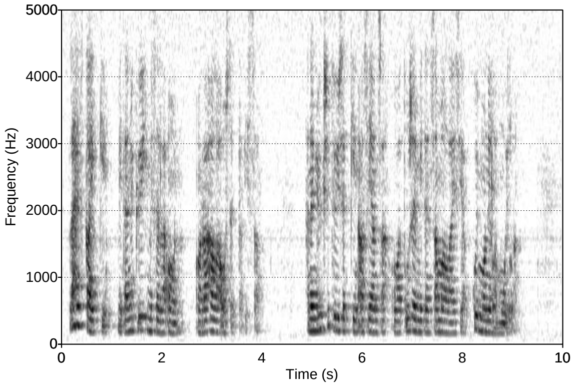

Figure 3.7: A narrowband spectrogram of the same utterance shown in Fig. 3.6. Notice the harmonics of $F0$ seen as dark horizontal bars in the voiced portions.

There is a trade-off between the time and frequency resolutions. If the time resolution is high, i.e., a short analysis window is used, the frequency resolution gets worse and vice versa. The resolutions are approximately inversely proportional to each other. For instance, a time resolution of 20 milliseconds (0.02 seconds) gives approximately a frequency spacing of $1/0.02 = 50$ Hz. In signal processing and physics, the relationship between time- and frequency resolutions is known as *uncertainty principle* [98, 129].

There are two types of spectrograms: *wideband* and *narrowband* spectrograms (see Figures 3.6 and 3.7. In wideband spectrograms, the bandwidth of the analysis filter is around 300 Hz and thus the time spacing is approximately $1/300$ s $= 3.33$ ms. For narrowband analysis, the bandwidth is around 50 Hz

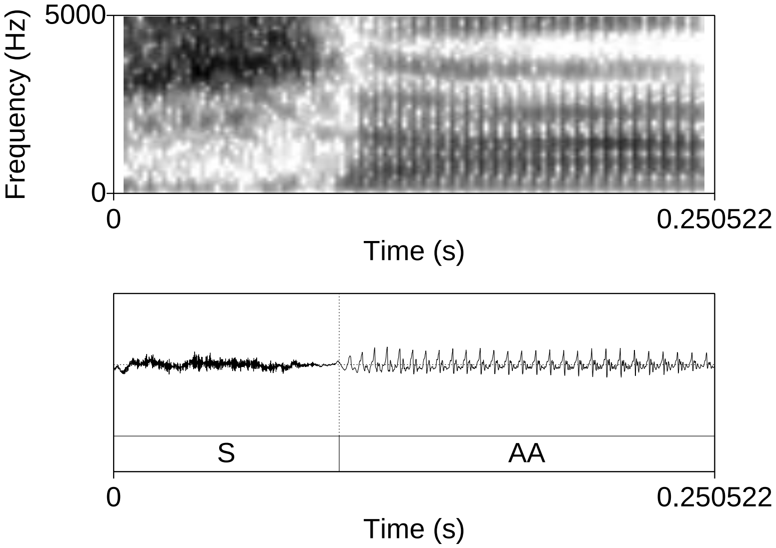

Figure 3.8: Examples of unvoiced and voiced sounds. The "saa" part of the phrase "osaaminen" (know-how) uttered by female speaker.

and thus the time spacing is around 1/50 s = 20 ms [51]. Wideband spectrograms are suitable for tracking vowel formants whereas the narrowband spectrograms can be used in $F0$ estimation [51].

3.2.2 The Source-Filter Model

Speech production can be modeled by so-called *source-filter model* [36]. As the name suggests, the model considers the voice production mechanism as a combination of two components: the *voice source* and the *acoustic filter*. The "source" refers to the airstream generated by the larynx and the "filter" refers to the vocal tract. Both of the components are inherently time-varying and assumed to be independent of each other.

The Voice Source

Let us return to the phonation mechanism. According to the source-filter model, there are two possible voice sources: (1) a periodic stream of air puffs which emerges from the vibration of vocal folds as described in Section 3.1.2, and (2) non-periodic turbulent airflow which results when vocal folds are open. Periodic voice source is characteristic for all vowels and nasal. Non-periodic turbulence, in turn, is the acoustic input for noise-like sounds such as fricatives [f] and [s]. A coarse classification of speech sounds can be based on the phonation type, which is referred either to as *voiced* or *unvoiced* for the two voice source types, respectively.

Examples of voiced and unvoiced sounds are shown in Fig. 3.8. For voiced sounds, a quasi-periodic structure is apparently present in the waveform. The periodicity results in a *harmonic spectrum*, in which most of the acoustic energy is distributed on the integer multiplies of the fundamental frequency, i.e. $kF0, k = 1, 2, 3, \ldots$ For instance, the harmonics of $F0$ can be seen clearly in the spectrum in Fig. 3.5. The unvoiced sounds do not have a periodical structure, and their spectra is non-harmonic and often spread on wider frequency range.

The Acoustic Filter

In the source filter model, the vocal tract is considered as an acoustic filter which is characterized by its natural resonance frequencies [23, 67, 58, 85]. In the neighborhood of the resonances, the frequencies of the source signal are boosted. For voiced sounds, these local maxima of the spectrum are called *formants* and they can be seen in spectrograms as dark areas. The formants are numbered as $F1$, $F2$ and so on. For most of the vowel sounds, the first two formants bear most of the phonetic information [23, 108].

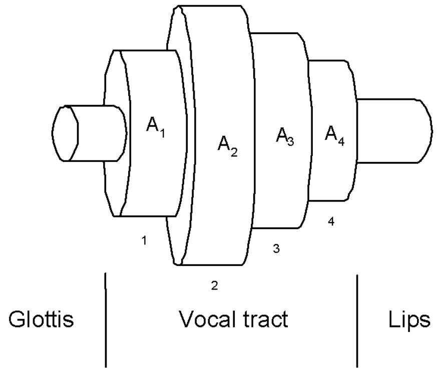

Figure 3.9: Multitube model of the vocal tract.

Often the acoustic filter is modeled as a hard-walled *tube resonator* (see Fig. 3.9). In this so-called *lossless tube model*, the vocal tract is considered as a cascade of N lossless tubes with varying cross-sectional areas. For this kind of resonator, the resonances can be computed analytically. In the case of a single tube ($N = 1$), the resonances of the tube (formant frequencies) are given by the following equation [51]:

$$F_n = \frac{(2n-1)c}{4l}, \tag{3.1}$$

where F_n is the nth formant frequency [Hz], c is the speed of sound in air [m/s], and l is the total length of the tube [m]. Actually the single-tube model predicts fairly well the formant frequencies of the neutral vowel, since during its articulation, the cross-sectional area along the vocal tract is approximately constant. For an average adult male speaker (l=17.5 cm) the formants of the neutral vowel would be predicted by the equation occurring at 500 Hz, 1500 Hz, 2500 Hz and so on.

Spectrogram and formant tracks of the neutral vowel in the word "bird" uttered by the author are shown in the upper and lower panels of Fig. 3.10, respectively. It can be seen that the tube model predicts the formant locations quite accurately, especially for the formants $F1$, $F2$ and $F4$ in this case. The differences are due to assumption of lossless tube and the length of the tube.

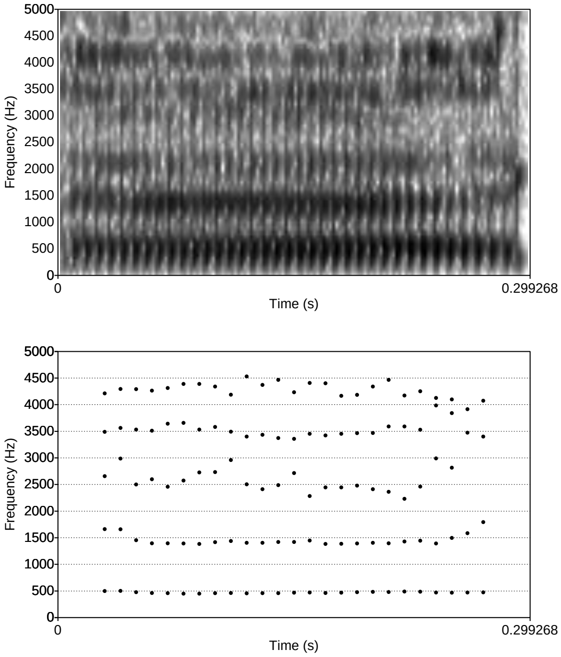

Figure 3.10: Spectrogram and formant tracks of the neutral vowel in "heard" uttered by the author.

The production of other vowels can be modeled by a three-tube model [36], where the first tube from the glottis corresponds to the pharyngeal cavity, the last tube corresponds to the oral cavity, and the middle tube represents the place of the major constriction. The place of the constriction defines the phonetic quality of the vowel.

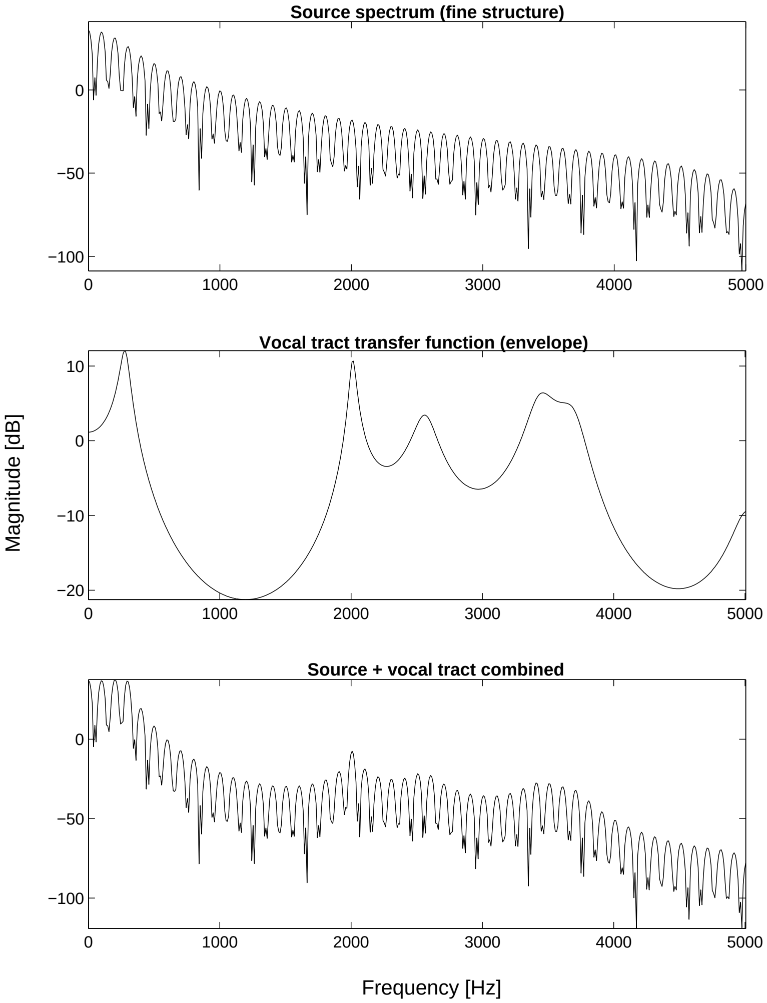

Figure 3.11: Example of the source filter model ($F0$ = 100 Hz, transfer function of [i] estimated via linear prediction).

The the tube model of Fig. 3.9 is at the best a crude approximation of the actual physiology of the vocal tract, since (1) vocal tract is not a cascade of discrete hard-walled tubes but the cross-sectional areas vary continuously, (2) there are energy losses in the vocal tract due to vibration of cavity walls, friction and heat conduction [51]. Furthermore, (3) in nasal sounds, there is a side tube (closed oral cavity) which is not taken into account in the modeling.

Spectra of the Source and the Filter

According to source-filter theory, the resulting spectrum of a cascade of source and filter is the product of their spectra:

$$S(z) = U(z)H(z), \tag{3.2}$$

where $S(z)$ is the speech spectrum, $U(z)$ is the source spectrum and $H(z)$ is the transfer function of the vocal tract filter. In other words, the filter emphasizes the source frequencies around the vocal tract resonances. This is illustrated in Fig. 3.11. Notice that the source spectrum is responsible for generating the harmonic fine structure and spectrum roll-off (spectral tilt), whereas the vocal tract transfer function modifies the overall spectral envelope.

The key point in the source-filter model is that the source and filter are *independent* of each other. The assumption of the uncoupling makes it possible to separate the two components from each other. Although being a reasonable assumption in some cases, this is not true in general [36, 88, 150]. During phonation, the output of the larynx is affected by the vocal tract, and this interaction of the source and the filter can be seen especially in the cases when the first formant is low [88]. Vowels that have a low $F1$ tend to have higher fundamental frequency.

3.2.3 Segmentals and Suprasegmentals

The terms *segmental* and *suprasegmental* refer to the time span of the acoustic analysis. Segmental measurements are done for a short segment of speech, e.g. for a single phoneme. The duration for typical segmental analysis is in the order of milliseconds. An example of a segmental measurement is the spectrum of a single phoneme.

Suprasegmental parameters, on the other hand, are spread over several segments as the name suggests. Suprasegmental parameters are also called *prosodic* parameters. Prosodics is responsible for controlling the intonation, stress, and rhythmic organization of the speech [23, 85]. Typical acoustic suprasegmental measurements are the intensity and $F0$ contours. An example is shown in Fig. 3.12.

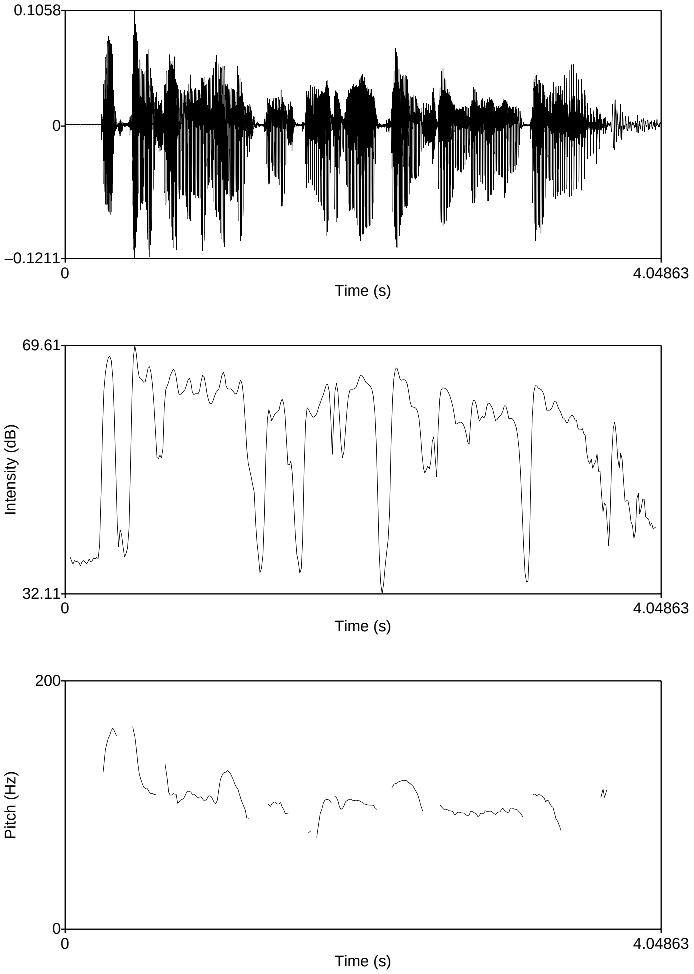

Figure 3.12: Example of acoustic analysis of prosodic parameters. Waveform (upper panel), intensity contour (middle panel) and $F0$ contour (bottom panel).

Suprasegmentals in the spoken language have a somewhat different function in the spoken language compared to the segmentals. In our everyday life, suprasegmental features are used, for instance, to signal our attitudes and emotions to the listener. Suprasegmental features can also give a clue of the speaker's dialect, social status, and the language spoken. Also, in so-called *tonal languages*, the pitch contour plays also role in defining the linguistic message [85].

It is obvious that the prosodic strand of speech bears also information about the speaker itself. However, the prosodic parameters can be quite easily impersonated. For instance, in [4] a popular Israel imitator imitated three different politicians. Three different parameters were studied: pitch contours, formant frequencies and cross-sectional tube areas derived from the lossless tube model of the vocal tract. It was demonstrated that the prosodic parameter, i.e. pitch contour of the imitator, matched very closely to the original target pitch contour. Interestingly, also the second and third formant of the imitator matched closely to the target speaker's formants. The first formant did not match as closely. Even the cross-sectional areas were changed towards the target utterance, but not as much as the pitch contour which was most easily modified. Although Ashour and Gath's study was a preliminary one (small number of speakers and parameters), it gives an unfortunate example that even the vocal tract parameters can be modified.

The main motivation for using prosodics in speaker recognition systems is their robustness against transmission channel [134]. This makes the features attractive for telephone-based applications.

3.3 Perceptual Approach

The last viewpoint of speech communication considers the *perception* of speech, i.e. how human listener's auditory system processes speech sounds. The discipline of sound perception in general is referred to as *psychoacoustics*. Techniques adopted from psychoacoustics are extensively used in audio- and speech processing systems for reducing the amount of perceptually irrelevant data. Psychoacoustics aims at finding connections between the physical, objectively measurable auditory stimuli, and the subjective impression about what the listener has about the stimuli. A few typical acoustic attributes and their auditory counterparts are listed in Table 3.1. We will discuss the first three attributes in more detail since the underlying psychoacoustic principles of these are widely used in speaker recognition systems.

Table 3.1: Some physical attributes and their perceptual counterparts [58].

Physical attribute	**Perceptual attribute**
Intensity	Loudness
Fundamental frequency	Pitch
Spectral shape	Timbre
Onset/offset time	Timing
Phase difference in binaural hearing	Location

The *loudness* of a sound is not linearly proportional to the measured sound intensity. For instance, if the sound intensity is doubled, it is not perceived "twice as loud" in general. The *decibel scale (dB)* is a more convenient way of describing this relationship. Decibel scale is a means of comparing intensities of two sounds [58]:

$$10 \log_{10} \left(\frac{I}{I_0} \right), \tag{3.3}$$

where I_0 is the intensity of the reference sound being compared with. For instance, if the intensity I is twice the reference intensity I_0, its dB level is approximately +3 dB.

Fundamental frequency ($F0$) is defined as the rate at which the vocal folds vibrate during voiced phonation. Psychoacousticians call perceived $F0$ *pitch*. Even if a speech signal is filtered so that the frequency region of the fundamental is not present in the signal, humans can perceive it [67].

The human ear processes fundamental frequency on a logarithmic scale rather than a linear scale [67]. It has been observed that in the high frequencies, the $F0$ must change more that a human listener can hear a difference between two tones. *Mel* is a unit of perceived fundamental frequency. It was originally determined by listening tests, and several analytic models have been proposed for approximating the mel-scale. For instance, Fant [36] has proposed the following mapping:

$$F_{mel} = 1000 \log_2 \left(1 + \frac{F_{Hz}}{1000} \right). \tag{3.4}$$

Although $F0$ and pitch are in principle different quantities, they are used interchangeably in literature to refer to the frequency of vocal fold vibration. We do not make a difference between them here either.

The relative amplitudes of different frequencies determine the overall *spectral shape*. If the fundamental frequency is kept the same and the relative amplitudes of the upper harmonics are changed, the sound will be perceived as having different *timbre*. Thus, timbre is the perceptual attribute of the spectral shape, which is known to be an important feature in speaker recognition. For instance, the widely used mel-cepstrum feature set measures the perceptual spectral shape.

Studies of the human hearing mechanism [124] show that in the early phases of the human peripheral auditory system, the input stimulus is split into several frequency bands within which two frequencies are not distinguishable. These frequency bands are referred to as *critical bands*. The ear averages the energies of the frequencies within each critical band and thus forms a compressed representation of the original stimulus. This observation has given impetus for designing perceptually motivated filter banks as front-ends for speech and speaker recognition systems.

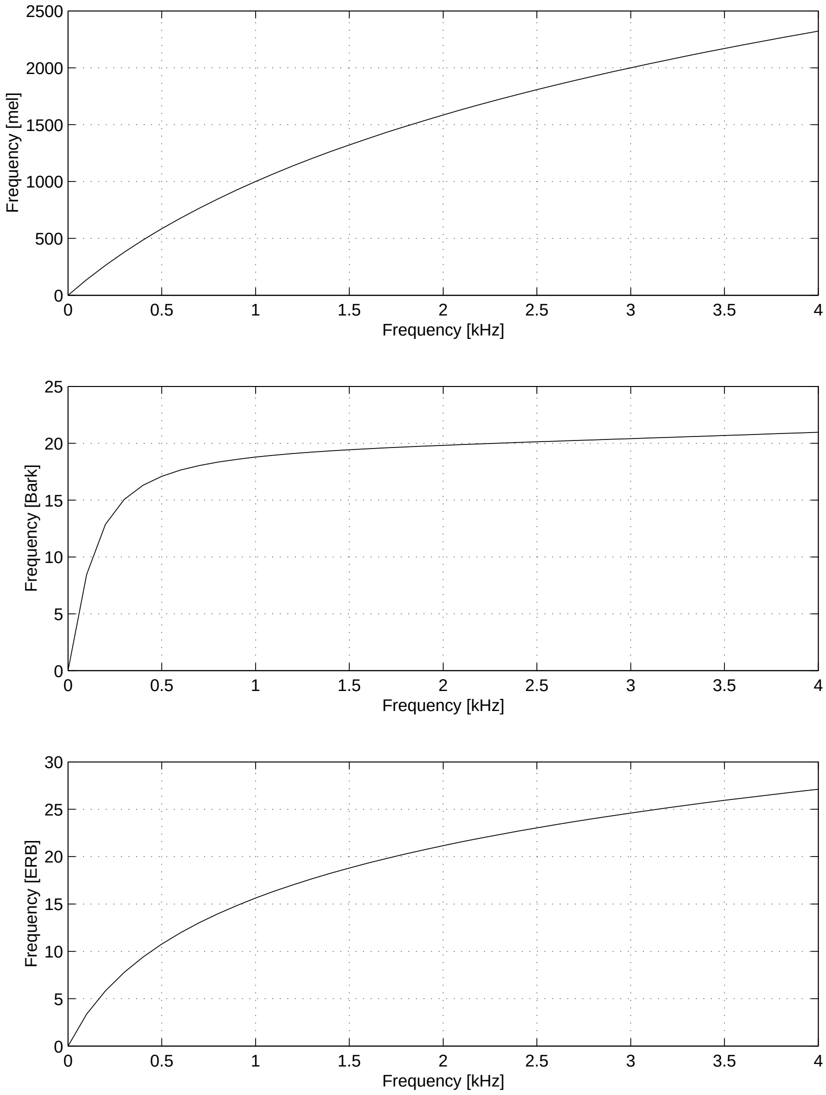

Figure 3.13: Mel-, Bark- and ERB-scales.

Many approximations to the critical band scale have been proposed. A well-known mapping is the *Bark-scale* [51]. For the Bark scale, several analytical formulae have been proposed. One of them is the one proposed by Zwicker and Terhardt [171]:

$$F_{Bark} = 13\tan^{-1}\left(\frac{0.76F_{Hz}}{1000}\right) + 3.5\tan^{-1}\left(\frac{F_{Hz}}{7500}\right)^2. \tag{3.5}$$

Another example of Bark-scale approximation is the following:

$$F_{Bark} = 6\sinh^{-1}\left(\frac{F_{Hz}}{600}\right). \tag{3.6}$$

In addition to the Bark-scale, another critical band scale is so-called *ERB-scale*[3] [51], which is defined as follows:

$$ERB = 21.4 \log_{10} \left(1 + \frac{4.37 F_{Hz}}{1000}\right). \tag{3.7}$$

Mel, Bark and ERB-scales are depicted in Fig. 3.13 for comparison. The shapes of the curves are different, but the message of all three is the same. In the higher frequency region, two different stimuli must have larger difference in order the human ear to distinguish them. In the lower frequencies, the spectral resolution of the human ear is higher.

Given the center frequency of a critical band, its bandwidth can be computed as follows [125]:

$$BW = 25 + 75\left(1 + 1.4\left(\frac{F_{Hz}}{1000}\right)^2\right) \tag{3.8}$$

.

One should question the usefulness of perceptual frequency scales in speaker recognition. Perceptually motivated representations have been used successfully in speech recognition, and a little ironically, in speaker recognition as well, despite the opposite nature of the tasks. The implicit assumption made when using psychoacoustical representations is that the human ear is the optimal recognizer. If this is not true, then we are throwing useful information away!

3.4 Speaker Individuality

Speaker individuality is a complex phenomenon which builds up from both the anatomy of the speaker's vocal organs, as well as learned traits. There has been debate about the inadequateness of the binary division into physiological (organic) and learned (functional) speaker cues [112, 139]. According to Nolan [112], no acoustic feature escapes the plasticity of the vocal tract. In other words, the vocal organs are not fixed but they can be altered intentionally, and rather than being a static organ, there exists *limits* within which the variation of the organs can take place. In this sense, voice is a different biometric than fingerprint, for instance. Fingerprint[4] remains the same but the speech signal varies from time to time.

[3] ERB = equivalent rectangular bandwidth of the auditory filter

[4] The very famous and as much misleading term *voiceprint* coined by Kersta [69] is still widely used in daily media. As Bonastre & al. [17] point out, a voiceprint is nothing but a printed spectrogram, and it does not contain robust feature points as fingerprints.

3.4.1 The Voice Source

The larynx is quite individual. It is a well-known fact that females and children have smaller vocal folds, and as a result, their overall pitch is higher than an adult male's. There is also variation between individuals in the pitch distributions of each gender. The long-term $F0$ statistics, especially its mean [101, 112, 22] and median [63] values carry important speaker information.

The fundamental frequency alone carries only one source of individuality in the voicing mechanism. Tension of the vocal folds affects directly the glottal pulse parameters, such as the rate of the closing phase, and the degree of the opening. For some speakers, there is a complete glottal closure in the voicing. For some others, the glottis is never complete closed, and the auditive impression is perhaps a "breathy" voice [85] (see Fig. 3.3). Due to this intuitively appealing notion of speaker differences in voice quality, it seems beneficial to exploit voice quality information in speaker recognition systems. However, reliable measurement of voice quality is difficult [34].

The shape of the glottal pulse affects for instance *spectral tilt*, the overall downward slope of the spectrum. Spectral tilt can be estimated from the long-term spectrum as the ratio of the energy of the higher frequency band to the energy of the lower frequency band [112].

Since the glottal flow is modified by the filtering effect of the vocal tract, direct measurement of the glottal flow is not possible. Plumpe & al. [126] used inverse filtering based on linear prediction in order to obtain an estimate of the glottal flow. Then, "coarse" structure of the glottal flow derivative waveform was estimated by finding the parameters of the so-called *Liljencrants-Fant glottal model* by an iterative gradient search procedure. From the difference between the coarse model and the actual glottal waveform, "detail" features were then derived. It appeared that although the glottal features were observed to contain useful speaker-related information, their measurement is difficult, especially from noisy speech.

3.4.2 The Vocal tract

Vocal tracts of individuals differ, first of all, in their overall size [108]. This is especially true between genres. Vocal tract sizes are progressively smaller for male, female and children, respectively. If we assume that the articulatory configurations of two speakers are the same and the only difference is the length of the vocal tract (measured from glottis to lips), then the acoustic theory predicts that the formant frequencies are inversely scaled by the ratio of the speakers' vocal tract lengths.

In addition to the overall size of the vocal tract, the relative sizes of the individual cavities differ between individuals [108]. Oral and pharyngeal parts of the vocal tract are scaled differently from speaker to speaker. The length of the mouth cavity does not vary as much as that of the pharyngeal cavity between speakers. Also, the front and back cavities as defined by the major constriction of articulation have been found to be correlated with the "phonetic" and "speaker" features. Mokhtari [108] also cites a study where it was found out that the regions of the vocal tract with larger cross-sectional area are more individual than the places of lingual constriction. In summary, both the length and the shape of the vocal tract are individual.

3.4.3 Segmental Differences

The discriminatory properties of different phonemes and phoneme groups have been studied. Eatock and Mason [32] compared discrimination properties of phoneme groups and individual phonemes from a database of 125 speakers. The speech files were annotated by hand, and the extracted segments were parametrized using LPC-derived cepstral coefficients. They found out that the nasals and vowels performed the best and stops the worst, and this was in consensus with previous studies carried out by different authors. The only exception was the unvoiced fricative [s], which Eatock and Mason found to be comparable with the vowels and nasals, which was not the case in the studies they cite.

Kajarekar and Hermansky [68] introduced an automatic speaker verification system that segmented the input speech into four broad phone categories: (1) silence+stops, (2) glides+nasals, (3) vowels+diphtongs, and (4) fricatives. They used two large corpora (539 and 1003 speakers) in their experiments. The segmentation was carried out using Hidden Markov Model (HMM) approach, and the spectral features were mel-cepstral coefficients. The lowest equal error rate (EER) was obtained with the combination of categories (3) and (4). The highest EER was obtained by using the category (1) alone. When used individually, the fricative category gave lowest EER. Although the studies [68, 32] are not directly comparable, the results are consistent: vowels, nasals and fricatives have good discrimination properties, whereas stops have small inter-speaker variation.

It has been reported in several studies that nasal sounds are an effective speaker cue [142, 112, 32, 68, 139]. This can be partially explained by the fact that the nasal cavity is both quite individual, and more importantly, fixed in the sense that one cannot change its volume or shape. Therefore, the measurement from the nasal sounds are expected to remain quite stationary in different recording sessions. However, since the pharyngeal and oral cavities are part of the acoustic resonator in nasal production, there will necessary be some variation. Moreover, nasal sounds are easily affected by e.g. head cold [142].

Vowels have been studied a lot due to their high occurrence in European languages [85]. The general agreement is that the first two formants are mostly responsible for the phonetic quality of vowels. In other words, a vowel can be (almost) uniquely determined by only two acoustic parameters, $F1$ and $F2$. The third and higher formants, on the other hand, are assumed to be more speaker-dependent. Both acoustic and perceptual studies have given evidence for this hypothesis according to studies cited by Mokhtari [108]. This observation agrees with the claim regarding the speaker specificity of the pharyngeal cavity: above larynx, there is so-called "larynx tube" that generates a larynx resonance whose frequency is generally higher than those of $F1$ and $F2$.

Absolute locations of lower formants are often not the same for different speakers producing the same vowel. Instead, the *relative* locations (e.g. relative to neutral vowel or other reference sound) seem to have less variation between speakers [108]. Thus, the lower vowel formants also carry information about the speaker, although their inter-speaker variation is smaller than those of the higher vowel formants. Also, the discrimination power of the formants depend on the vowel: for instance, $F2$ of the front vowels carries speaker information [108].

Several studies have indicated the middle- and high-frequency portions of the spectrum to be important for speaker recognition [108, 10, 145]. Also, the low end (approximately the region of $F0$) of the spectrum carries useful information [10, 71]. The lower frequency portion of the spectrum roughly carries most of the phonetic content, i.e. the message. However, phonetic and speaker information are mixed in a complex way over the spectrum [165], and the extraction of speaker information cannot not be done just by a simple band-pass filtering. Also, the discriminative frequency regions depend on the phoneme [10, 71]. See [108] for an extensive treatment of this *speech-speaker dichotomy*.

3.4.4 Suprasegmental Differences

We know from our everyday life by intuition that prosodic parameters carry speaker-related information: intonation, stress, timing and rhythm vary from speaker to speaker. These parameters are not affected by noise and transmission line as much as the spectral parameters. However, their disadvantage is that they depend on the speaker's emotional state and the spoken utterance. Furthermore, they can be more easily impersonated, and thus they are not considered as reliable as the segmental cues.

Yet another complication associated with the suprasegmentals is their measurement and modeling. To be able to model a suprasegmental strand, e.g. a pitch contour, it must be preceded by some sort of segmental measurement ($F0$ estimation). Then, the time sequence obtained in this way must be modeled somehow.

Historically, prosodic parameters were studied in early studies of automatic speaker recognition [5, 140], but the interest has clearly grown in the past few years [22, 147, 83, 160, 9, 21, 39, 135, 119, 2]. The reasons for this might be the increased computing power of modern microprocessors, and the increased need to develop more robust systems beyond laboratory environments.

Chapter 4

SIGNAL PROCESSING BACKGROUND

This chapter gives a brief overview of the digital signal processing methods used in feature extraction. A comprehensive treatment of the subject can be found in general DSP books such as [114, 62, 129, 146].

4.1 A/D Conversion

A speech signal is a form of wave motion carried by a medium (e.g. air particles) [18, 58], and it can be captured by a microphone, which converts the continuous air pressure changes into continuous voltage changes. The analog signal $s_a(t)$ is then sampled to a digital form $s[n]$ by an *analog-to-digital converter* (A/D converter). The A/D converter samples the analog signal uniformly with the *sampling period* T:

$$s[n] = s_a(nT). \tag{4.1}$$

The inverse of T is the *sampling frequency* (or *sampling rate*) and marked here by $F_s = 1/T$. Given that the original signal $s_a(t)$ contains frequencies only up to $F_s/2$, it can be fully reconstructed from the samples $s[n]$ [114, 129]. The frequency $F_s/2$ is called the *Nyquist rate* [114] of the signal and it is the upper limit for frequencies present in the digital signal. For instance, if one wants to preserve frequencies up to 4 kHz, the sampling rate must be chosen $F_s > 8$ kHz. In addition to the sampling, the ADC *quantizes* the samples into a finite precision. The number of bits used per sample determines the dynamic range of the signal. Adding one bit extends the dynamic range of the signal rougly +6 dB [62].

4.2 Fourier Analysis

Fourier analysis provides a way of analyzing the *spectral properties* of a given signal in the frequency domain. For instance, the spectrograms in Figures 3.6 and 3.7 were produced by computing windowed discrete Fourier Transform of the speech signal.

The Fourier analysis tools consider a signal as being composed of a superposition of *sinusoidal* basis functions of different frequencies, phases and amplitudes. An example is shown in Fig. 4.1, which shows three sinusoids and their superposition (sum). Fourier analysis provides a tool for finding the parameters of the underlying sinusoids (*forward transform*) or for synthesizing the original time-domain signal from the frequency domain presentation (*inverse transform*).

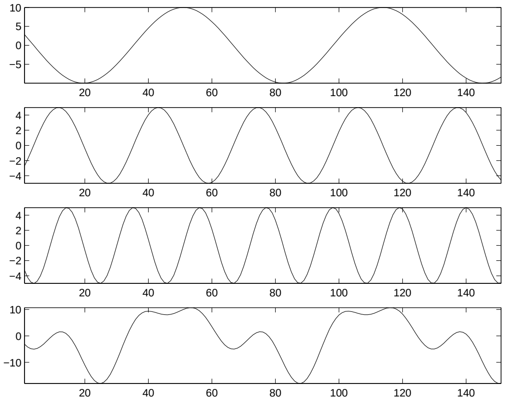

Figure 4.1: Three sinusoids and their superposition.

4.2.1 The Discrete Fourier Transform (DFT)

Suppose that $s[n], n = 0, 1, \ldots, N-1$ is a discrete-time sequence of N samples. The *discrete Fourier transform* or *DFT* of $s[n]$ is defined as follows [67, 129]:

$$\hat{S}[k] = \mathcal{F}\{s[n]\} = \sum_{n=0}^{N-1} s[n] e^{-j2\pi nk/N}, 0 \leq k \leq N-1, \tag{4.2}$$

where k represents the discrete frequency variable and j is the imaginary unit. The result of the DFT is a complex-valued sequence of length N. The value $k = 0$ ($\omega_k = 0$) corresponds to *zero frequency* or the *DC component* of the signal and $k = N/2$ ($\omega_k = \pi$) corresponds to the Nyquist frequency.

The *inverse DFT* or *IDFT* is defined as

$$s[n] = \mathcal{F}^{-1}\{\hat{S}[k]\} = \frac{1}{N}\sum_{k=0}^{N-1} \hat{S}[k]e^{j2\pi nk/N}, 0 \leq n \leq N-1, \tag{4.3}$$

where n represents the discrete time variable. In other words, the original signal can be reconstructed from its Fourier transform by the inverse transform. Both DFT and IDFT are linear transformations, i.e.

$$\begin{aligned}\mathcal{F}\{\alpha s_1[n] + \beta s_2[n]\} &= \alpha\mathcal{F}\{s_1[n]\} + \beta\mathcal{F}\{s_2[n]\}\\ \mathcal{F}^{-1}\{\alpha s_1[n] + \beta s_2[n]\} &= \alpha\mathcal{F}^{-1}\{s_1[n]\} + \beta\mathcal{F}^{-1}\{s_2[n]\}\end{aligned}$$

for all constants α, β and sequences $s_1[n]$, $s_2[n]$.

4.2.2 The Magnitude and Phase Spectra

The kth harmonic component of the DFT is a complex number $\hat{S}[k] = \hat{S}_{\mathrm{Re}}[k] + j\ \hat{S}_{\mathrm{Im}}[k]$. It can be expressed in polar form as $\hat{S}[k] = |\hat{S}[k]|e^{j\angle\hat{S}[k]}$, where

$$\begin{aligned}|\hat{S}[k]| &= \sqrt{\hat{S}_{\mathrm{Re}}[k]^2 + \hat{S}_{\mathrm{Im}}[k]^2}\\ \angle\hat{S}[k] &= \tan^{-1}\left(\frac{\hat{S}_{\mathrm{Im}}[k]}{\hat{S}_{\mathrm{Re}}[k]}\right).\end{aligned}$$

$|\hat{S}[k]|$ is the *magnitude* and $\angle\hat{S}[k]$ is the *phase* of the kth harmonic component.

DFT is periodic with N, i.e. $\hat{S}[k+N] = \hat{S}[k]$. For real signals such as speech, the magnitude spectrum is symmetric with respect to the frequency $N/2$. Furthermore, the phase spectrum for real signals is antisymmetric with respect to the frequency $N/2$. Due to this redundancy, any real-valued signal is fully represented by the harmonic components up to $N/2$.

In speech analysis, the phase spectrum is usually neglected, since it is generally believed that it has little effect on the perception of speech [51, 44]. However, some studies have indicated that the phase actually *is* important for perception of speech. For instance, recently Paliwal and Alsteris [117] demonstrated that the phase spectrum is actually more important than the magnitude spectrum for speech perception in certain conditions! This raises a question whether phase information should be exploited for speech and speaker recognition front-ends.

4.2.3 Fast Fourier Transform (FFT)

From the definition (4.2) it is easy to see that the time complexity of DFT is $O(N^2)$. However, DFT can be computed via a faster algorithm called *fast Fourier transform* or *FFT* [62]. A requirement for FFT is that input signal (vector) has a length of 2^M for some $M \in \mathbb{N}_+$, i.e. a power of two. In practice, the input signal is first *zero-padded* to the next highest power of two and the zero-padded signal is given as an input for the FFT. For instance, if the length of signal is 230 samples, it is zero padded to length $N = 256$ for which the FFT can be computed. Zeros can be added to the beginning or end of the signal, and it does not affect the result of the DFT [129]. Time complexity of the FFT is $O(N \log_2 N)$. The savings in computation time in practise are on the order of hundred folds. For instance, for $N = 1024$, the ratio of DFT multiplications to FFT multiplications is about 200 and the ratio of additions about 100 [62].

4.3 Digital Filters

A *filter* is a system that modifies the input signal $s[n]$ into output signal $y[n]$ [62, 129, 114]. There are several ways of specifying a digital filter. In the time domain, filter is characterized by its *impulse response* $h[n]$ that can be finite (*FIR*-filter) or infinite (*IIR*-filter). In the frequency domain, filter can be specified by its *transfer function* $H(z)$, where z is a complex variable.

In the time domain, filtering is presented as a *convolution* between the input signal and the impulse response $h[n]$ [58, 62, 129, 114]:

$$y[n] = s[n] \star h[n] = \sum_{k=-\infty}^{\infty} s[k]h[n-k]. \tag{4.4}$$

In practise, this is implemented using a recursive relationship [62]:

$$y[n] = \sum_{k=0}^{N} a[k]s[n-k] - \sum_{k=1}^{M} b[k]y[n-k], \tag{4.5}$$

where the coefficients $a[k], b[k]$ are determined from the filter specifications. The latter sum in (4.5) represents the feedback part of the filter, and it vanishes for FIR filters ($b[k] = 0$ for all k). The *transfer function* $H(z)$ of (4.5) is obtained by taking z-transforms of both sides and solving for $Y(z)/S(z)$:

$$H(z) = \frac{Y(z)}{S(z)} = \frac{\sum_{k=0}^{N} a[k]z^{-k}}{1 + \sum_{k=0}^{M} b[k]z^{-k}}. \tag{4.6}$$

The roots of the numerator of (4.6) are called *zeros* of the system, and the roots of the denominator are called *poles* of the system. A pole causes a *resonance* (peak) in the magnitude response of the filter, whereas a zero causes an *anti-resonance* (valley). For instance, the vocal tract transfer function of vowel sounds can be well characterized by its poles only, which correspond to the formant locations. On the other hand, the nasal sounds such as [n] have, in addition to resonances, anti-resonances in their spectrum, and both the poles and zeros are needed in modeling [51].

In the frequency domain, filtering is performed by multiplying the DFT of the input signal pointwise by the transfer function of the filter. According to the *convolution theorem* [114], multiplication in the frequency domain corresponds to convolution in the time domain, and vice versa:

$$s[n]\star\, h[n] \quad\longleftrightarrow\quad S(z)H(z) \tag{4.7}$$

$$s[n]h[n] \quad\longleftrightarrow\quad S(z)\star H(z). \tag{4.8}$$

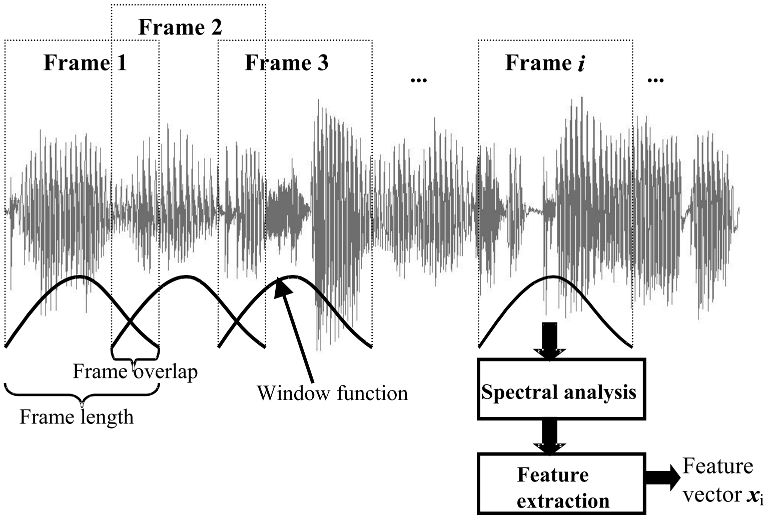

Figure 4.2: Short-term spectral analysis.

4.4 Short-Term Spectral Analysis

Since the speech signal changes continously due to the articulatory movements of the vocal production organs, the signal must be processed in short segments, within which the parameters remain quasi-stationary (see Fig. 4.2). Computing the DFT over the entire signal would discard the local spectral properties which emerge from the realizations of different phonemes. Instead of performing DFT for the whole signal, a *windowed DFT* is computed. A short *frame*, typically around 10-30 milliseconds, is multiplied by a *window function*, and the DFT of the windowed frame is then computed. This process is repeated over the entire speech signal so that the frame is shifted forward by a fixed amount, typically around 30 to 75 % of the frame length.

4.4.1 Window Functions

The purpose of the windowing is to reduce the effect of the spectral artefacts that result from the framing process [52, 67, 129, 114]. Windowing in the time domain is a pointwise multiplication of the frame and the window function. According to the *convolution theorem* [114], this corresponds to convolution of the short-term spectrum with the window function magnitude response. In other words, the transfer function of the window will be present in the observed spectrum. A good window function has a narrow *main lobe* and low *sidelobe* levels [129, 52] in their transfer functions. There is a trade-off between these two: making the main lobe narrower increases the side-lobe levels, and vice versa. Harris [52] lists also several other desirable properties of a good window function. In general, a proper window function tapers smoothly down at the edges of the frame so that the effect of the discontinuities is diminished.

The intuitively most simple windowing is “no windowing”, or *rectangular window* defined as follows [58]:

$$w[n] = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases} \tag{4.9}$$

Although rectangular window preserves the original waveform unchanged, it is seldom used due to its poor *spectral leakage* effects. The most commonly used window function in speech processing is the *Hamming window* defined as follows [58]:

$$w[n] = \begin{cases} 0.54 - 0.46\cos\frac{2\pi n}{N}, & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases} \tag{4.10}$$

The time-domain shapes and magnitude responses (computed using DFT) of rectangular and Hamming windows are shown in Fig. 4.3.

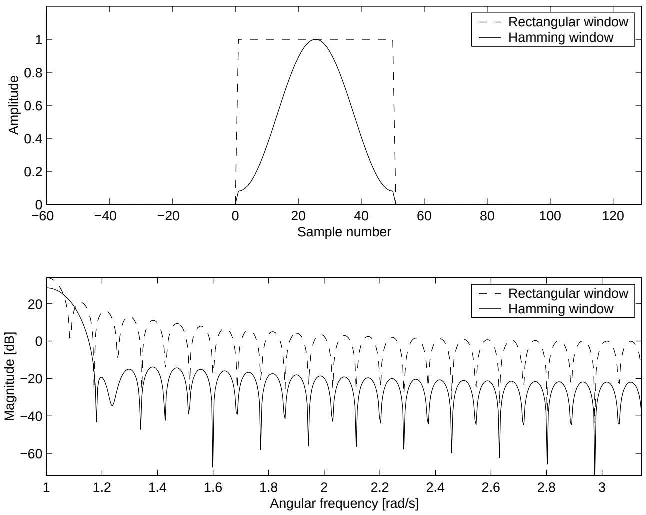

Figure 4.3: Waveforms and magnitude responses of rectangular and Hamming windows estimated by the DFT.

Harris [52] has compared over 20 different window functions. One window that has a good compromise between the main lobe width and the sidelobe levels is the *Kaiser-Bessel* window defined as follows:

$$w[n] = \begin{cases} \frac{I_0\left[\pi\alpha\left(1-(2n/N)^2\right)^{\frac{1}{2}}\right]}{I_0(\pi\alpha)}, & -N/2 \leq n \leq N/2 \\ 0, & \text{otherwise,} \end{cases} \tag{4.11}$$

where

$$I_0(x) = \sum_{k=0}^{\infty} \left[\frac{(x/2)^k}{k!}\right]^2. \tag{4.12}$$

Although the sum (4.12) is in theory infinite, in practise it converges very fast. According to Ifeachor and Jervis [62] a 32-term partial sum is enough to approximate (4.12). The parameter α controls the trade-off between the main lobe width and the level of the sidelobes. For smaller α, the main lobe is more narrow but the sidelobes levels are higher, and vice versa. Typical values are from $\alpha = 2.0$ to $\alpha = 3.5$ [52].

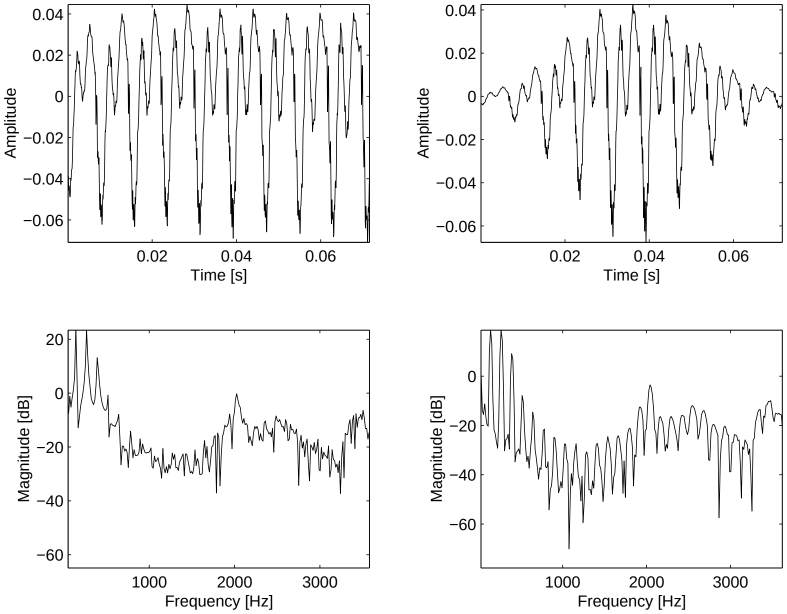

Figure 4.4: Voiced speech segment [i] windowed using rectangular (left) and Hamming (right) windows. Notice the spectral leakage in the case of the rectangular window.

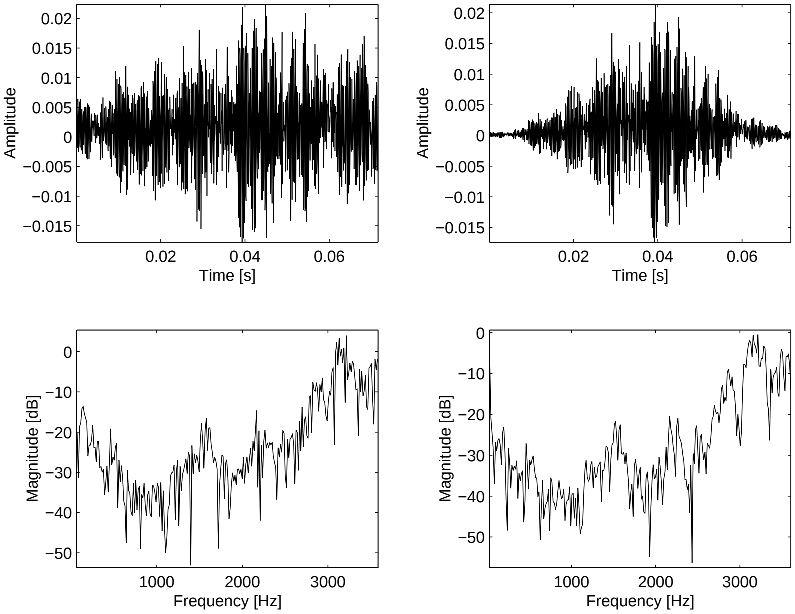

Figure 4.5: Unvoiced speech segment [s] windowed using rectangular (left) and Hamming (right) windows.

Examples of windowing are shown in Figures 4.4 and 4.5 which show a voiced and an unvoiced speech segment spoken by the same male speaker. As can be seen from the voiced speech segment, the Hamming window exhibits less spectral leakage. The rectangular window causes the harmonics of $F0$ to "leak" energy to the neighboring harmonics, and as the result, the individual harmonics are smoothed away. For the Hamming window, the individual harmonics can be seen. The unvoiced segment in Fig. 4.5 does not contain any harmonics, but the spectral leakage can still be seen.

4.4.2 Frame Length and Overlap

The selection of the frame length is a crucial parameter for successful spectral analysis, due to the trade-off between the time and frequency resolutions. The window should be long enough for adequate frequency resolution, but on the other hand, it should be short enough so that it would capture the local spectral properties. Typically a frame length of 10-30 milliseconds is used. For females and children, pitch tends to be higher, and a shorter frame should be used than for low-pitched male speakers [58]. Usually adjacent frames are overlapping by some amount. A typical frame overlap is around 30 to 50 % of the frame size. The purpose of the overlapping analysis is that each speech sound of the input sequence would be approximately centered at some frame.

4.4.3 Adaptive Framing

Typically, frames have a fixed length because of the straightforward implementation. However, fixed-length frame does not take into account neither the natural variation in the durations of speech sounds, nor the coarticulation effects. Methods that adapt the frame length according to the local pitch period are known as *pitch-synchronous analysis* methods [58]. The motivation for pitch-synchronous analysis is the following. During the open phase of voicing, the trachea, larynx, and the vocal tract are acoustically coupled, and this is affects the resonances of the vocal tract [58, 150]. Therefore, the assumption of the independence of the source and the filter is no longer valid. During the closed phase of phonation, the resonances of the vocal tract are less affected by the source. However, reliable closed-phase analysis, especially in the presence of noise, is a demanding task.

Because the spectrum within the analysis frame should remain approximately stationary, one could use a simple on-line segmentation algorithm based on some acoustic features. This class of algorithms is known as *variable frame rate analysis* (VFR) methods. For instance, in [168] a simple algorithm using VFR for speech recognition was proposed. The algorithm computes weighted Euclidean distances between the adjacent frame parameters computed with initially high frame rate. When the accumulated distance exceeds a pre-determined threshold, a frame boundary is marked. Another example is given by Adami and Hermansky [1]. They proposed a simple segmentation method that utilizes dynamics of F0 and intensity in order to locate the boundaries of broad phonetic categories.

More complex approaches to VFR have been proposed. For instance, Nguyen & al. [109] proposed to use the *temporal decomposition* (TD) on the line spectral frequency (LSF) parameters. The idea in TD [7] is to represent the original speech waveform as a weighted sum of *event functions* that overlap in time with each other. The TD method finds the parameters of the event functions, and the events itself present spectrally stable "phoneme-like" events.

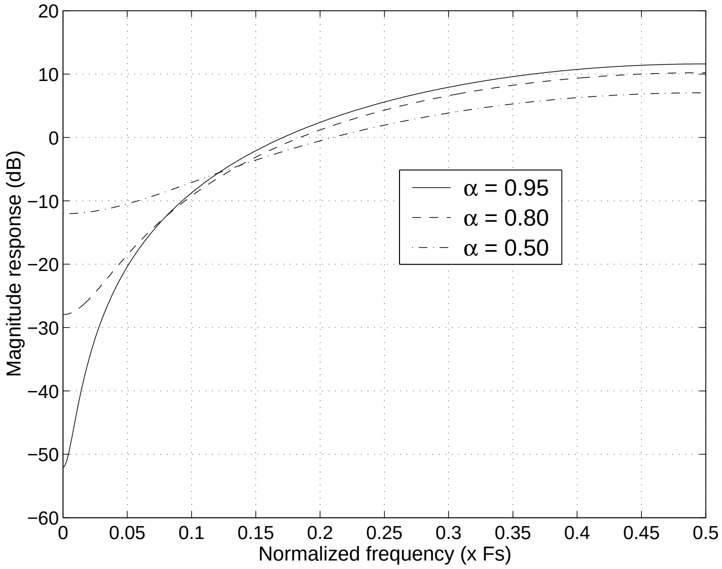

Figure 4.6: Magnitude responses of pre-emphasis filter (4.13) for different values of α.

4.5 Speech Pre-Emphasis

Usually speech is *pre-emphasized* before any further processing. Pre-emphasis refers to filtering that emphasizes the higher frequencies. Its purpose is to balance the spectrum of voiced sounds that have a steep roll-off in the high frequency region. For voiced sounds, the glottal source has an approximately

-12 dB/octave slope [51]. However, when the acoustic energy radiates from the lips, this causes a roughly +6 dB/octave boost to the spectrum. As a net result, a speech signal when recorded with a microphone from a distance, has approximately a -6 dB/octave slope downward compared to the true spectrum (of the vocal tract). Therefore, pre-emphasis removes some of the glottal effects from the vocal tract parameters. Because the spectrum of unvoiced sounds is already flat, there is no reason to pre-emphasize them [51, 125].

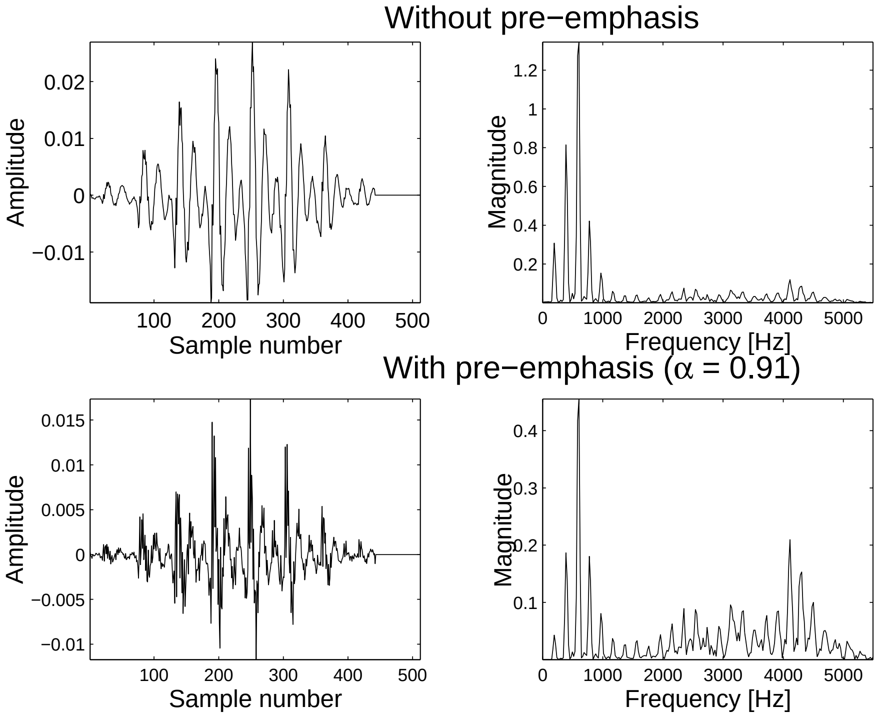

Figure 4.7: Example of pre-emphasis of a single frame (40 ms Hamming window).

Pre-emphasis has other advantages as well. Makhoul [97] shows that the numerical stability of the linear predictive analysis (LP) is inversely proportional to the dynamic range of the spectrum being analyzed by LPC. Therefore, a filter that flattens the spectrum should be used prior to LPC to avoid numerical problems, and this is what the pre-emphasis filter does.

The most commonly used pre-emphasis filter is given by the following transfer function:

$$H(z) = 1 - \alpha z^{-1} \tag{4.13}$$

where $\alpha > 0$ controls the slope of the filter. The impulse response of the filter is $h[n] = \{1, -\alpha\}$ and the filter is simply implemented as a first order differentiator:

$$y[n] = s[n] - \alpha s[n-1]. \tag{4.14}$$

The frequency response of the filter is

$$\begin{aligned} H(e^{j\omega}) &= 1 - \alpha\ e^{-j\omega} \\ &= 1 - \alpha(\cos\omega - j\sin\omega). \end{aligned}$$

Hence, the squared magnitude response is

$$\begin{aligned} |H(e^{j\omega})|^2 &= (1 - \alpha\cos\omega)^2 + \alpha^2\sin^2\omega \\ &= 1 - 2\alpha\cos\omega + \alpha^2\cos^2\omega + \alpha^2\sin^2\omega \\ &= 1 - 2\alpha\cos\omega + \alpha^2(\cos^2\omega + \sin^2\omega) \\ &= 1 - 2\alpha\cos\omega + \alpha^2. \end{aligned} \tag{4.15}$$

The magnitude responses in dB scale[1] for different values of α are shown in Fig. 4.6. An example of pre-emphasized frame in time and frequency domains is shown in Fig. 4.7. Notice that the pre-emphasis makes the upper harmonics of $F0$ more distinct, and the distribution of energy across the frequency range is more balanced.

4.6 Filterbanks

Filterbank is a generic term which refers to the class of methods that process on multiple frequency bands of a given signal. Terms *filterbank* and *subband processing* refer more or less to the same concept, and we will use them interchangeably. Another branch of signal processing that parallels subband processing closely is *wavelet analysis* [151, 98]. Wavelet-based feature extraction has been experimented in speaker recognition also [122, 154, 162]. Although wavelets may provide a better signal representation compared to the short-term DFT, there are several open questions regarding the selection of the *mother wavelet*, *wavelet decomposition structure*, and the extraction of features from the wavelet transform. All of the approaches [122, 154, 162] are more or less heuristics, without theoretical background.

Examples of two different filterbank magnitude responses are shown in Fig. 4.8. In both cases, the filters are linearly spaced on the frequency range 0-4 kHz. In both figures, the filter that analyzes band 2000-2500 Hz is emphasized to put explicit that a given filter in the filterbank has a zero response outside of its passband.

[1] Magnitude response in dB: $10\log_{10}|H(e^{j\omega})|^2 = 20\log_{10}|H(e^{j\omega})|$

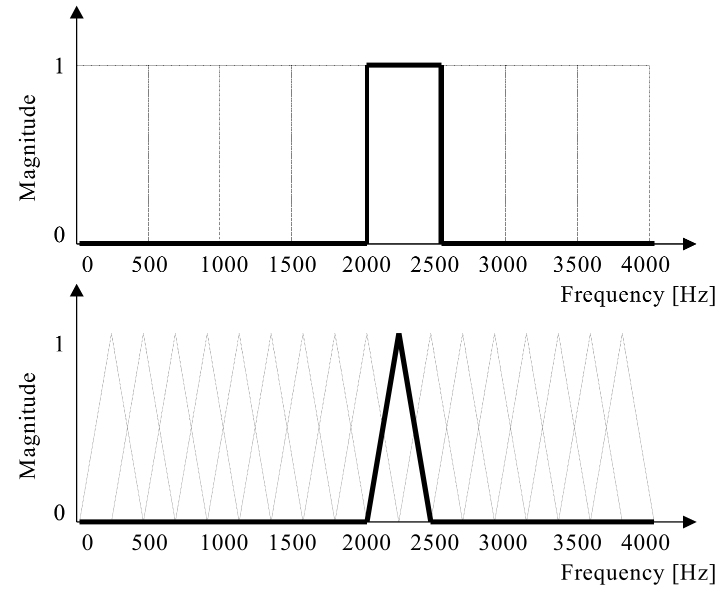

Figure 4.8: Magnitude responses of rectangular- and triangular-shaped filterbanks.

In designing a filterbank, one must decide whether filterbank is implemented in the time domain by set of recursive equations (4.5) or in the frequency domain by multiplying the signal spectrum with the filter magnitude response. In the time-domain implementation [24], the feature extraction can be done for each bandpass signal using the regular frame-by-frame processing (see Fig. 4.9). One advantage of this is that each subband can be processed by the same techniques as for the fullband signal. This has a practical consequence that the resolution of each subband can be controlled more easily than in the fullband processing.

Suppose that an N-point magnitude spectrum $S[j], j = 1, \ldots, N$ is produced by the short-term DFT. Suppose an M-channel filterbank, whose sampled magnitude response is specified in arrays $H_i[j], i = 1, \ldots, M\ ; j = 1, \ldots, N$. The output of the ith filter $Y[i]$ is given by

$$Y[i] = \sum_{j=1}^{N} S[j] H_i[j]. \tag{4.16}$$

In other words, the output of the ith channel is the output of the DFT magnitudes in that frequency region weighted by the filter response. In this way, filterbank provides a smoothed version of the original spectrum with $M < N$ components. Notice that while the DFT of the input frame must be

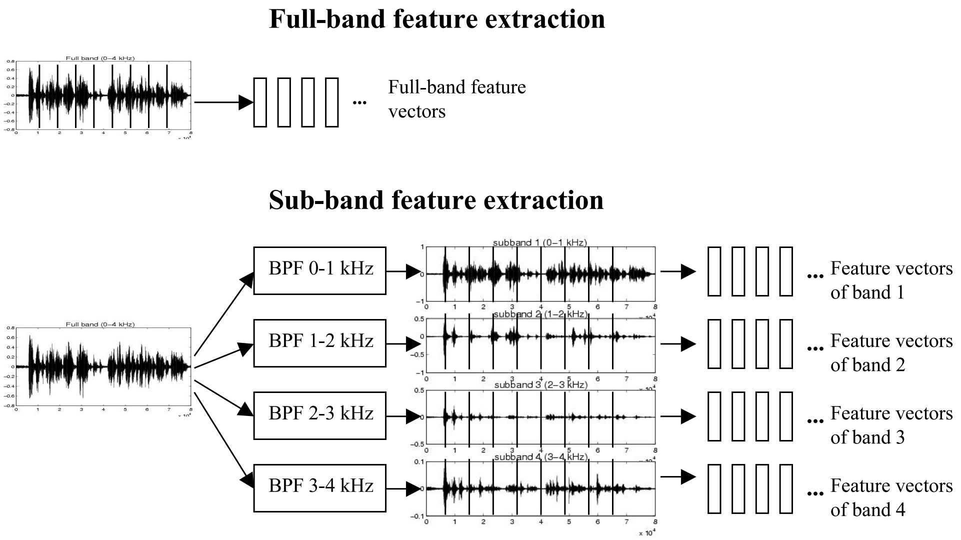
Full-band feature extraction
Full band (0–4 kHz)
...
Full-band feature vectors
Sub-band feature extraction
Full band (0–4 kHz)
BPF 0-1 kHz
BPF 1-2 kHz
BPF 2-3 kHz
BPF 3-4 kHz
subband 1 (0–1 kHz)
subband 2 (1–2 kHz)
subband 3 (2–3 kHz)
subband 4 (3–4 kHz)
... Feature vectors of band 1
... Feature vectors of band 2
... Feature vectors of band 3
... Feature vectors of band 4

Figure 4.9: Illustration of full-band and subband feature extraction.

computed online, the filterbank responses need to be computed only when the recognition system is initialized. It is also worth noticing that only the non-zero elements of the filter responses need to be stored in practise.

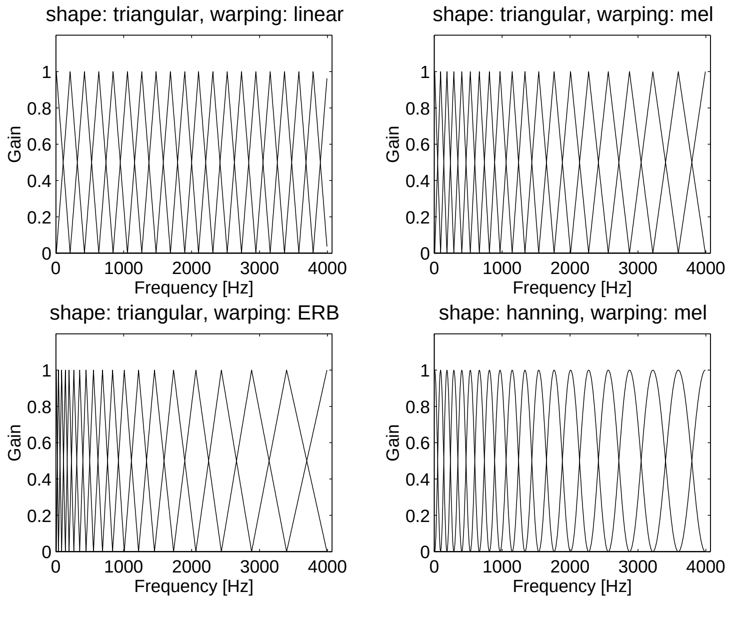

Figure 4.10: Examples of filter banks with different filter shapes and frequency warpings.

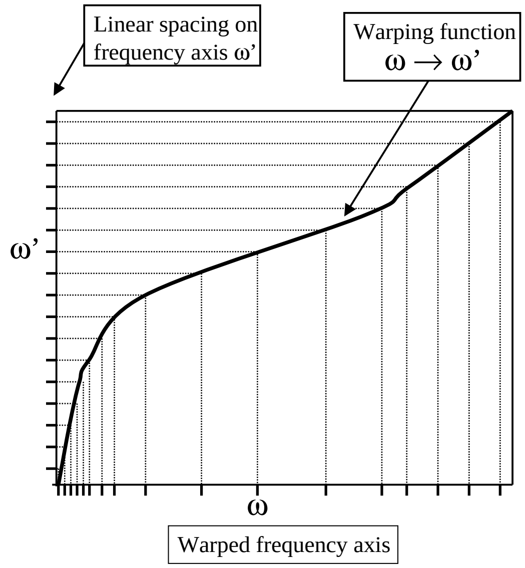

Figure 4.11: Principle of frequency warping.

A desirable property of a filterbank is that its magnitude response sums to unity at every frequency band, i.e.

$$\sum_{i=1}^{M} H_i[j] = 1 \tag{4.17}$$

for all j. This ensures that the whole frequency range of interest is processed with equal significance. Due to the finite precision and finite number of samples in the bank, (4.17) might be difficult to reach in practise. The filters in the low- and high-ends of the frequency range may cause problems also. For instance, the triangular filterbank in Fig. 4.8 has a response < 1 at both ends.

Center frequencies of the filters are often evenly spaced on some frequency axis. The axis can be linear as in Fig. 4.8, or *warped* according to some non-linear function as the mel-, Bark- or ERB-scales shown in Fig. 3.13. By frequency warping, one can adjust the amount of resolution that is desired around a certain frequency. The idea in wavelet analysis [151] is somewhat the same, but the terminology and formulations are different. There are two approaches to utilize frequency warping into spectral analysis [49]: *parametric* and *non-parametric* approaches. In the former case, the warping function is directly plugged into a parametric signal model such as *generalized linear prediction*. In the latter case, the warping function is sampled in finitely many points, and the sampled points represent the filter locations. This approach is used with the DFT-implemented filterbanks. Examples of warped filterbank magnitude responses are shown in Fig. 4.10 with different filter shapes and warping functions.

The idea of non-parametric frequency warping in the design of a warped filterbank is illustrated in Fig. 4.11. A desired number of filter center frequencies are placed linearly on the ω' axis (such as mel). Then, the inverse mapping $\omega' \mapsto \omega$ is used to resolve the center frequencies in the frequency axis ω (Hz). Thus, the warping function must be a bijective mapping so that it can be inverted uniquely. As a result of the warping, the axis ω is stretched/shrinked.

Chapter 5

FEATURE EXTRACTION

5.1 Filterbanks

Filterbanks have the advantage over the most of the other spectral representations that the features have a direct physical interpretation. This enables, for instance, utilizing *a priori* knowledge of the discrimination powers of subbands by appropriate subband weighting [145, 71, 115]. Also, if some of the subbands are contaminated by noise, the uncontaminated subbands can still be used [11, 105].

As in general data fusion [132, 73], we recognize from literature two main approaches to utilize filterbanks in speaker recognition: (1) *feature-level fusion* (*input fusion*) and (2) *classifier fusion* (*output fusion*). In the former case, subbands outputs are combined into a single M-dimensional feature vector and a single speaker model is trained. In the latter case, each subband output is considered independently and for each subband, a separate model is created.

5.1.1 Input Fusion

The simplest approach to subband-based feature extraction is to consider the subband outputs (possibly compressed using logarithm or other nonlinearity) directly as the features. The natural extension of this is to weight each subband output by the discrimination power of that subband. In [115], the subband weights were determined using *F-ratio* [161, 20] and a recognition performance index called *vector ranking*.

The author has presented an extension of the simple subband weighting [115] called *adaptive discriminative filterbank* (ADFB) [71]. The motivation for this processing is that a single discrimination value for all speech sounds might smooth out the effect of such subbands that are in general poor but

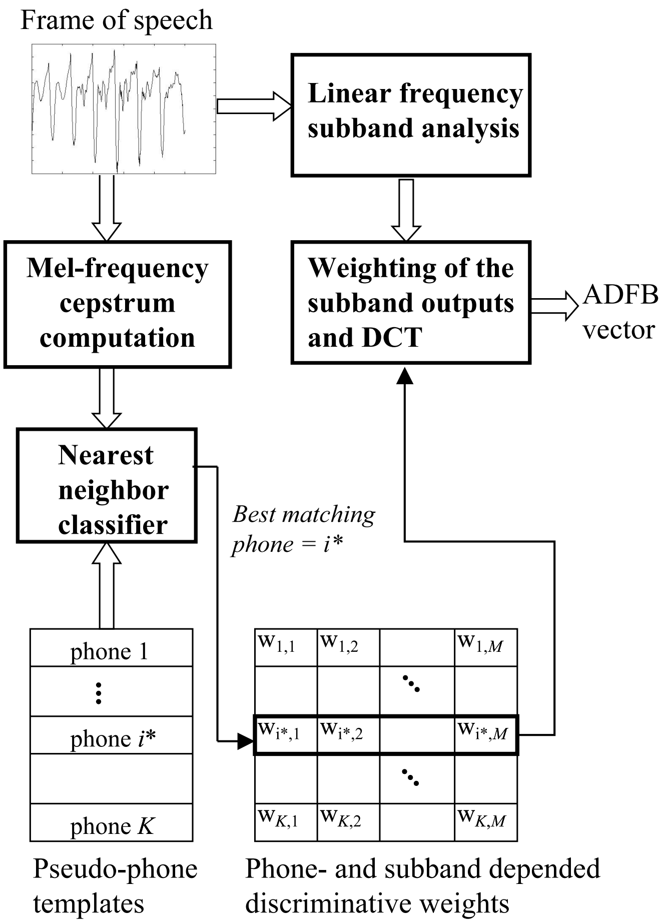
Frame of speech
Linear frequency subband analysis
Mel-frequency cepstrum computation
Weighting of the subband outputs and DCT
ADFB vector
Nearest neighbor classifier
Best matching phone = i*
phone 1
phone i*
phone K
$w_{1,1}$
$w_{1,2}$
$w_{1,M}$
$w_{i*,1}$
$w_{i*,2}$
$w_{i*,M}$
$w_{K,1}$
$w_{K,2}$
$w_{K,M}$
Pseudo-phone templates
Phone- and subband depended discriminative weights

Figure 5.1: Flowchart of the adaptive discriminative filter bank front-end [71].

discriminative for a certain phone. The flowchart of the method is shown in Fig. 5.1. The subband weights are adapted according to the coarse phone class ("pseudo-phone") of the input frame. In other words, there is a weight for each phone-subband pair that is determined in the training phase. The pseudo-phone templates are generated by clustering a large amount of speech data from a variety of speakers. The pseudo-phone templates are presented by the mel-frequency cepstral coefficients (MFCC) [131]. The subband processing itself is similar to MFCC computation, but the mel-frequency filterbank is replaced by a linear frequency filterbank, and the filterbank outputs are weighted by their discrimination values. As in MFCC, finally the filter outputs are decorrelated using discrete cosine transform (DCT).

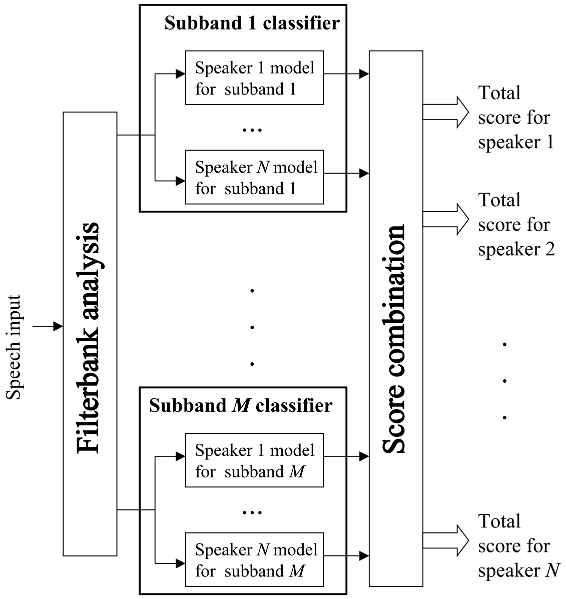
Speech input
Filterbank analysis
Subband 1 classifier
Speaker 1 model for subband 1
...
Speaker N model for subband 1
Subband M classifier
Speaker 1 model for subband M
...
Speaker N model for subband M
Score combination
Total score for speaker 1
Total score for speaker 2
Total score for speaker N

Figure 5.2: Structure of a subband-based classifier output fusion system.

5.1.2 Output Fusion

Another approach to combining evidences of several subbands is to model each subband independently and combine the scores of the subband classifiers [11, 10, 145, 24, 105]. The principle is illustrated in Fig. 5.2. Score combination on the classifier output level is flexible, since different classifier architectures can be used for different frequency bands.

The design of such systems includes designing of the front-end (filterbank), the individual classifier architectures, and the score combination method which is not a straightforward task. Some of the commonly used classifier output score combining rules have been summarized by Kittler & al. [77]. They found out that simple *sum rule*, i.e. combining the individual classifier outputs by summing them, gave the best recognition performance. They found out theoretically that the sum rule was most resilient to estimation errors.

The major drawbacks of the classifier output fusion systems are increased time- and memory requirements. For each speaker-subband pair, a separate model must be stored and in the recognition stage, each classifier must compute its own match score. The overall time increases with the number of subbands and the complexity of the subband classifiers.

The systems described in [145, 24] are text-dependent in which each subband is modeled by the Hidden Markov Model (HMM). The systems

[11, 10, 105] are text-independent. In [11, 10], each subband is modeled using a unimodal Gaussian distribution [14], and in [105] is the subband models are Gaussian Mixture Models (GMM).

Damper and Higgins combine directly the subband classifier outputs (log-probabilities) by the sum rule [77], whereas Sivakumaran & al. propose three different weighting schemes for the subbands [145]. In the first approach, the weight for the subband is determined by the empirical error rate of the subband, similar to [115]. In the second approach, the segmental signal-to-noise ratio (SNR) is estimated and a subband with higher SNR is associated with a larger weight. In the third approach, they define the weight using a competitive speaker model. The last approach is a well-known *score normalization* technique in speaker verification [92, 8]. The competing speaker model approach performed the best.

The method proposed by Ming & al. [105] attempts to select only those subbands that are less contaminated by noise. This is based on maximizing the *a posterior* probability of a given speaker model with respect to the uncontaminated subbands; the underlying assumption is that a noisy channel has a low probability, and will not be selected to the final scoring. What makes their algorithm attractive is that they did not make any *a priori* assumptions about neither the amount/type of noise, nor the number of contaminated subbands. Furthermore, the algorithm itself is very simple.

5.1.3 Frequency-Warped Filterbanks

Using psychoacoustically motivated warping functions (especially the mel- and Bark-scales) is common in speaker recognition. To the authors knowledge, these scales were first applied in the *speech* recognition task and later adopted to speaker recognition. However, it is likely that human ear is *not* optimally designed to recognize speakers, and for this reason, one should be cautious in using such transformations. In perceptually motivated signal representations, one implicitely assumes that the information ignored by the human peripheral auditory system is not important for the task at the hand, which may not be the case with automatic speaker recognition.

For the reason explained, it is worthwhile to study other than perceptually-motivated frequency warpings. Alternative frequency warpings have been proposed for speaker recognition in [49, 106, 107]. For example, the warping function proposed by Gravier & al. [49] is[1]

$$\omega' = \tan^{-1}\left|\frac{(1-\alpha^2)\sin\omega}{(1+\alpha^2)\cos\omega - 2\alpha}\right|, \tag{5.1}$$

where ω and ω' are the original and warped frequencies in radians, and $\alpha \in [-1, 1]$ is a control parameter. Positive values of α provide better resolution at low frequency region whereas negative values give emphasis to high frequencies. The case $\alpha = 0$ corresponds to linear frequency scale, i.e. no warping.

Miyajima & al. [106, 107] have proposed a more general approach to frequency warping, in which the warping function is specified by two parameters α and θ:

$$\begin{aligned}\omega' = \omega \quad &+ \quad \tan^{-1}\left(\frac{\alpha\sin(\omega-\theta)}{1-\alpha\cos(\omega-\theta)}\right) \\ &+ \quad \tan^{-1}\left(\frac{\alpha\sin(\omega+\theta)}{1-\alpha\cos(\omega+\theta)}\right).\end{aligned} \tag{5.2}$$

The parameter θ specifies the frequency around which more resolution is desired, and α specifies the amount of resolution at that frequency. Examples of these warping functions are shown in Fig. 5.3 for the sampling frequency $F_s = 8$ kHz.

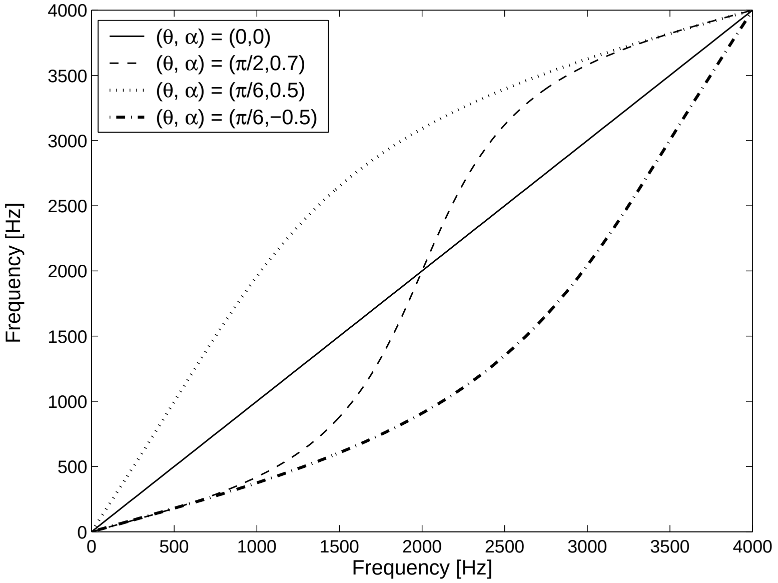

Figure 5.3: Examples of warping function (5.2) with diffent parameters.

[1] The formula is probably incorrect. The author did not succeed to reproduce the figures presented in the original paper using formula (5.1).

According to Miyajima & al. [106, 107], linear-, mel- and Bark-scales can be obtained as special cases of their warping function by selecting $\theta = 0$ and $\alpha = 0, 0.42, 0.55$ for the three cases, respectively. They optimized the parameters of the warping function and GMM-based speaker model jointly by using an iterative gradient search algorithm.

5.2 Linear Prediction

5.2.1 Time-Domain Interpretation

The rationale in *linear prediction* (LP) analysis is that adjacent samples of the speech waveform are highly correlated and thus, the signal behaviour can be predicted to certain extent based on the past samples. The LP model assumes that each sample can be approximated by a linear combination of a few past samples [58, 131]:

$$s[n] \approx \sum_{k=1}^{p} a[k] s[n-k], \tag{5.3}$$

where p is the *order* of the predictor. The goal of the LP analysis is to determine the *predictor coefficients* $\{a[k] \mid k = 1, \ldots, p\}$ so that the average prediction error (or *residual*) is as small as possible. The prediction error for nth sample is given by the difference between the actual sample and its predicted value:

$$e[n] = s[n] - \sum_{k=1}^{p} a[k] s[n-k]. \tag{5.4}$$

Equivalently,

$$s[n] = \sum_{k=1}^{p} a[k] s[n-k] \; + e[n]. \tag{5.5}$$

When the prediction residual $e[n]$ is small, predictor (5.3) approximates $s[n]$ well. The total squared prediction error is given by

$$\begin{aligned} E &= \sum_{n} e[n]^2 \\ &= \sum_{n} \Big(s[n] - \sum_{k=1}^{p} a[k] s[n-k] \Big)^2. \end{aligned} \tag{5.6}$$

To find the minimum value, partial derivatives of E with respect to the model parameters $\{a[k]\}$ are set to zero:

$$\frac{\partial E}{\partial a[k]} = 0, k = 1, \ldots, p. \tag{5.7}$$

By writing out the expressions (5.7) for $k = 1, \ldots, p$, the problem of finding the optimal predictor coefficients reduces in solving of so-called *(Yule-Walker) AR equations* [58, 62, 97]. Depending on the choice of the error minimization interval in (5.6), there are two methods for solving the AR equations: *covariance method* and *autocorrelation method* [131]. According to [58], for unvoiced speech, the two methods do not have large difference, but for voiced speech, the covariance method can be more accurate. However, according to [131, 67], the autocorrelation method is the preferred method since it is computationally more efficient and guarantees always a stable filter.

The AR equations for the autocorrelation method are of the following form:

$$\boldsymbol{R}\boldsymbol{a} = \boldsymbol{r}, \tag{5.8}$$

where $\boldsymbol{R}$ is a special type of matrix called *Toeplitz matrix*, $\boldsymbol{a}$ is the vector of the LPC coefficients and $\boldsymbol{r}$ is the autocorrelation (see [97, 131] for more details). Both the matrix $\boldsymbol{R}$ and vector $\boldsymbol{r}$ are completely defined by p autocorrelation samples. The *autocorrelation sequence* of $s[n]$ is defined as [67]:

$$R[k] = \sum_{n=0}^{N-1-k} s[n]s[n-k]. \tag{5.9}$$

Due to the redundancy in the AR equations, there exists an efficient algorithm for finding the solution, known as *Levinson-Durbin recursion* [97, 131, 62]. The Levinson-Durbin procedure takes the autocorrelation sequence as its input, and produces the coefficients $a[k], k = 1, \ldots, p$. The time complexity of the procedure is $O(p^2)$ as opposed to standard Gaussian elimination method [159] whose complexity is $O(p^3)$. The steps in computing the predictor coefficient using the autocorrelation method are summarized in Fig. 5.4.

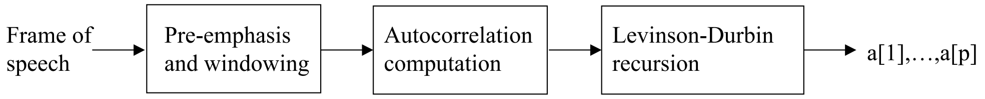
Frame of speech
Pre-emphasis and windowing
Autocorrelation computation
Levinson-Durbin recursion
a[1],…,a[p]

Figure 5.4: LPC coefficient computation using the autocorrelation method.

The Levinson-Durbin procedure produces predictors of order $1, 2, \ldots p-1$ as its side-product. Another side-product of the procedure are intermediate variables called *reflection coefficients* $k[i], i = 1, \ldots, p$, which are bounded by $|k[i]| \leq 1$. These are interpreted as the reflection coefficients between the tubes in the lossless tube model of the vocal tract [67].

Makhoul [97] has shown that if the original spectrum has a wide dynamic range, the LP model becomes numerically instable. This justifies the use of pre-emphasis filter prior to LP analysis: the spectrum of the signal is whitened and the dynamic range is reduced. An adaptive formula for the pre-emphasis can be used with LPC analysis [67, 97]:

$$\alpha = \frac{R[1]}{R[0]}, \tag{5.10}$$

where $R[i]$ is the autocorrelation sequence as defined in (5.9). The criterion (5.10) represents a simple *voicing degree detector* [67], that emphasizes more the voiced segments. An example is shown in Fig. 5.5.

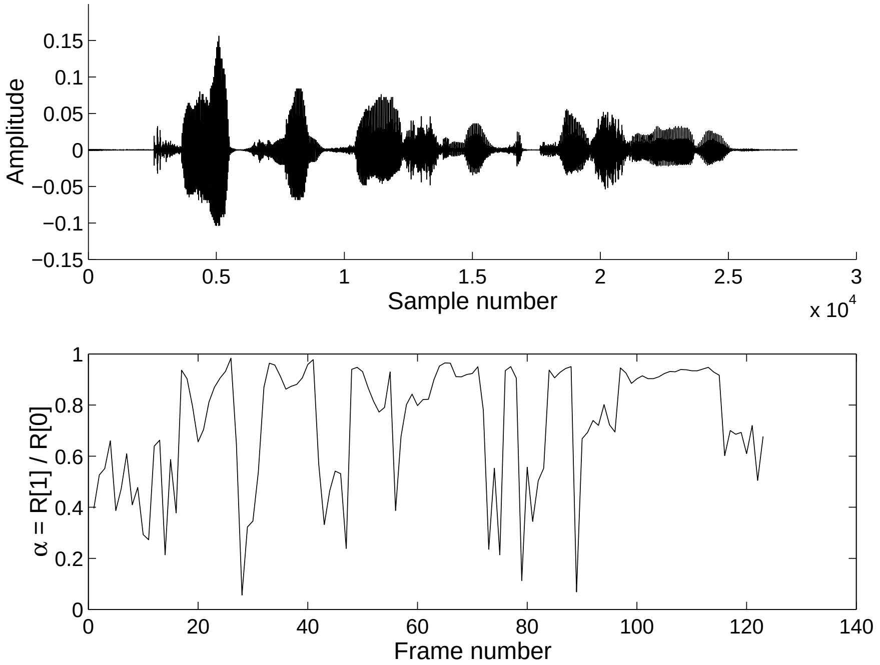

Figure 5.5: Example of voicing degree detector (5.10) using a 30 ms Hamming window.

Any signal can be approximated with the LP model with an arbitrary small prediction error [97]. The optimal model order depends on what kind of information one wants to extract from the spectrum. More insight into this can be seen by considering the frequency-domain interpretation of the LP. Makhoul [97] has proved that the minimization of (5.6) in equivalent to minimizing the square error between the signal magnitude spectrum and the model magnitude response. In other words, the LP model transfer function is a least square approximation of the original magnitude spectrum.

5.2.2 Frequency-Domain Interpretation

Equation (5.3) can be turned into equality as follows [131]:

$$s[n] = \sum_{k=1}^{p} a[k]s[n-k] \; + Gu[n], \tag{5.11}$$

where $u[n]$ represents excitation sequence and G is its gain. Considering this as the recurrence equation of an IIR filter, the sum term presents the feedback part, and $Gu[n]$ represents the input signal. In other words, the resulting signal $s[n]$ is a result of convolving the scaled excitation signal with the filter kernel.

The transfer function of the filter is given by:

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{k=1}^{p} \; a[k]z^{-k}}. \tag{5.12}$$

This filter has no zeros, and therefore, it is called an *all-pole filter* [97]. Since a pole represents a local peak in the magnitude spectrum, the model is restricted in the sense that it models only the peaks of the spectrum (resonances of the vocal tract). Nasal sounds and nasalized vowels include, in addition to the resonances, so-called *antiresonances* that result from the closed side-tube formed by the oral cavity [51]. Modeling of these antiresonances requires zeros in the filter [67, 100]. However, as the order p of the all-pole model is sufficiently high, the nasal resonances are also modeled in arbitrary accuracy [97]. This means that for a given error value, nasal sounds and nasalized vowels require higher order predictor than non-nasal speech sounds.

If we compare Eqs. (5.5) and (5.11), we can see that when the actual system that generated $s[n]$ is close to model (5.11), $e[n] \approx Gu[n]$. In other words, the residual signal $e[n]$ can be used in estimating the excitation signal. The poles of the transfer function (5.12), on the other hand, model the envelope of the short-term spectrum. The poles of $H(z)$ are expected to be located at the formant frequencies when the all-pole assumption is valid. This suggests that the smoothed spectral envelope obtained via LP analysis can be used in formant estimation [58, 131].

An example of using LP analysis in the spectral envelope extraction is shown in Fig. 5.6. The upper panel shows the original FFT spectrum, and the lower panel shows three different order LPC envelopes. It can be seen that $p = 6$ undersmooths the original spectrum. On the other hand, if the predictor order is high ($p = 100$), the LP model starts to fit the individual harmonics. A compromise ($p = 15$) gives the information about the spectral structure generated by the vocal tract filter. The formants are visible in the smoothed spectrum, showing approximate locations $F1$=250 Hz, $F2$=2000 Hz and $F3$=2500 Hz for the first three formants.

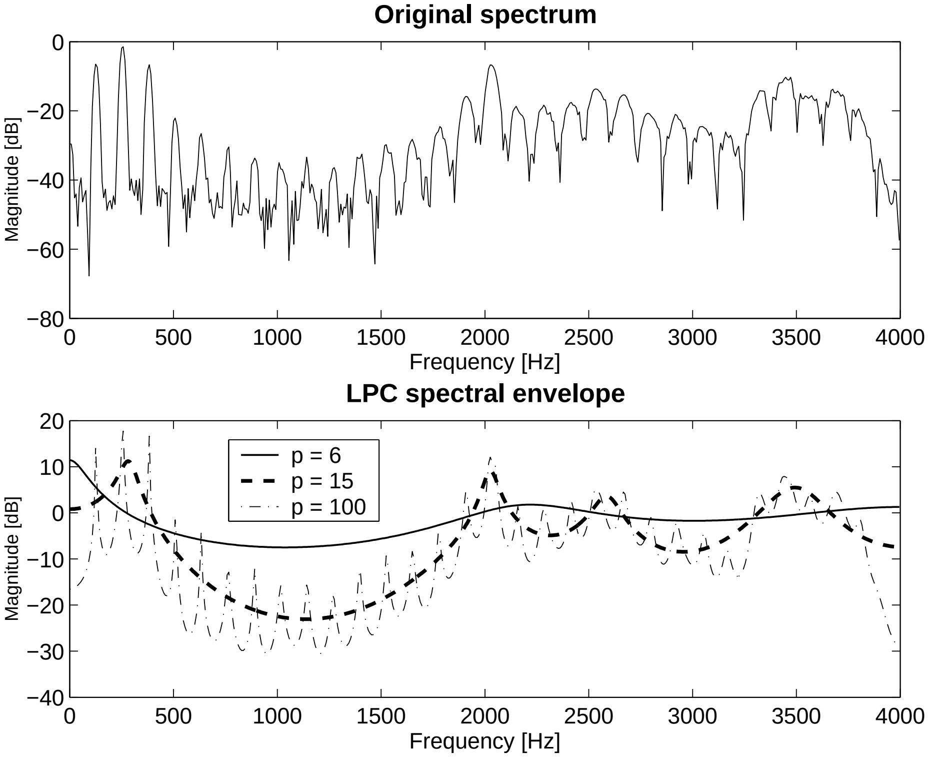

Figure 5.6: Estimation of the spectral envelope of vowel [i] by LP analysis using different order predictors ($p = 6, 15, 100$).

A thumb rule for the order selection is to select one complex pole per each kilohertz plus 2-4 poles to model the lip radiation and glottal effects [58]. For instance, for telephone speech the effective frequency range is 0-4 kHz. We therefore need approximately 4 poles + 2-4 poles = 6-8 poles. Since the complex poles must be real or occur in complex conjugate pairs to ensure that the filter coefficients are real, the model order is twice the number of poles. Thus, we would choose the order from $p = 12$ to $p = 16$. However, again we must remember that this rule is designed for *speech* recognition purposes, and we might have a different rule for speaker recognition.

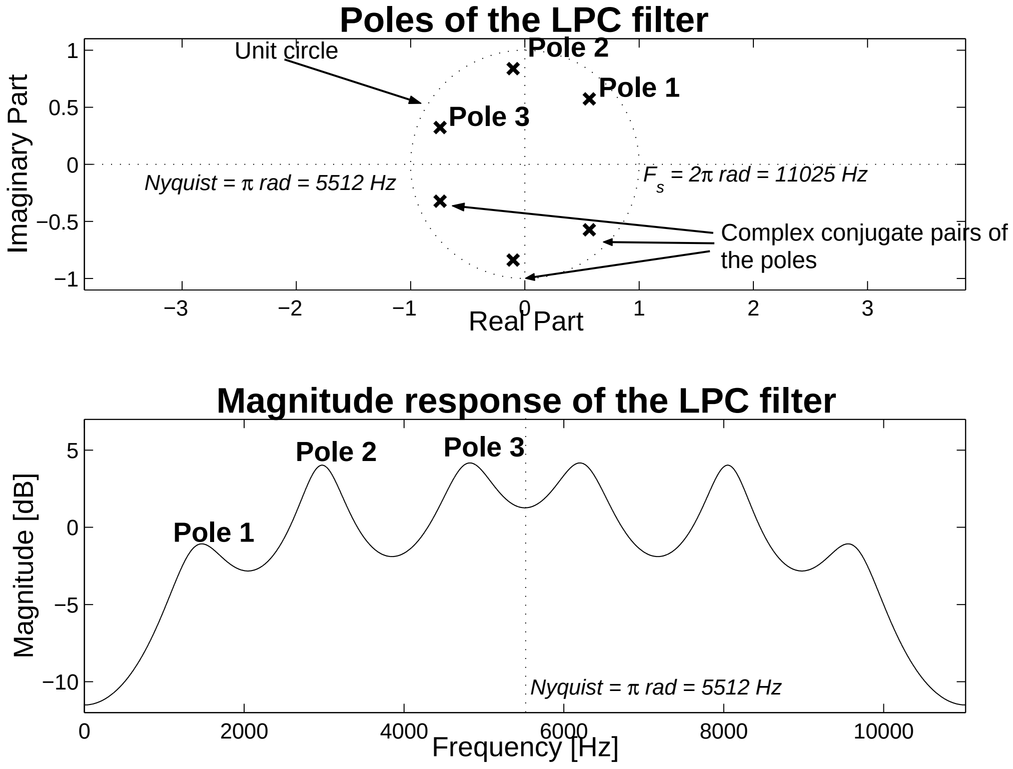

Figure 5.7: Example of the LPC poles on the complex plane and the corresponding magnitude spectrum (LPC order $p = 6$).

5.2.3 Representations Computed from LPC

Several alternative representations can be derived from the LPC coefficients. If the autocorrelation method is used, the Levinson-Durbin algorithm produces the reflection coefficients $\{k[i]\}, i = 1, \ldots, p$ as its side-product. They are also called *partial correlation coefficients*. When the vocal tract is modeled with the lossless tube model, at each tube junction, part of the wave is transmitted at the remainder is reflected back [20]. The reflection coefficients are the percantage of the reflection at these discontinuities.

Assuming the lossless tube model, ratio of the areas of the adjacent tubes is given by [58]:

$$\frac{A_{i+1}}{A_i} = \frac{1 - k[i]}{1 + k[i]}. \tag{5.13}$$

A new parameter set is obtained by taking the logarithm of the area ratio, yielding *log area ratios* (LAR). Since the LARs are derived from the LP coefficients, they are subject to the assumptions made in LP. To avoid singularity at $|k[i]| = 1$, an alternative for log area ratios are *arcsin reflection coefficients* [20], simply computed as taking inverse sine of the reflection coefficients.

LPC analysis can be used in formant estimation also [67, 58]. Given the transfer function (5.12), the roots of the denominator (i.e. the poles) can be found by any numerical root-finding method. Let the poles be $z_1, z_2, \ldots, z_p$. Each pole corresponds to a local peak in the spectrum, and therefore the poles are assumed to be correlated with the formant structure. Estimates for the formant frequencies and bandwidths are given by [67]:

$$\begin{aligned} \hat{F}_i &= \frac{F_s}{2\pi} \tan^{-1}\left(\frac{\text{Im } z_i}{\text{Re } z_i}\right) && (5.14) \\ \hat{B}_i &= -\frac{F_s}{\pi} \ln |z_i|. && (5.15) \end{aligned}$$

In practise, the prediction coefficients are highly correlated [141] and they have poor quantization and interpolation properties [44]. This has motivated to develop feature sets that are less correlated and more robust against quantization. An equivalent presentation to LPCs are so-called *line spectral frequencies (or pairs)* (LSF,LSP) [67, 58, 44]. The line spectral frequencies are formed as follows [67]. The LPC *inverse filter* $A(z) = 1 - \sum_{k=1}^{p} a[k] z^{-k}$ is decomposed into two $(p+1)$-order polynomials $P_{1,2}(z) = A(z) \pm z^{-(p+1)} A(z^{-1})$ so that $A(z) = \frac{1}{2}[P_1(z) + P_2(z)]$. The roots of the polynomials $P_{1,2}(z)$ can be shown to lie on the unit circle. Furthermore, they are interlaced with each other. Since the roots lie on the unit circle, they can be specified by a one parameter, the phase angle (argument) ω_i. These angles are the line spectral frequencies. The line spectral frequencies are ordered as follows [44]: $0 < \omega_1 < \omega_2 < \ldots < \omega_p < \pi$. An important property of LSF's compared to LPC coefficients is that in quantization, only the frequencies around the quantized coefficient are affected. Therefore, LSF's might be well-suited for vector quantization based speaker recognition.

LSF's are commonly used in speech coding, but they have been applied with good results to speaker recognition also [93, 91, 20, 169, 109]. Liu & al. [93] experimented mean and difference of adjacent LSF frequencies as new features in VQ-based speaker recognition. These new feature sets were abbreviated as MALS and DALS, respectively. According to Liu & al., these correlate with formant frequencies and bandwidths, respectively. The DALS feature set performed the best on their data set[2], and the results using LSF-based features were better than results obtained using the linear predictive cepstral coefficients (LPCC) [6].

[2] Consisting unfortunately only of 20 speakers.

Perceptual linear prediction (PLP) [55] is a form of generalized linear prediction that exploits some of the psychoacoustics principles, including critical band analysis (Bark), equal loudness pre-emphasis, and the intensity-loudness relationship. PLP and its variants have been used succesfully in speaker recognition [164, 113, 133, 156, 50]. For instance, in [156] it was observed that the PLP feature outperformed LPC coefficients in all conditions. In general, it seems that conventional features like MFCC can outperform PLP in clean environment, but PLP gives better results in noisy and mismatched conditions.

5.3 Cepstral Analysis

Linear prediction uses all-pole modeling of the spectrum. An alternative method to LPC is the so-called *cepstral analysis* [67]. In cepstral analysis, the magnitude spectrum is represented as a combination of cosine basis functions with varying frequencies. The cepstral coefficients are the magnitudes of the basis functions. Figure 5.8 shows a comparison of the spectral envelope estimates using the LPC (all-pole model) and the cepstrum representation. Notice that the peaks in the LPC model sharp, whereas the cepstrum presents a smoother envelope. In this sense, the LPC model preserves more details about the spectrum with the same number of coefficients.

Formally, the *real cepstrum* of digital signal $s[n]$ is defined as the inverse Fourier transform of the logarithm of the magnitude spectrum [58]:

$$c[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} C_s(\omega) e^{j\omega n} d\omega, \tag{5.16}$$

where we have denoted the logarithm of the magnitude spectrum by $C_s(\omega) = \log |S(e^{j\omega})|$.

It can be shown [67] that the coefficients $c[n]$ are the Fourier series coefficients of the log-spectrum and that the Fourier series presentation reduces to cosine series. In other words, log spectrum is represented as an infinite summation of cosines of different frequencies, and the cepstral coefficients are the magnitudes of the basis functions. The lower cepstral coefficients represent the slow changes of the spectrum and the higher coefficients the rapidly varying components of the spectrum. In voiced speech sounds, there is a periodic component in the magnitude spectrum, the harmonic fine structure that results from the vocal fold vibration. The slow variations are resulting from the filtering effect of the vocal tract, and the spectral tilt of the voice source. An example of spectrum modeling using real cepstrum is shown in

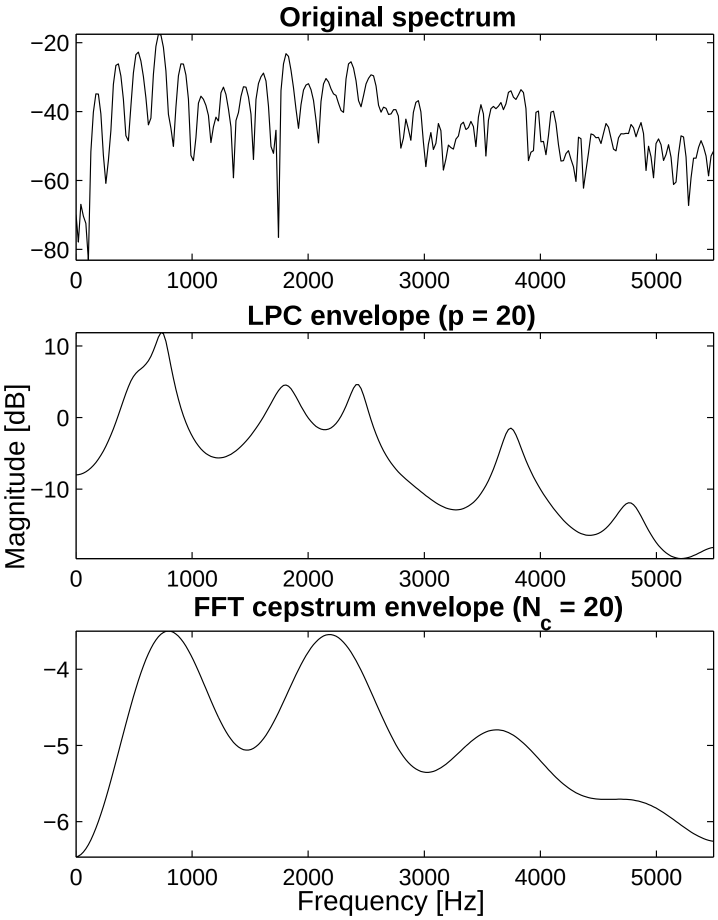

Figure 5.8: Example of LPC- and FFT-cepstrum based spectral envelope estimates.

Fig. 5.9. The log spectrum is reconstructed by setting the cepstral coefficients after N_c to zero and taking the Fourier transform of this sequence. Similar to LPC analysis, increasing the number of coefficients results in more details.

The reason for taking logarithm of the spectrum can be explained as follows [67]. According to source-filter theory, $|S(e^{j\omega})| = |U(e^{j\omega})||H(e^{j\omega})|$, where S, U and H correspond to the speech signal, source and filter, respectively. By taking the logarithm, the multiplicative components are converted into additive components: $\log\ |S(e^{j\omega})| = \log|U(e^{j\omega})| + \log|H(e^{j\omega})|$. Taking the logarithm corresponds to performing a *homomorphic transformation* [114]: multiplicative sequences are converted into a new domain, where they are additive. Therefore, the log-spectrum can be thought as a composition

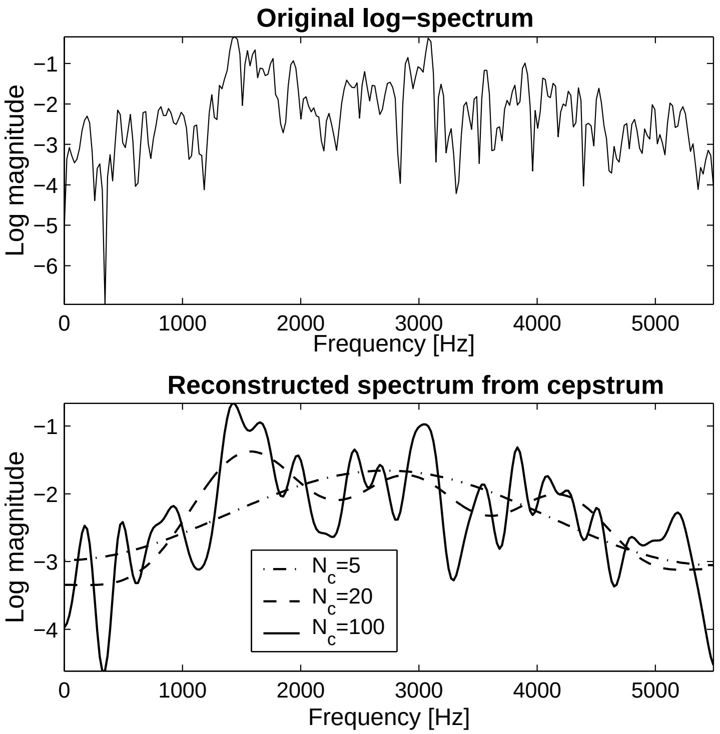

Figure 5.9: Example of spectrum reconstruction from cepstrum using different number of coefficients ($N_c = 5, 20, 100$).

of additive components that have different base frequencies.

The practical formula for computing the real cepstrum is obtained by using DFT and IDFT:

$$c[n] = \mathcal{F}^{-1}\{\log|\mathcal{F}\{frame\}|\}, \tag{5.17}$$

where $frame$ is the windowed analysis frame. In other words, the real cepstrum is obtained by applying inverse DFT to the logarithm of the magnitude of the DFT.

One should remember that the assumption of the harmonic component (fast variations) are present only in voiced sounds. However, cepstrum is used with both unvoiced and voiced segments since it has shown to be work in practise.

5.3.1 LPC-Cepstrum

The LPC coefficients are seldom used as features themselves [67]. It is observed in practice that adjacent predictor coefficients are highly correlated [141], and therefore, representations with less correlated features would be more efficient. A popular feature set is *linear predictive cepstral coefficients*. Given the LP coefficients $\{a[k]\}_{k=1}^{p}$, cepstral coefficients $c[n]$ are computed using the following recursive formula [58]:

$$c[n] = \begin{cases} a[n] + \sum_{k=1}^{n-1} \frac{k}{n} c[k] a[n-k], & 1 \leq n \leq p \\ \\ \sum_{k=n-p}^{n-1} \frac{k}{n} c[k] a[n-k], & n > p. \end{cases} \tag{5.18}$$

The relationship (5.18) was originally derived by Atal [6] as a new parameter set for speaker recognition. Since then, the LPC cepstrum has been used succesfully in both speech and speaker recognition. A noticeable thing is that although there are finite number (p) of LP coefficients, the LPC cepstrum sequence $c[n]$ is infinite. However, the magnitudes $|c[n]| \rightarrow 0$ fast with n, and thus a relatively small number of coefficients is needed to model the spectrum.

There are two worth noting things that the author wants to emphasize. First, it is important to notice that the LPC cepstral coefficients are derived from the predictor coefficients, and thus they are subject to the all-pole assumption of the LPC model. Therefore, in general the LPC cepstral coefficients are *not* the same as the cepstral coefficients derived from the magnitude spectrum directly. Secondly, in literature, the formula (5.18) is often given without reference to the used LPC model. Sometimes by convention there is a minus sign in front of the LP equation (5.3). This changes the LPC cepstrum equations to the following form[3]:

$$c[n] = \begin{cases} a[n] - \sum_{k=1}^{n-1} \frac{k}{n} c[k] a[n-k], & 1 \leq n \leq p \\ \\ -\sum_{k=n-p}^{n-1} \frac{k}{n} c[k] a[n-k], & n > p. \end{cases} \tag{5.19}$$

Atal [6] has compared the performance of the LPCC parameters with the following parameters for speaker recognition: LPC coefficients, impulse response of the filter specified by the LPC coefficients, autocorrelation function, and area function. From these features, the LPC cepstral coefficients performed the best. Unfortunately, Atal's data consists only of 10 speakers.

[3] The author spent several frustrating weeks in trying to find a bug from the LPC cepstrum computation, until it turned out that the used software (Matlab) assumed a minus sign in the front of the predictor (5.3).

5.3.2 Mel-Cepstrum

Maybe the most commonly used feature in speech recognition is *mel-cepstrum* [25]. Somewhat ironically, mel-cepstrum is one of the most commonly used parameters in speaker recognition. Computation of mel-cepstrum is similar as described in the previous Section. However, mel-warped (or any other) filterbank is applied in the frequency domain before the logarithm and inverse DFT. The purpose of the mel-bank is to simulate the critical band filters of the hearing mechanism. The filters are evenly spaced on the mel-scale, and usually they are triangular shaped [25, 67, 58]. The triangular filter outputs $Y(i), i = 1, \ldots, M$ are compressed using logarithm, and discrete cosine transform (DCT) is applied [58]:

$$c[n] = \sum_{i=1}^{M} \log Y(i) \cos\left[\frac{\pi n}{M}\Big(i - \frac{1}{2}\Big)\right]. \tag{5.20}$$

Notice that $c[0]$ presents the log magnitude, and therefore it depends on the intensity. Typically $c[0]$ is excluded for this reason. Important property of cepstral coefficients is that they are fairly uncorrelated with each other. This property has some important practical consequences. For instance, if a Mahalanobis distance is used as the metric in a classifier, there is little gain in using full covariance matrix in the Mahalanobis distance. Since the off-diagonals of the covariance matrix are close to zero, a diagonal covariance matrix is a good choice[4]. This leads to both savings in computation time and numerically more stable distance calculations.

5.4 Spectral Dynamics

In the previous sections we have assumed that each spectral parameter vector represents a "snapshot" of the continuously evolving spectrum at a certain time instant. Each spectral vector is assumed to be a representation of a short-term stationary signal; there is no time information encoded in these features.

While speaking, the articulators are continuously changing their positions with a certain rate. The articulatory movements are then reflected in the measured spectrum as, for instance, changes in formant frequencies and bandwidths. The rate of these spectral changes depends on the speaking

[4] Actually the Mahalanobis distance with diagonal covariance matrix is equal of first weighting the coefficients by their inverse variances and then computing Euclidean distance.

style, speaking rate and speech context. Some of these dynamic spectral parameters are clearly indicators of the speaker itself. Also, dynamic changes contain information of the message spoken (for instance, formant changes in diphthongs).

5.4.1 Delta- and Delta-Delta Features

A widely method to encode some of the dynamic information of spectral features is the following, known as *delta-features* [58, 131]. The time derivatives of the features are estimated by some method, and then the estimate of the derivative is appended to the feature vectors, yielding a higher-dimensional feature vector. As an example, if 12 mel-frequency cepstral coefficients are appended with their time derivative estimates, the dimensionality of the new feature vectors is $12 + 12 = 24$.

It might be argued that the feature space formed by concatenating static and dynamic features does not have an interpretation [73]. Also, since the dimensionality of the space is increased, more training data is required for reliable model estimates. Instead of concatenation in the feature level, Soong and Rosenberg [149] combined the static and dynamic feature classifier output scores, in their case VQ distortions.

Often, the time derivatives of the delta features are estimated also, yielding so-called *delta-delta parameters*. These are again appended to the delta-appended feature vectors, resulting in a higher dimensional feature space.

Sometimes delta-features and delta-delta-features are termed as *velocity*- and *acceleration*-coefficients, respectively [103]. This is due to the physical phenomena, i.e. velocity is the time derivative of displacement, and acceleration is the time derivative of velocity. Since the mass of the vocal tract does not change, according to the Newton's second law, the acceleration is proportional to the force and therefore the delta-delta parameters might be interpreted as descriptors of the force applied for moving the vocal tract. Although this kind of analogies can be made to give a "physical interpretation" for the dynamic parameters, they are only loosely connected with the real phenomena and caution should be made in doing inferences of the physical properties of the speaker's vocal tract. Naturally the selection of the original static features affect this "interpretation" strongly.

Delta- and delta-delta-parameters are used with several forms of parameters, especially the cepstrum and its variants [3, 42, 60, 149]. We do not fix the static parameter set here; as long as the parameter set is a short-term spectrum descriptor, the dynamic parameters represent spectral changes over time. An example of delta- and delta-delta features is shown in Fig. 5.10. The uppermost panel shows the time trajectory of a single mel-cepstral co-

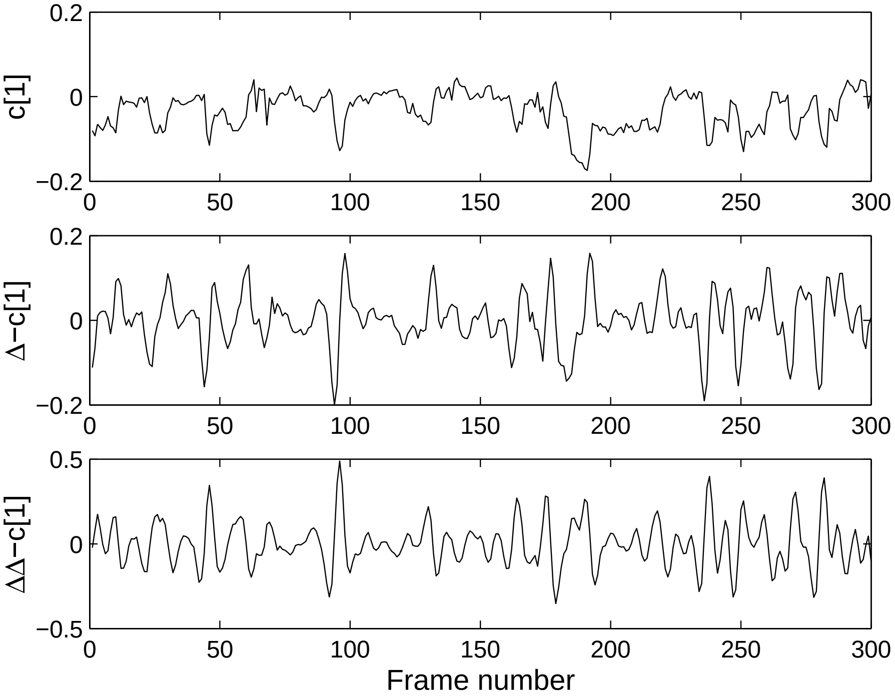

Figure 5.10: Time trajectories of mel-cepstral coefficient $c[1]$ and its Δ- and $\Delta\Delta$-trajectories computed using linear regression with M=2.

efficient $c[1]$ over time. The middle and bottom panels show the estimate of the first and second derivative.

There are two general principles to estimate the derivatives [67, 42, 60, 149]: (1) differentiating, and (2) fitting a polynomial expansion. Let $f_k[i]$ denote the ith feature in the kth time frame. In *differentiating*, the delta-parameter of the ith feature is defined as

$$\Delta f_k[i] = f_{k+M}[i] - f_{k-M}[i], \tag{5.21}$$

where M is typically 2-3 frames. The differentiation is done separately for each feature i, resulting in a delta feature vector.

The differentiating method is simple, but since it acts as a high-pass filtering operation on the parameter domain, it tends to amplify noise [149, 42]. For this reason, fitting a polynomial curve to the time trajectory of the parameter may result in better estimates. In statistics, this fitting problem is called *regression analysis*. Fitting a polynomial curve over several samples represents derivative estimate of a low-pass filtered time trajectory.

Similarly as with the linear prediction discussed in Section 5.2, the order of the polynomial is first fixed. Then, the least squares solution for the polynomial coefficients is obtained. As an example, for *linear regression*, i.e. first-order polynomial, the least squares solution is easily shown to be of the following form [131]:

$$\Delta f_k[i] = \frac{\sum_{m=-M}^{M} m f_{k+m}[i]}{\sum_{m=-M}^{M} m^2}. \tag{5.22}$$

Notice that the denominator is constant and can be interpreted merely as a scaling factor that can be replaced by another constant. Higher-order polynomials can be used to obtain smoother estimates, but in practise the first order polynomial is adequate according to [42, 149].

Figure 5.11 shows a comparison between the differentiator and linear regression methods for the $c[1]$ trajectory of Fig. 5.10. It can be seen that increasing the number of frames (M) smoothes the estimates with both methods, and the regression method generates smoother estimates.

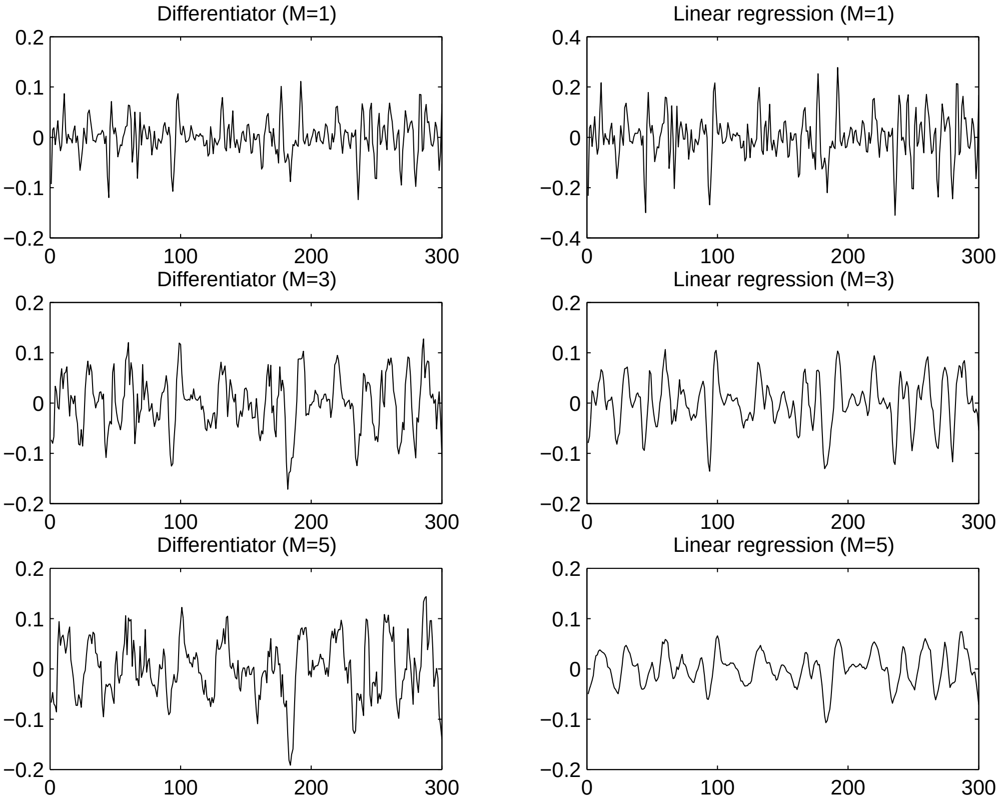

Figure 5.11: Comparison of derivative estimation using differentiator and linear regression with different orders.

In both methods, computation of $\Delta f_k[i]$ requires M frames from past and future. Therefore, the utterance endpoints need to be treated as special cases. Simple methods include padding of the M extra frames in both ends with zeroes, random numbers, or copies of the adjacent frames [60]. If the delta-deltas or higher order derivatives are estimated, the boundaries should be handled with more care since the error accumulates each time the deltas are computed from the previous deltas. Hume [60] compared a few padding methods, and the zero padding gave the best results.

For a given static feature, there is an optimum length for the regression window size M [3, 149, 103]. As seen already, too short a window leads to noisy estimates. If the window is too long, the temporal resolution decreases. Furthermore, it has been demonstrated with cepstrum that different coefficients should have a different regression window length, simply due to the fact that the dynamics of the coefficients are different [60]. Hume [60] found out that for the higher order cepstral coefficients, smaller regression windows should be used since these coefficients vary more rapidly. Motivated by this, Hume suggested the following simple algorithm. First, the regression window sizes of the first and last coefficients are specified. Then, the window lengths for the middle coefficients are linearly interpolated between these two values. The nearest odd window lengths to the interpolated values are used as window lengths.

5.4.2 Alternatives to Delta Processing

A few alternatives to dynamic feature extraction beyond the delta processing have been proposed. The *RASTA* processing [56] is one of the most well-known methods. RASTA is based on the model of human hearing mechanism. Human ear is more sensitive to certain modulation frequencies, and the RASTA processing attempts to filter out unimportant modulation frequencies. The importance of modulation frequencies for speaker recognition have been studied in [157]. RASTA and related methods have been used for speaker recognition in [133, 50, 113].

A somewhat different approach for speech dynamics estimation has been presented by Petry and Barone [120, 121]. The approach is based on nonlinear chaos-theoretic approach called *largest Lyapunov exponent*. In [121], improvement was obtained by adding the Lyapunov exponents along with the cepstrum and delta-cepstrum.

5.5 Alternative Spectral Representations

A large number of methods have been proposed for spectral feature extraction. One approach [48] attempts to use, instead of sines and cosines, a better set of basis functions called *Fourier-Bessel functions*. The Fourier-Bessel functions are decaying with time, and therefore they might be more better suited for the physics of sounds. However, the results obtained by Gopalan & al. [48] were not as promising as expected, and the selection of the final features was somewhat an *ad hoc* approach.

To the author's opinion, a very interesting work has been carried out by Jang & al. [66]. They use a *data-driven* approach to the feature extraction by finding for each speaker his/her personal basis functions by using *independent component analysis* (ICA) [61].

Artifical neural networks [53, 16] have also been used for feature extraction. For instance, Konig & al. [80] used a *multilayer perceptron* (MLP) in transforming a large-dimensional feature space (162 features) into low-dimensional representation (34 features). They had three hidden layers in their MLP. The middle hidden layer had a small number of neurons. After training the network, they removed the output layer and last hidden layer, and the remaining network was used to project the high-dimensional features into low-dimensional space.

Chapter 6

EXPERIMENTAL SETUP

6.1 Speech Material

Two speech corpora are evaluated in this thesis, a Finnish corpus collected by University of Helsinki [63], and a standard American English TIMIT corpus provided by Linguistic Data Consortium [90].

6.1.1 Corpus 1: Helsinki Corpus

The first corpus, denoted as *Helsinki corpus*, was collected by Päivikki Eskelinen-Rönkä at the Department of Phonetics of the University of Helsinki [33]. The author acknowledges the Department of Phonetics for providing this corpus for the experiments.

The Helsinki corpus consists of 110 native Finnish speakers from various dialect regions of Finland. There are 57 males and 53 females. All speakers were prompted to read the same material. The recordings were done in a silent environment with a professional reporter C-cassette recorder. The data was digitized using a sampling frequency of 44.1 kHz and a quantization resolution of 16 bits per sample. For our experiments, we downsampled and re-quantized the files in order to simulate better telephone line quality. The new sampling rate was set to $F_s = 11.025$ kHz, thus giving an effective bandwidth of about 5.5 kHz. Anti-aliasing FIR filtering was performed prior to downsampling. The files were stored in 8-bit μ-law compressed NeXT/SUN "au"-files. The duration of a file is 20 seconds per speaker. Each file was divided into disjoint training and evaluation files, both 10 seconds in duration.

6.1.2 Corpus 2: TIMIT Corpus

The second corpus is an American English *TIMIT corpus* [90]. The corpus consists of native American English speakers from 8 dialect regions. The corpus contains in total 630 speakers, of which 438 are males (70 %) and 192 females (30 %). There are 10 speech files for each speaker. Two of the files have the same linguistic content for all speakers, whereas the remaining 8 files are phonetically diverse. The corpus has been recorded in a soundproof environment with a high-quality microphone. Speech files are stored in NIST/Sphere "wav"-file format with a sampling frequency of 16 kHz and a quantization resolution of 16 bits per sample. As with the Helsinki corpus, the files were downsampled and re-quantized to $F_s = 11.025$ kHz at 8 bps. The files were stored in μ-law compressed NeXT/SUN "au"-files.

In order to speed up simulations and in order to be more comparable with the Helsinki corpus, we selected a smaller subset of the TIMIT. We decided to use the speakers from the dialect region DR7 (western dialect). The selection of the subset was arbitrary. This subset contains 100 speakers (74 males and 26 females). We notice that the proportion of male speakers is higher than in the case of the Helsinki corpus. Majority of speakers in TIMIT are males.

We selected 6 files for training and 4 files for evaluation for each speaker. The two phonetically identical files ("sa" and "sx" sentences) were included in the training set. The training and testing files were concatenated into a single file for each speaker. The average durations of the training and evaluation data are 19.1 and 11.6 seconds, respectively. The attributes of both corpora are summarized in Table 6.1.

Table 6.1: Summary of the evaluated corpora.

	Helsinki	TIMIT (subset DR7)
Language	Finnish	American English
Speakers	110 (57 M + 53 F)	100 (74 M + 26 F)
Speech type	Read speech	Read speech
Recording conditions	Clean	Clean
Sampling frequency	11.025 kHz	11.025 kHz
Resolution	8 bps (μ-law)	8 bps (μ-law)
Training speech	10 sec.	19.1 sec.
Evaluation speech	10 sec.	11.6 sec.

6.2 Measuring the Performance

In all experiments, the performance evaluation is carried out by dividing the extracted feature vectors into two disjoint sets, one for modeling (training set) and another for recognition (evaluation set).

6.2.1 Individual Features

The effectiveness of the individual features is measured by the Fisher's F-ratio defined as [161, 20]

$$F_i = \frac{\text{Variance of speaker means of feature } i}{\text{Average intraspeaker variance of feature } i}. \tag{6.1}$$

The F-ratio compares two variances, the variance of the feature *between* different speakers and the variance *within* speakers. A good feature has a large variance between speakers, but a small variance within a speaker (see Section 2.1). Thus, high F-ratios are desirable. However, the F-ratio does not take into account possible correlations between different features, and it has other limitations also [20]. In this work, we do not use the F-ratio to measure absolute performance of a given feature. We use it in comparing the effectiveness of *different features of the same feature set*, e.g. different MFCC coefficients. This information can be directly exploited in the classification phase by giving more weight to those features that are more discriminative.

6.2.2 Classification Experiments

The absolute performance of the different feature sets is measured by classification experiments. We use vector quantization based classification [148, 75]. From each speaker's training set, a fixed-sized codebook is generated using the *Randomized Local Search* (RLS) algorithm [40][1]. This iterative algorithm is similar to the famous GLA (K-means) algorithm [89], and it is very simple to implement. RLS is less sensitive to the initial solution (codebook), and it is guaranteed to give always better quality codebook than the GLA. Running times are longer than for GLA, but this is not a problem in practise since the speaker models are created off-line.

The speaker models are trained independently of each other, and therefore the recognition rates might not be as high as they would be if a discriminative training algorithm was used instead [54]. However, our main goal here is

[1] This method is documented also in `ftp://ftp.cs.joensuu.fi/pub/Reports/A-1999-5.ps.gz`

not to optimize the classifier, but to compare different feature sets. Thus, the selection of the training and classification algorithms are of secondary interest. The author is aware that these two are interrelated - a certain modeling technique might be better for a certain feature set but not good for some other. However, since the VQ is a non-parametric modeling approach, we make minimal assumptions about the underlying feature distribution. We believe that the results generalize to other modeling techniques such as the GMM modeling.

The evaluation sets are matched against the speaker models in the database. As the matching function, we use the average quantization distortion (2.1) with the Euclidean distance metric unless otherwise mentioned. We assume a closed speaker database, and therefore the speaker whose codebook yields the smallest distortion for the test sequence is selected as the identification decision. The type of the recognition task (closed-set identification, open-set identification, verification) is not considered here, since it only affects the type of the decision. In other words, if a certain feature set gives good performance in the closed-set identification task, it is expected to generalize to the other two tasks also. Several optimization tricks (e.g. score normalization [13, 8]) could be applied in order to get better *absolute* performance. We emphasize that we are not seeking for an optimal classification system, but instead, attempt to compare different feature sets.

The performance of the classification is measured by the identification error rate:

$$Error = \left(\frac{N_e}{N}\right) \times 100\%, \tag{6.2}$$

where N_e is the number of incorrectly classified test sequences, and N is the total number of sequences. In preliminary experiments, we classified full

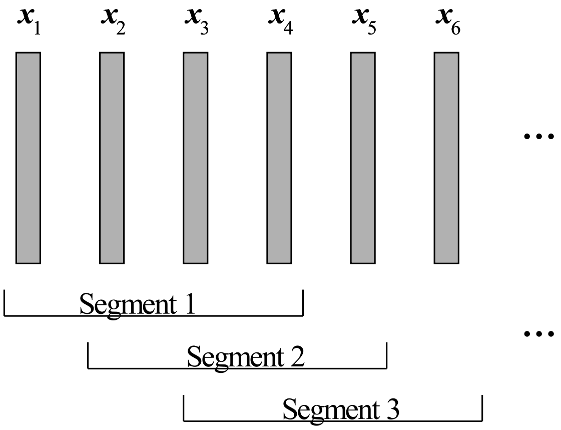

Figure 6.1: Blocking of the evaluation data in segments .

test sequences. However, it turned out quickly that the error rates were zero in many cases and therefore it was hard to observe any differences. For this reason, we decided to split the test sequences into smaller segments that are classified one by one. The process is illustrated in Fig. 6.1. This procedure enables better comparison between different features. Notice that this method is purely for improving the resolution of the test results. The recognition rates for full test sequences are slightly better.

6.3 Outline of the Experimental Setup

Since there is a large number of different feature sets to be evaluated, and each of these has many adjustable parameters, the evaluation of all possible parameter combinations is not possible. We apply simple *line search* strategy, that is, we vary one parameter at a time while keeping the rest of the parameters fixed. Although this procedure does not guarantee a globally optimal parameter combination, it gives an idea what are the most critical parameters that we should focus on.

6.3.1 Comparison of Different Feature Sets

First, we seek for the most promising feature sets from a large number of candidates. The preprocessing steps preceding feature extraction are fixed as follows. The DC offset is removed from the input signal by subtracting its mean value from it. Pre-emphasis is carried out using the differentiator specified in (4.14), where the filter coefficient is computed adaptively for each frame using the formula (5.10). Silence detection is not carried out. The window function is a 30 ms Hamming window, shifted forward by 20 ms.

After the general preprocessing step, several feature candidate sets are computed. We selected the most commonly used features in speaker recognition, that we classify into following categories:

1. Filterbanks

2. FFT-cepstral features

3. LPC-derived features

4. Delta features

We use mainly classification in optimizing the parameters of the different feature sets. We also apply the F-ratio analysis of individual features when we feel that it might bring some insight into understanding better the features.

For the *filterbank* feature set, we use FFT-based implementation with linearly spaced filter center frequencies on the interval $[0, 5512]$ Hz (see Fig. 6.2). The first and the last filters are lowpass and highpass type, respectively, whereas the middle filters are bandpass type. The passband of i:th filter begins from the center frequency of $(i-1)$:th filter and ends to the center frequency of $(i+1)$:th filter. The adjacent filters cross at the point where their magnitudes are $\frac{1}{2}$. This design ensures that the total response of the filterbank is close to 1 at all frequencies (requirement (4.17)). The filter shape and the number of filters affects how well the requirement is satisfied. We made a few experiments with some of the well-known window function [62] and decided to select the following: (1) rectangular window, (2) triangular window, and (3) Hanning window. For other window types like Gaussian and Hamming windows, requirement (4.17) was harder to satisfy. We believe that the shape of the filter is not an important parameter as long as we use the short-term DFT, whose spectral estimation errors and resolution are governed by the time-domain window shape and the frame length, respectively.

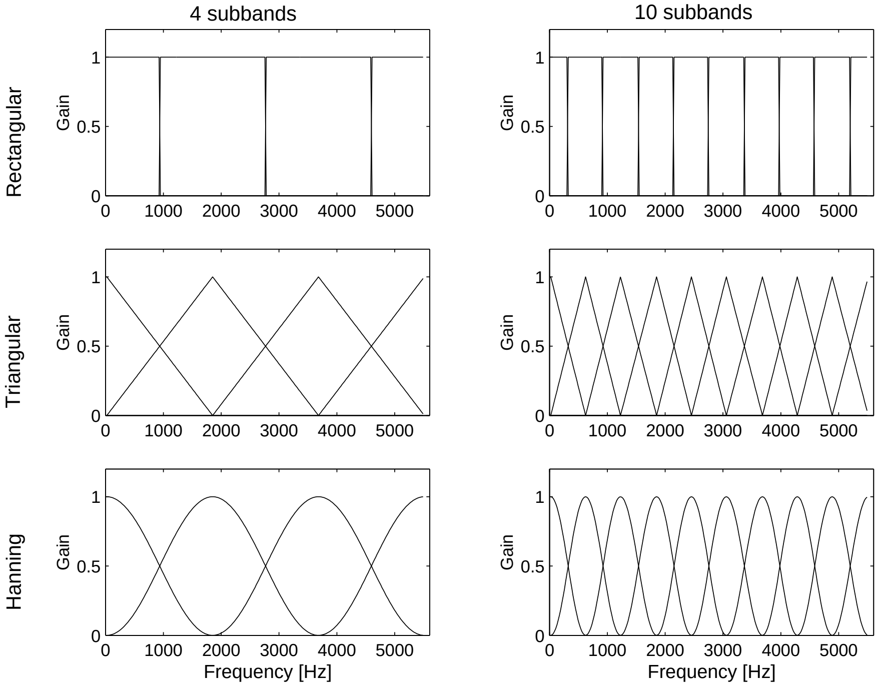

Figure 6.2: Examples of designed filterbanks.

In addition to using filter bank outputs directly as the features, we compress them by using logarithm and cube root. Logarithmic compression was selected since it is widely used in the mel-frequency cepstral coefficient calculation. In order to avoid numerical problems, we added constant 1 to the filter outputs before taking the logarithm. This was found experimentally to give better results than adding a small constant that is sometimes used. The cube root compression, on the other hand, exploits the intensity-loudness law of hearing [55] and it has reported to give better results in some cases. With all three functions, each filter output is further normalized by the sum of all filter outputs to ensure that the feature vector is independent of the input signal intensity.

By *FFT-cepstral features* we mean the cepstral parameters that are derived from the FFT-implemented filterbank analysis. The processing is very similar to subband processing described above. The difference is that the (compressed) filter outputs are further decorrelated using DCT. This is reasonable, since adjacent frequency bands are very likely to be highly correlated for most speech sounds.

In addition to linear frequency spacing, we apply mel-, Bark-, and ERB-warped filterbanks. In each case, the filter center frequencies are linearly spaced on this frequency scale, and the true filter center frequencies are computed by the corresponding one-to-one inverse mapping. For mel-, Bark- and ERB-scales, we used the formulae (3.4), (3.6) and (3.7), respectively.

We study the following *LPC-derived features*:

- LPC coefficients themselves (LPC)
- Linear predictive cepstral coefficients (LPCC)
- Line spectral frequencies (LSF)
- Reflection coefficients (REFL)
- Log area ratios (LAR)
- Arcus sine coefficients (ARCSIN)

All of these feature sets present somewhat the same information, but some of them might be more robust to quantization and better suited for the Euclidean distance classifier that we use. Two of the feature sets are of special interest: LPCC and LSF. The LPCC feature set is widely used in speaker recognition. The LSF feature set is not as popular, but it has good quantization properties, and therefore it might suit well for the VQ-based modeling.

We also study *formant estimates* computed from the LPC poles (Equations (5.14) and (5.15)). This formant estimation certainly produces spurious formants since the formant trajectories are not smoothed. However, from the viewpoint of automatic speaker recognition, it does not matter if a feature has no phonetic interpretation, if it gives good results.

The *delta features* depend on the selected static features and their parameters. Therefore, this feature set is evaluated after the individual static features have been compared and their parameters have been optimized. We limit ourselves to the delta parameters of those static features that give the most promising results. One might argue that a poor performance of a static feature does not necessary imply that its dynamics would not be discriminative. However, considering a practical viewpoint, a system that uses a static feature and its corresponding delta features is easy to implement as opposed to a system that uses delta parameters of some other parameter. Also, the computational complexity is smaller since only one static feature set needs to be computed.

6.4 Test Hypotheses

6.4.1 Performance of the Feature Sets

The subband-based representations (filterbanks) are expected to give good recognition rates and to give a good ground for the FFT-based cepstral features. We hypothesize that in general, a high number of subbands should be used. However, if we use too many subbands, the results are expected to get worse since the features do not anymore describe neither the spectral envelope, nor the harmonic structure. High-dimensional presentations also require a lot of training data (see Section 2.1).

FFT-based cepstral features are expected to give good results, since they are based on the subband presentation, and they have been reported consistently to give good results. Based on the author's previous experience, it is expected that a rather small number of cepstral coefficients (10-20) is enough to describe the speaker characteristics.

The mel-cepstrum [25] is probably the most commonly used feature in speaker recognition. However, we hypothesize that this might not be the best cepstrum representation. The mel-scale simulates human hearing, but there is no reason to assume that the human ear resolves frequency bands optimally in respect to the speaker recognition task. Furthermore, several studies have indicated that mid- and high-frequencies are more important for speaker recognition than low-frequencies (see e.g. [108, 145]). This is in contradiction with the definition of mel-cepstrum that emphasizes low frequencies. In general, we believe that any auditory-based spectral presentation should not be used as a black box, but instead, we should focus on understanding where in the spectrum we have relevant information for speaker discrimination.

Regarding the LPC-based representations, we expect slightly worse results than the subband- and FFT-cepstral features give. Throughout all experiments, we use the VQ-classifier with Euclidean distance measure, but this might not be the best distance measure. For instance, for the LPC features and LPC cepstrum, several alternative distance measures have been proposed (see [131]).

However, two of the LPC-derived feature sets are expected to give good results. These are the linear predictive cepstral coefficients (LPCC) and the line spectral frequencies (LSF). Both of these have been reported to give good results. There has been a great deal of debate whether the FFT- or the LPC-based cepstral coefficients are better features. It is often stated that the LPC-cepstrum is computationally more efficient, but the FFT-based cepstrum is more accurate. However, these two feature sets are often compared with parameters that are not comparable. For instance, if the conventional LPCC feature set is used, then the FFT-cepstrum should be based on linear frequency scale and not on the mel scale as it is often done.

Based on the author's experience and on literature, we expect that the dynamic parameters give slightly worse results than any of the static parameters. We hypothesize that the regression method gives a better representation than the differentiator method. The optimal regression (or differentiator) order depends both on the frame overlap, the number of frames used in the derivative estimation, as well as the original static feature set. If the order is high (long temporal span), rapid spectral changes that might be speaker-dependent, are smoothed out. On the other hand, if the order is low (short temporal span), it is likely that the feature trajectories become noisy.

6.4.2 Classifier Parameters

Based on the author's previous experience, we expect that VQ-based speaker modeling works well for most of the features. The Euclidean distance for subbands and cepstral features has a well-established theoretical background [131], and more importantly, it has been found to work in practise. We do not have previous experience with the LPC-derived features, and therefore we do not make any hypotheses about them.

In general, we expect that using larger codebooks improves recognition performance. For the minimum size, we expect a number between 16-64, depending on the feature set, the dimensionality, and the amount of training data.

Chapter 7

RESULTS AND DISCUSSION

7.1 Filterbanks

7.1.1 Subband Discrimination

First, individual subbands were investigated using the F-ratio measure. Filter outputs were compressed using three different functions: linear (no compression), logarithm and cube root. The 30 triangular filters were linearly spaced over the Nyquist range of the signal, which is 5.5 kHz. Therefore, each filter covers approximately a 5500/30 = 183 Hz frequency range. On average, the first two subbands contain the region of the first harmonic of $F0$.

The discrimination results of the individual frequency bands are shown in Figures 7.1 and 7.2 for the Helsinki and TIMIT corpora, respectively.

We observe from both figures that the two nonlinearities give, on average, slightly higher F-ratios for the subbands than the linear processing. The cube root gives highest values on average. The three methods follow the same trend over different frequencies. An exception occurs with the TIMIT corpus on the lowest frequency band where the cube root indicates good discrimination, while the other two methods indicate poor discrimination.

Both corpora show a “peaking” behaviour on the curves. We can see that the middle frequency range (about 2-3 kHz and 2.5-4 kHz for the Helsinki and TIMIT corpora, respectively) has higher discrimination compared to other frequencies. The lowest band (0-183 Hz) also seem to have good discrimination in both corpora when the cube root is applied. This might be due to the fact that on average, the first $F0$ harmonic falls in this region (notice that on both corpora majority of the speakers are male). The strength of the first harmonic correlates to some extent with the spectral tilt (and therefore, the voice quality) which is very likely speaker-specific. The author does not know if there is a systematic and large-scale study on the spectral tilt measures for speaker recognition.

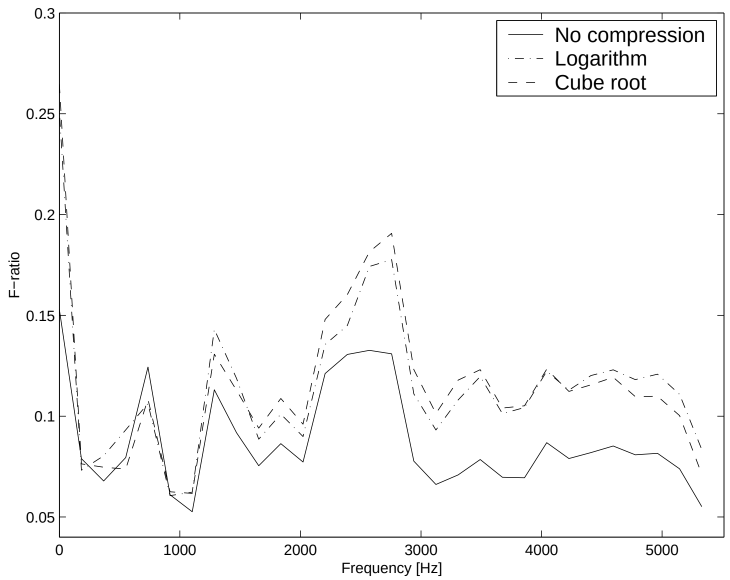

Figure 7.1: Discrimination of individual subbands of the Helsinki corpus.

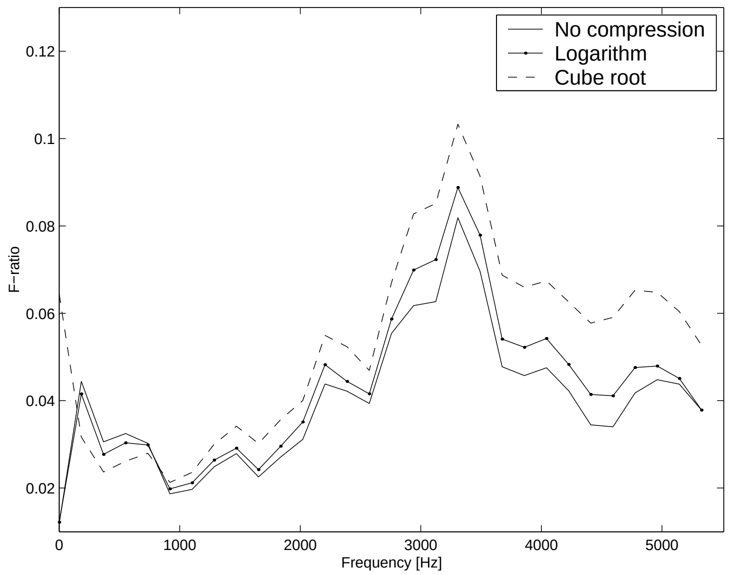

Figure 7.2: Discrimination of individual subbands of the TIMIT corpus.

The TIMIT corpus indicates unimportance of the frequency range 0.2-2 kHz, which is consistent with other studies [108, 115, 145]. The Helsinki corpus indicates similar behaviour, but not as clearly and uniformly as the TIMIT corpus. Since the analysis procedures were identical, the differences can be attributed to the differences in language and speaker population.

The observations suggest that, instead of using a critical band filter bank, a better one could be the one that places a lot of filters in the important middle- and high-frequency regions. In [115] the mel-cepstrum filterbank was replaced by a new filter bank where the emphasis was on middle frequencies. Orman & Arslan [115] reported identification error rate of 17 % using the mel-cepstrum, which was dropped down to 11 % with the tailored filter bank. Although this kind of filterbanks certainly need to be tailored separately for a given population and language, it shows the potential of a very simple and intuitive subband processing. The author has proposed an adaptive subband weighting method that takes into account the rough phoneme class of the input frame [71]. This method is a generalization of the simple static subband weighting.

7.1.2 Classification Results

Next, we conducted classification experiments using a segment size of 350 vectors for both corpora. This corresponds approximately to 7 seconds of speech with the current frame rate (a 30 ms frame, shifted forward by 20 ms).

First, we fixed the number of subbands to 30 and compared the nonlinearities. The classification error rates for codebook size $K = 64$ are listed in Tables 7.1 and 7.2. Several observations can be made from these results. First of all, subband processing works well, and is worth further studies. An error rate of 0 % can be reached with this very simple processing, without any weighting of the subbands.

The classification results agree with the F-ratios obtained in the previous experiment: the cube root gives systematically the lowest error rates while the linear processing gives the highest error rates. In general, the classification accuracy improves with increasing codebook size, as expected. Of the two data sets, the Helsinki corpus gives better results even though it has more speakers than the TIMIT subset. This signifies the importance of the language. At this stage, we are not going to make a detailed comparisons between the two languages.

Table 7.1: Subband classification error rates (%) on the Helsinki corpus using different dynamic range compressions.

Codebook	**Nonlinearity**		
size	None	Logarithm	Cube root
1	16.83	11.27	**7.35**
2	20.06	5.24	**4.74**
4	5.06	2.43	**1.22**
8	1.81	1.42	**0.00**
16	0.74	0.02	**0.00**
32	0.57	**0.00**	**0.00**
64	0.73	**0.00**	**0.00**
128	0.45	**0.00**	**0.00**

Table 7.2: Subband classification error rates (%) on the TIMIT corpus using different dynamic range compressions.

Codebook	**Nonlinearity**		
size	None	Logarithm	Cube root
1	52.83	52.46	**49.49**
2	48.16	42.63	**30.60**
4	36.32	27.65	**22.53**
8	17.32	11.55	**3.85**
16	8.46	5.44	**0.65**
32	6.25	4.84	**0.50**
64	5.84	4.84	**0.81**
128	5.78	4.86	**1.56**

7.1.3 The Shape of the Filters

In the previous experiments, the shape of the filters was fixed to rectangular. Next, we studied the effect of the filter shape. In addition to the rectangular shape, we studied triangular (Bartlett) and Hanning windows (see the previous Chapter for details). Notice that the bandwidths of the triangular and Hanning filters are about twice the bandwidth of the rectangular filter (see Fig. 6.2).

Table 7.3: Subband classification error rates (%) on the Helsinki corpus using different filters.

Codebook	**Filter shape**		
size	Rectangular	Triangular	Hanning
1	**7.35**	10.20	9.14
2	**4.74**	8.03	6.62
4	1.22	1.40	**1.16**
8	**0.00**	0.02	**0.00**
16	**0.00**	**0.00**	**0.00**
32	**0.00**	**0.00**	**0.00**
64	**0.00**	**0.00**	**0.00**
128	**0.00**	**0.00**	**0.00**

Table 7.4: Subband classification error rates (%) on the TIMIT corpus using different filters.

Codebook	**Filter shape**		
size	Rectangular	Triangular	Hanning
1	**49.49**	52.98	51.89
2	**30.60**	33.15	32.55
4	22.53	**20.86**	22.56
8	**3.85**	4.52	4.46
16	**0.65**	1.20	0.67
32	**0.50**	**0.50**	0.56
64	0.81	0.86	**0.80**
128	**1.56**	1.99	1.73

The classification results are shown in Tables 7.3 and 7.4. We observe that the rectangular filter gives the best results on average. The triangular and Hanning filters give similar results, the Hanning being slightly better. However, the results here are not perfectly comparable since the bandwidth of the rectangular filter is half of the triangular and Hanning filters. In other words, the rectangular filter has higher spectral resolution. On the other hand, the number of filters is the same in all three cases. We conclude that the filter shape itself does not play a critical role, as long as there are "enough" filters, 30 or more according to our experiments. The filter shape and the number of filters controls the smoothness of the spectral estimate. By decreasing the number of filters, or by using tapered and overlapping filters, the spectrum can be smoothened if so desired. The author's personal preference is the triangular-shaped filterbank for two reasons. First, it is easy to satisfy the desired property (4.17). Second, the triangular filter shape is the most commonly used in the mel-frequency cepstral coefficient (MFCC) computation [25, 58, 67]. By using the standard approach, we can more easily compare the subband processing and the corresponding FFT-cepstrum.

7.1.4 The Number of Subbands

The number of subbands affects the spectral resolution and therefore it is expected to be a very important parameter. In this experiment, we fixed the filter shape to triangular. Although the cube root seems to be the best nonlinearity, it is not significantly better than the logarithmic nonlinearity. Since the computation of the mel-cepstrum includes a logarithmic compression, we fix the nonlinearity here to logarithm. In this way, we can reduce the number of test runs since the subband processing of FFT-based cepstrum has already been partly tested. In general, one classification experiment requires a lot of CPU time.

In summary, in the next experiment the filter shape was triangular, the output nonlinearity was logarithm, and the filters were linearly spaced on frequency axis (no frequency warping). We varied the number of subbands from $M = 5$ to $M = 50$ with steps of 5, and the speaker codebook size was varied from $K = 1$ to 128. Selected results are shown in Figures 7.3 and 7.4.

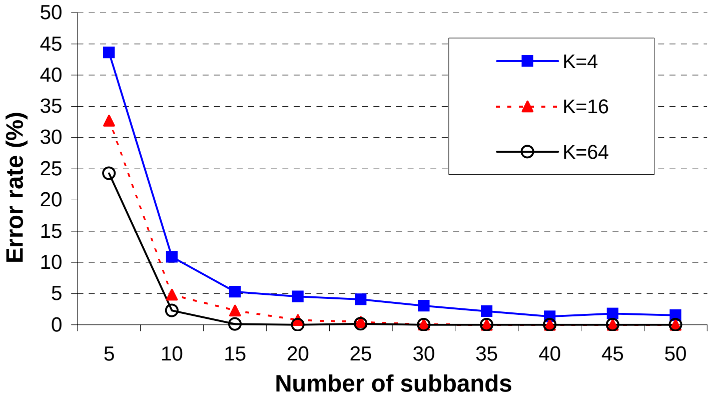

Figure 7.3: The effect of the number of subbands on the Helsinki corpus with different codebook sizes $(K = 4, 16, 64)$.

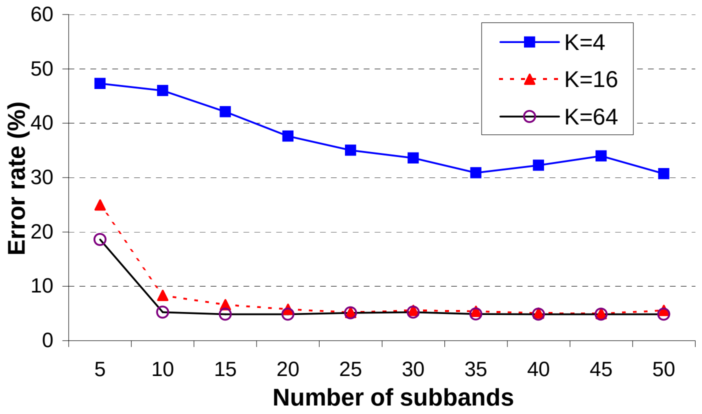

Figure 7.4: The effect of the number of subbands on the TIMIT corpus with different codebook sizes ($K = 4, 16, 64$).

As in previous experiments, larger speaker codebooks give consistently better results. A more interesting observation is that the number of subbands should be as high as possible. By increasing the number of subbands with any codebook size, we can reduce the error rate. This is an unexpected result. We hypothesized that the performance would degrade with too many subbands since the features do not anymore describe the spectral shape which is assumed to be more speaker dependent than the harmonic structure (of voiced sounds). Also, a high number of subbands results in highly correlated features which gives reason to expect that the performance would degrade. Furthermore, high-dimensional features require more training data in order to give a good estimate of the underlying feature distribution. Since we do not have any better explanation, we conclude that the fine structure of the spectrum contains significant amount of speaker information. Obviously, the number of subbands must be considerably higher than in the speech recognition task.

We conducted one more experiment by varying the number of subbands from 50 to 250 by steps of 50. These experiments are very time-consuming due to high dimensionality, and therefore we fixed the codebook size as low as 16. Note also that there are only 512 points in the zero-padded frame with these parameters, and thus the maximum number of points in the magnitude spectrum is 256. In the case of 256 filters, we would be using the (compressed) DFT output bins directly as the features. The classification results are given in Table 7.5. We observe that the TIMIT corpus shows monotonous increase in the error rate when the number of subbands increases. The Helsinki corpus also shows some oscillation in the error rates, but the increase is not monotonous.

Based on the corpora that we have used, we conclude that the number of subbands should be "high enough", and it depends on the corpora used. For the Helsinki corpus, any number of subbands is good, but for the TIMIT corpus, the optimum number seems to be around 30-50. Therefore, for the TIMIT corpus, the optimum filter bandwidth is 110-183 Hz. Interestingly, this range is close to the average $F0$ values (majority of speakers on the TIMIT subset are males). The number of subbands should be significantly higher than in speech recognition [125, 131]. Roughly speaking, this indicates that the "linguistic features" are in the smoothed spectrum whereas the "speaker features" are both in the smoothed spectrum and in the fine structure of the spectrum.

Running time is an important practical consideration in automatic speaker recognition. If the number of subbands is high, the running times easily become impractically high. Therefore, other spectral representations such as FFT-cepstrum can be considered. The main advantage of "raw" subband processing is that the features have a direct physical interpretation, and emphasizing/de-emphasizing certain frequency bands is easy.

Table 7.5: Classification error rates (%) using a high number of subbands (codebook size $K = 16$).

Number of	**Corpus**	
subbands	Helsinki	TIMIT
50	0.00	5.58
100	0.00	8.97
150	0.34	7.00
200	0.51	10.65
250	0.00	10.80

7.2 FFT-Based Cepstral Features

The previous results with the filterbanks demonstrated the effectiveness of subband-based representations. The computation of the FFT-cepstrum includes subband analysis, followed by filter output decorrelation using the DCT. In the next experiment, the number of filters was fixed to 30, the filters were triangular-shaped and their outputs were compressed using the logarithm. The selected cepstral coefficients included $c[1], c[2], \ldots, c[15]$. The zeroth coefficient was excluded since it depends on the frame intensity. It is worth noting that the dimensionality of the feature space is half of the one used with the raw filterbank outputs.

The classification results using the FFT-cepstral parameters are presented in Tables 7.6 and 7.7. With small codebook sizes, the differences between different frequency warpings are larger and the differences become smaller with increasing codebook size. Interestingly, for the Helsinki corpus the mel-scale gives the best results on average, whereas the linear frequency scale seems to be the best for the TIMIT corpus. The difference can be understood from the discrimination values of individual subbands (see Figures 7.1 and 7.2). The discriminative frequency bands are located in different frequencies for the two corpora. For the TIMIT corpus the higher frequencies are more important, but the mel-warping de-emphasizes the spectral detail of these frequencies.

Table 7.6: Classification error rates (%) on the Helsinki corpus by using FFT-cepstrum with different frequency warpings.

Codebook	**Frequency warping**			
size	Linear	Mel	Bark	ERB
1	24.71	**24.67**	25.19	27.79
2	15.65	**13.31**	13.64	16.30
4	6.05	**2.84**	2.88	4.83
8	3.90	**3.17**	3.47	3.54
16	0.32	**0.27**	1.88	0.59
32	0.52	0.03	**0.02**	0.12
64	0.35	0.20	**0.01**	0.63
128	0.16	**0.00**	0.07	**0.00**

Table 7.7: Classification error rates (%) on the TIMIT corpus by using FFT-cepstrum with different frequency warpings.

Codebook	**Frequency warping**			
size	Linear	Mel	Bark	ERB
1	**60.18**	60.65	63.56	80.51
2	47.17	40.32	**38.17**	42.97
4	40.50	38.61	38.18	**37.87**
8	**12.67**	13.60	13.56	17.28
16	**5.88**	6.77	6.24	7.04
32	**5.16**	5.85	5.76	6.12
64	**4.84**	5.36	4.92	5.30
128	5.09	**4.84**	**4.84**	5.11

Both the number of filters and the number of cepstral coefficients affects the spectral resolution, and we therefore consider these two as the most crucial parameters. We selected the mel-frequency warping due to its popularity and the results obtained in the previous experiment. The number of subbands was fixed (30,40,50) while the number of coefficients was varied from 1 to 20, starting from the coefficient $c[1]$. The classification results for codebook size $K = 64$ are shown in Figures 7.5 and 7.6.

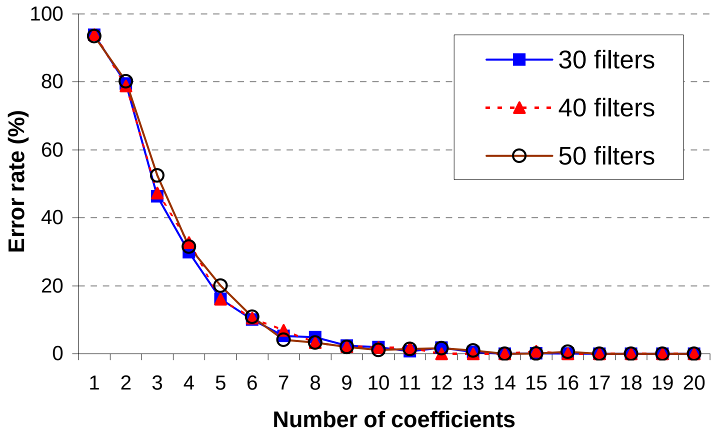

Figure 7.5: Classification results on the Helsinki corpus using mel-cepstral coefficients (codebook size $K = 64$).

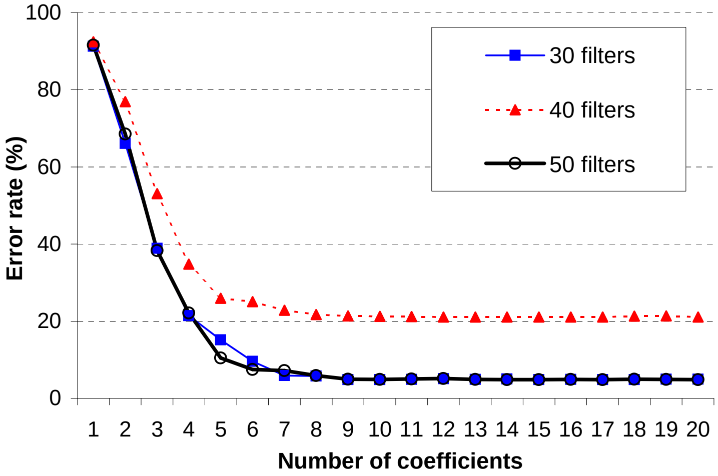

Figure 7.6: Classification results on the Helsinki corpus using mel-cepstral coefficients (codebook size $K = 64$).

With both corpora, the error rate decreases and stabilizes with the increasing number of coefficients. The error rate of Helsinki corpus reaches zero error rate with approximately 12 coefficients. The TIMIT corpus, in turn, stabilizes approximately with 9 coefficients, but the error rate does not drop to zero. An unexpected result is that in the case of 40 filters the error rates are clearly higher than with 30 and 50 filters. This is likely due to the specific populations we chose for our experiments. It seems in general that the TIMIT speakers are more difficult to model than the Helsinki speakers.

When the frequency axis is warped nonlinearly, the control over different frequency bands becomes difficult. Therefore, a methodologically more sound protocol is to study how the number of filters effects the results if the frequency warping is linear instead of the mel-scale. By using a linear frequency scale, each frequency band gets equal importance. In the following experiment, we fixed the number of coefficients to 15 and the speaker model size 64, while the number of filters was varied from 20 to 50 with steps of 5. The classification results are shown in Table 7.8. It can be seen that the number of filters does not have much effect. We conclude that the number of cepstral coefficients is the most critical parameter. Based on our results, about 15 coefficients seems to be enough for a fixed number of subbands. This is supported by the known property of the cepstral coefficients: the magnitude of the coefficients decays fast ($\sim 1/n^2$) with increasing coefficient index, and therefore the higher coefficients have only a minor contribution to the distance or likelihood values.

Table 7.8: Classification error rates (%) using linear frequency scale (15 cepstral coefficients, codebook size $K = 64$).

Subbands	**Corpus**	
	Helsinki	TIMIT
20	0.11	4.84
25	0.09	4.86
30	0.49	4.96
35	0.53	4.84
40	1.22	4.86
45	0.00	4.95
50	0.23	4.85

7.3 Linear Prediction Based Features

7.3.1 LPC Coefficients

Since all of the LPC-derived features are based upon the all-pole model specified by the predictor polynomial, it is natural to start by studying the performance of the LPC coefficients themselves. It is often stated that LPC coefficients should not be used directly, and there have been several propositions for the distortion measures between two LPC vectors [131, 67]. However, these statements are often done regarding the *speech* recognition task. For this reason, and due to very simple implementation, we apply the Euclidean distance in both the codebook generation (training) and in the recognition phase as previously. The pre-processing parameters are the same as before (a 30 ms Hamming window, shifted by 20 ms, adaptive pre-emphasis). The predictor coefficients are computed from the autocorrelation sequence using the Levinson-Durbin procedure. We vary the LPC predictor order ($p = 5, 6, \ldots, 30$) and the codebook size ($K = 16, 32, 64$). The results are shown in Figures 7.7 and 7.8.

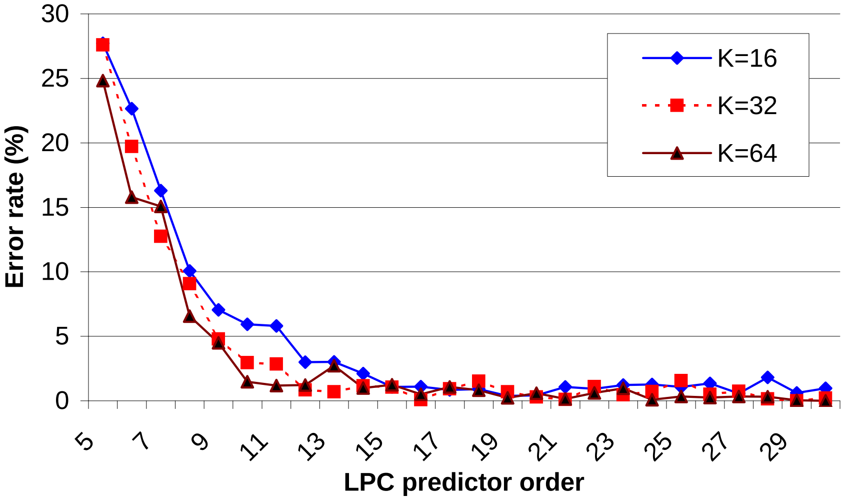

Figure 7.7: Performance of the LPC coefficients on the Helsinki corpus for different codebook sizes ($K = 16, 32, 64$).

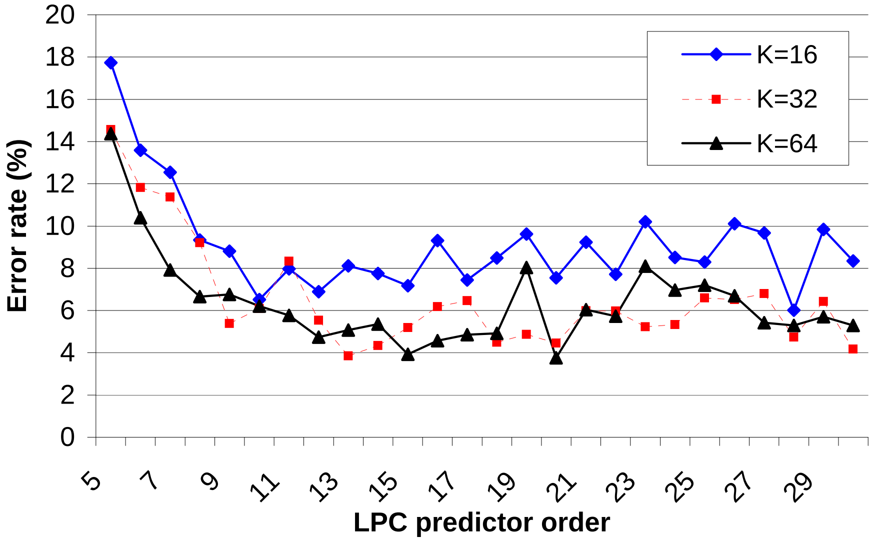

Figure 7.8: Performance of the LPC coefficients on the TIMIT corpus for different codebook sizes ($K = 16, 32, 64$).

We observe that the LPC coefficients give good results. The result is somewhat unexpected due to all warnings in literature to not to use LPC coefficients directly. However, the result is consistent with Reynolds' results who used Gaussian mixture modeling of the LPC coefficients [133]. The results here show that for the speaker recognition task, the LPC coefficients are potentially a good feature set even without any normalization and with the simplest distance measure. Increasing the codebook size reduces error rates as previously, although the results are not consistent for all predictor orders. The performance improves when more coefficients (higher order predictors) are used. Both corpora reach their minimum error rates around $p = 15$ coefficients. Whereas the Helsinki corpus shows saturation after this, the TIMIT corpus shows a slight increase in the error rate after this.

In literature, it is said that voiced speech sounds have approximately one complex pole per kilohertz and that 1-2 complex poles should be allocated to account for the glottal and lip radiation effects [58]. In our case ($F_s = 11.025$ kHz $\rightarrow$ 11 poles), this rule would suggest using about 12-13 poles, in other words, a linear predictor of order $p = 12 - 13$. Although the rules is heuristic, it gives a rough idea for the needed predictor order. In our case, the rule seems to give the *minimum* number of coefficients.

7.3.2 LPC-Derived Feature Sets

Based on the previous experiment, we fixed the predictor order to $p = 15$, and computed the other LPC-derived feature sets using this predictor. First, we fixed the codebook size to $K = 64$ and varied the number of coefficients. In each case, we selected the lowest coefficients as the features. The classification results are shown in Figures 7.9 and 7.10. For the TIMIT corpus, we decided to leave out the REFL feature set curve for clarity since it produced extremely poor results (error rate > 60 %). In general, we faced more numerical problems with the TIMIT corpus.

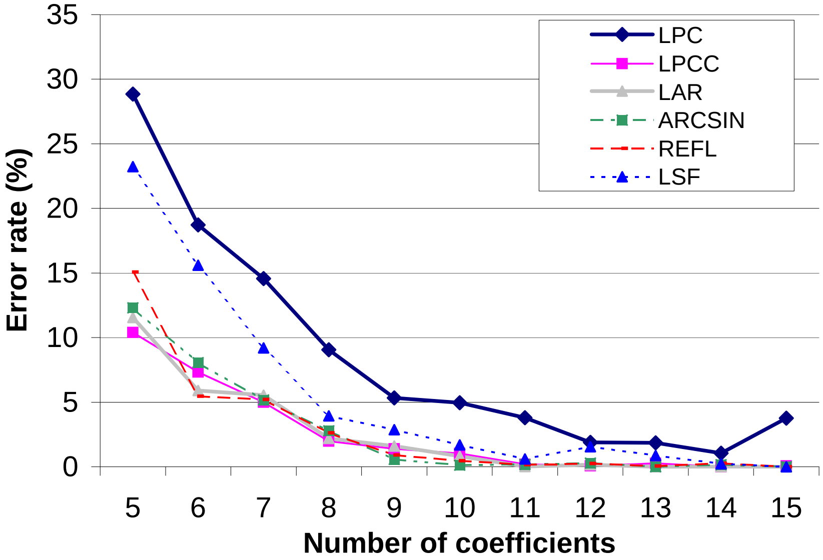

Figure 7.9: Performance of the LPC-derived features using the Helsinki corpus (predictor order $p = 15$, codebook size K=64).

We observe that all tested LPC-derived features outperform the raw LPC coefficients, provided that the number of coefficients is high enough. For the Helsinki corpus, we observe that the LSF coefficients give poor results for a small number of coefficients. The rest of the features (LPCC, LAR, ARCSIN, REFL) give better results in general. However, when the number of coefficients is increased, the LSF features perform equally well. For the TIMIT corpus, similar behavior can be seen. For the TIMIT corpus, LPCC and ARCSIN perform the best with a small number of coefficients, but the other feature sets give equal performance when the number of coefficients is increased. For these two corpora, the alternative LPC-derived features give approximately equally good results given that the LPC predictor order

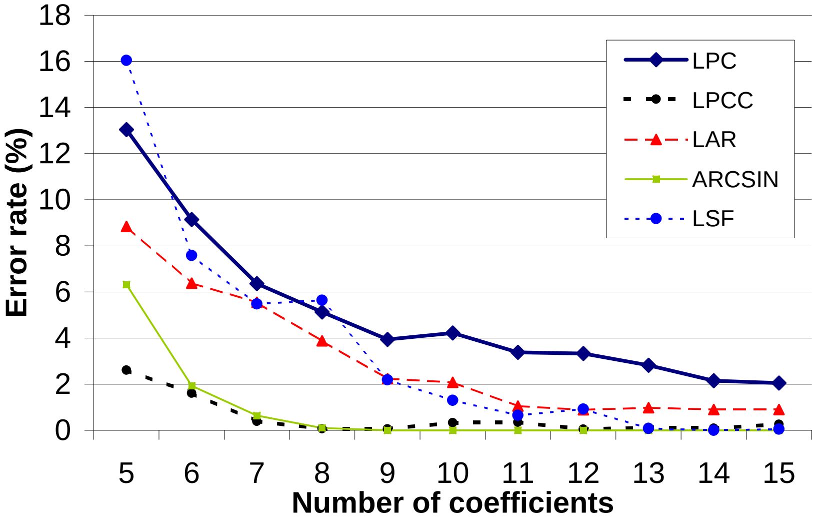

Figure 7.10: Performance of LPC-derived features using the TIMIT corpus (predictor order $p = 15$, codebook size K=64).

(≥ 15) and number of coefficients ($\geq 10 - 12$) are sufficiently high. The only exception is the REFL feature set, which gives an error rate > 60 % for the TIMIT corpus. The reason is unknown.

Next we fixed the predictor order to $p = 15$ and varied the codebook size from $K = 16$ to $K = 256$ with powers of two. The results are given in Tables 7.9 and 7.10. As a general conclusion we can say that most of the LPC-derived features give good results when the codebook size is increased. For the corpora used herein, a zero error rate can be reached with almost all feature sets. The raw LPC coefficients give most often the worst results.

Two negative observations can be made from Tables 7.9 and 7.10. The first one is the poor performance of the REFL feature set on the TIMIT corpus. However, for the Helsinki corpus this feature set gives very good results! The other observation is that the LPCC feature set gives good results otherwise, but for the codebook size $K = 64$ on the TIMIT corpus the error rate is almost 100 %! These inconsistencies give rise to two possible explanations. First, the Euclidean distance measure might not be good for some features. Second, the possibility of a programming bug is not excluded.

Table 7.9: Performance of the LPC-derived features ($p = 15$) on the Helsinki corpus with different codebook sizes.

Codebook	**Feature set**					
size	LPC	LPCC	LAR	ARCSIN	REFL	LSF
16	3.57	0.36	0.33	1.29	1.00	**0.12**
32	2.15	0.26	**0.00**	0.01	0.25	1.11
64	3.77	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
128	1.43	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
256	1.59	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**

Table 7.10: Performance of the LPC-derived features ($p = 15$) on the TIMIT corpus with different codebook sizes.

Codebook	**Feature set**					
size	LPC	LPCC	LAR	ARCSIN	REFL	LSF
16	2.85	0.39	0.43	**0.00**	61.61	0.11
32	2.04	**0.00**	0.76	**0.00**	60.77	0.36
64	2.05	97.26	0.90	**0.00**	62.14	0.05
128	1.96	**0.00**	0.50	**0.00**	63.17	**0.00**
256	3.20	**0.00**	0.94	**0.00**	64.10	**0.00**

We studied also the means and differences of adjacent line spectral pairs (*MALS* and *DALS*, respectively) as suggested by Liu & al. [93]. The MALS feature set gave systematically worse results than the original LSF parameters, whereas DALS feature set outperformed LSF in some cases. In the original paper, DALS feature set was reported to give good results, and in this sense the results are consistent. The results on Helsinki corpus were good, but for TIMIT the performance was very poor in all cases (error rates $\geq$ 60 %). It seems that the TIMIT corpus yields occassionally very poor results with the LPC-derived features.

7.3.3 LPC-Derived Formants

Next, we studied LPC-derived formant frequencies and their bandwidths. We varied the number of LPC coefficients from $p = 5$ to $p = 15$, fixing the codebook size to $K = 64$. Given a fixed predictor order p, we selected the minimum number of poles in the Nyquist range[1]. Visual inspection of the

[1] The number of complex poles in the Nyquist range can vary from frame to frame. The number of LPC poles N_p in the upper half plane is at least $p/2$ and $(p-1)/2+1$ for even and odd p, respectively. Thus, for $p = 15$ we have 8 formants.

formant tracks showed that the analysis procedure worked reasonably well. We noticed that for several frames, especially higher order predictors, two of the poles were very close to the frequencies 0 and $F_s/2$, which raises a doubt that these are LPC analysis artefacts. However, we did not make attempts to remove these since we wanted to keep the test protocol simple.

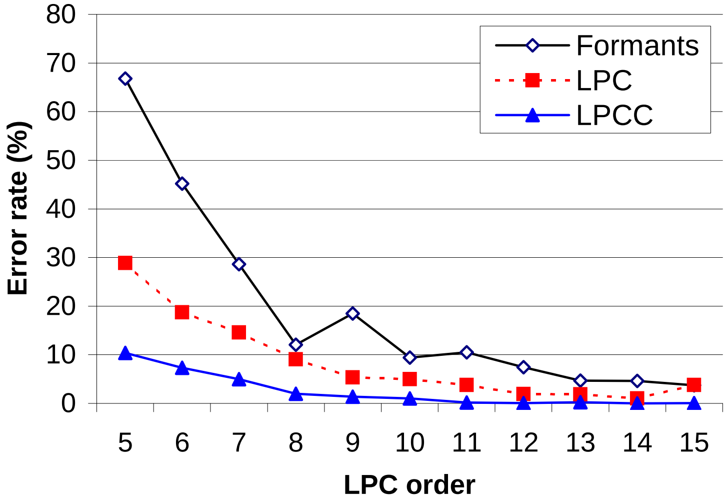

Figure 7.11: Performance of LPC-derived formants on the Helsinki corpus, compared with LPC and LPCC (codebook size K=64).

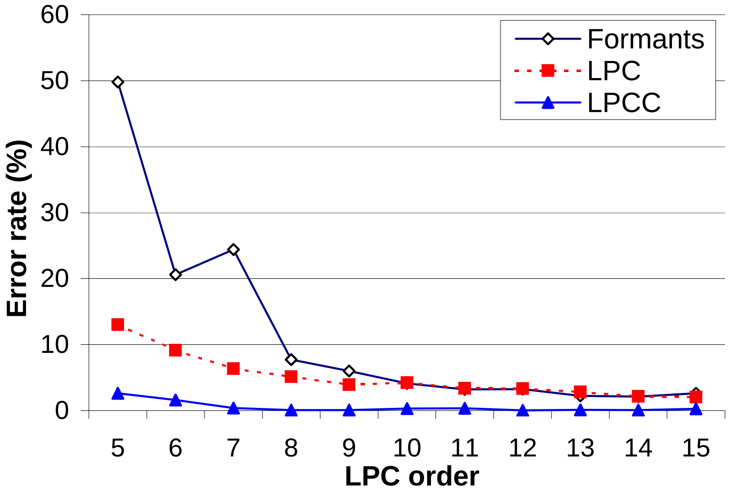

Figure 7.12: Performance of LPC-derived formants on the TIMIT corpus, compared with LPC and LPCC (codebook size K=64).

We noticed that increasing the analysis order decreased error rates as before, for both the formant frequencies and their bandwidths. The formant bandwidths gave very poor results for both corpora (error rates ≥ 80 %). However, the formant frequencies gave surprisingly good results. The comparison of the formant frequencies with the LPC and LPCC feature sets are shown in Figures 7.11 and 7.12. We observe that the formants give worse results than the LPC coefficients in general. Therefore, the formants have poorest discrimination properties of all LPC-derived features. Nevertheless, when the LPC order is high, formants give good results. Although the formant estimation procedure is the most simple one and produces a lot of spurious formants, it shows that a considerable amount of speaker information might be included only in the formant frequency *locations*.

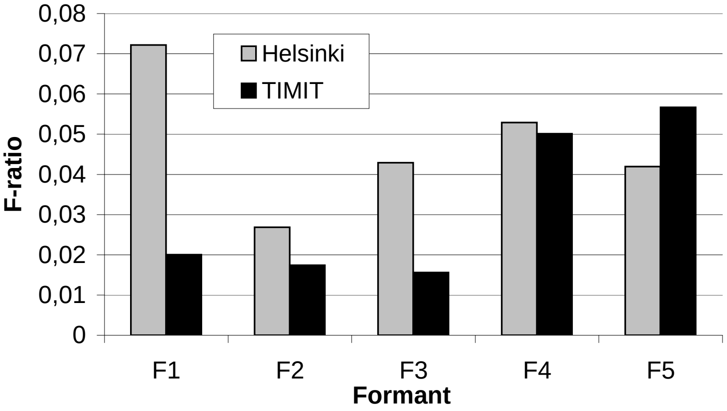

Figure 7.13: Discrimination of formant frequencies (LPC order $p = 10$).

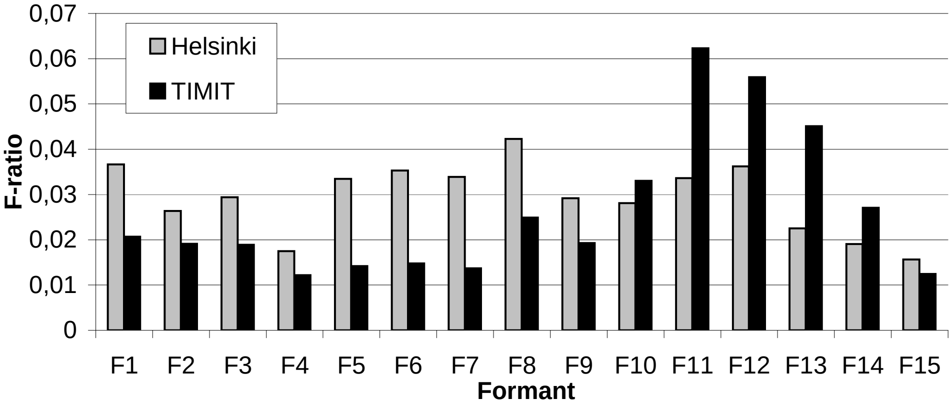

Figure 7.14: Discrimination of formant frequencies (LPC order $p = 30$).

An interesting question is whether the higher formant frequencies are more discriminative as it is generally believed. In particular, it is stated in many textbooks that the two lowest vowel formants ($F1, F2$) contain roughly most of the phonetic information, whereas $F3$ and higher formants are more speaker-specific. We computed the F-ratios of each formant for several LPC orders. The results for orders $p = 10$ and $p = 30$ are shown in Figures 7.13 and 7.14, respectively[2]. It seems that the higher formants indeed are more discriminative, especially for the TIMIT corpus. The Helsinki corpus shows more uniform discrimination over the different formants.

We must be cautious in making any "phonetic interpretation" of the results of Figures 7.13 and 7.14, since we did not segment the data. In other words, only the *average* discrimination values over all different phonemes are taken into consideration. This procedure averages out the effect of phonemes that are discriminative but infrequent. Also, the analysis procedure produces probably a lot of spurious formants. However, reliable formant estimation is a research topic on its own right and is not in the scope of this thesis.

Some similarity of the formant F-ratios with the subband discrimination values of Figures 7.1 and 7.2 can be observed. In particular, the TIMIT corpus shows a peak in high frequencies and higher formants. Although both the subband features and formants suggests that certain frequencies are better than some others, they contain somewhat different information. The subband feature set contains *relative magnitudes* of given frequency band(s), whereas the formant feature set includes the *locations* of high-magnitude frequency regions (resonances). This raises a question whether we could combine these two different information sources somehow. The first idea that comes to mind is to use formant estimation in the *selection* of the subbands for subband-based features. In other words, the frequency bands whose relative magnitude is high (indicated by the formant locations) would be selected as a basis for the subband processing, or for the FFT-cepstrum. This kind of procedure might be good for noise-corrupted speech, since perceptually more noise can be tolerated around the spectral peaks [100].

7.3.4 LSF Parameters Revisited

Although not showing consistently better results in the previous experiments, the LSF parameters have many attractive properties which gave us motivation to study them in more detail. First, the LSF parameters are known to have good quantization properties, and consequently, they might be suited

[2] Notice that the number of formants is half of the predictor order since the complex conjugate pairs of the LPC polynomial roots represent the mirror image of the spectrum around the Nyquist frequency.

well for VQ-based speaker recognition. Second, LSF's have more direct interpretation than, for instance, the LPC coefficients or the LPCC coefficients. The lower order LSF coefficients correspond to lower frequencies, and LSF's are closely related to the formant structure [93]. This allows easy control of weights given to certain frequency bands in the distance or likelihood calculations. Third, high recognition rates have been reported in several studies [93, 91, 20, 169, 109].

We studied the effects of the LPC analysis order ($p = 10, 20, 30$) and the codebook size ($K = 8, 16, \ldots, 256$) to the performance of the LSF parameters. For comparison, we selected the ARCSIN feature set, since it gave good results on both corpora (see Tables 7.11 and 7.12). We can see that both feature sets behave nicely on both corpora, and as expected, increased analysis order and model size decreases error rates. Both representations seem to be very effective in that they converge near the zero error rate with a small number of coefficients and a small number of code vectors. From the two feature sets, the ARCSIN seems to outperform LSF in most cases, but the difference is small. From a practical viewpoint, ARCSIN computation is simpler to implement and computationally more efficient.

Table 7.11: Comparison of ARCSIN and LSF feature sets for different LPC model order and codebook sizes on the Helsinki corpus.

Codebook	$p = 10$		$p = 20$		$p = 30$	
size	ARCSIN	LSF	ARCSIN	LSF	ARCSIN	LSF
8	4.96	4.58	0.13	2.02	0.00	0.26
16	2.74	3.86	0.75	0.51	0.27	0.19
32	0.87	0.55	0.00	0.23	0.00	0.06
64	0.13	0.99	0.00	0.32	0.00	0.64
128	0.60	0.11	0.00	0.30	0.00	0.28
256	0.98	0.18	0.00	0.00	0.00	0.56

Table 7.12: Comparison of ARCSIN and LSF feature sets for different LPC model order and codebook sizes on the TIMIT corpus.

Codebook	$p = 10$		$p = 20$		$p = 30$	
size	ARCSIN	LSF	ARCSIN	LSF	ARCSIN	LSF
8	3.76	3.81	2.13	0.58	1.98	1.96
16	0.26	0.97	0.00	0.82	0.30	0.36
32	0.05	0.34	0.00	0.40	0.07	0.03
64	0.17	0.86	0.00	0.54	0.00	0.00
128	0.00	0.21	0.00	0.00	0.00	0.00
256	0.00	0.51	0.00	0.25	0.00	0.00

7.4 Delta Parameters

In this section, we study the delta features. Based on the previous experiments with the static features, we consider the delta parameters of the following feature sets:

- Log-compressed triangular filterbank outputs (FB)
- Linear frequency FFT-cepstrum (FFT-cep)
- Linear predictive cepstral coefficients (LPCC)
- Arcus sine coefficients (ARCSIN)
- Line spectral frequencies (LSF)

We fixed the parameters as follows. A 30 milliseconds Hamming window with 25 % (7.5 milliseconds) overlap between adjacent frames was used. The filterbank consisted of 30 filters. The FFT-cepstrum was computed using the same filterbank, and 15 lowest cepstral coefficients (excluding $c[0]$) were retained. A linear predictor of order $p = 15$ was used in computing the LPCC, ARCSIN and LSF parameters. We studied both the *differentiator* and the *linear regression* method for the delta feature computation. The feature streams were augmented with zero vectors from both ends.

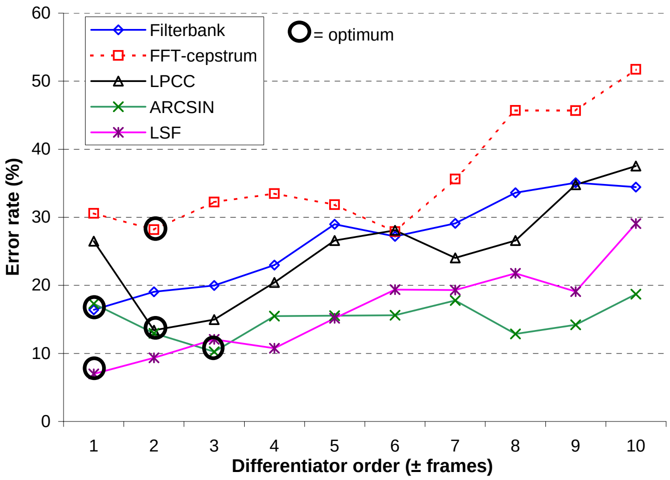

Figure 7.15: Performance of the delta features on the Helsinki corpus using the *differentiator* method (codebook size $K = 64$).

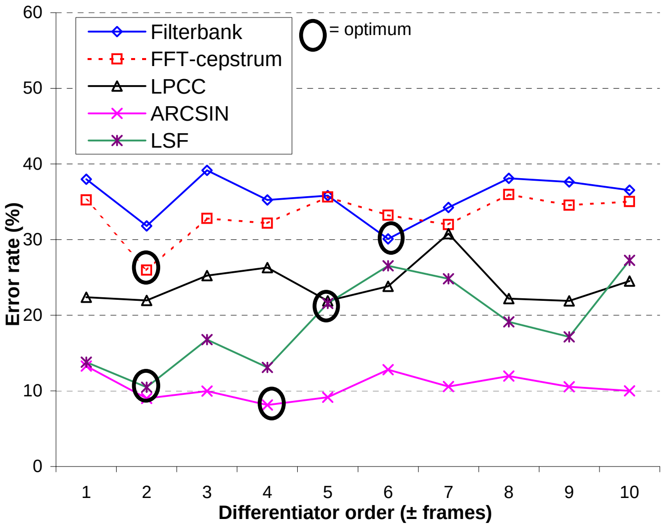

Figure 7.16: Performance of the delta features on the TIMIT corpus using the *differentiator* method (codebook size $K = 64$).

First, we studied the differentiator method by varying the differentiator order M from $M = \pm 1$ to ± 10 frames. The results for codebook size $K = 64$ are shown in Figures 7.15 and 7.15. We have marked the optimum point for each delta feature set with circles. For the Helsinki corpus, increasing the differentiator order tends to increases the error rates. The optimum differentiator order is 1-3 frames for all feature sets. The delta-LSF parameter with $M = 1$ gives the lowest error rate (7.0 %), whereas the delta-cepstrum (FFT) gives the highest error on average. The delta parameters of LPC-derived feature sets give the lowest error rates. The TIMIT corpus shows also increase in error rates with increasing differentiator order, but not as clearly as the Helsinki corpus. For the TIMIT corpus, the optimum order is between 2-6. The lowest error rate (8.1 %) is obtained using delta-ARCSIN with $M = 4$. Consistent with the Helsinki corpus, the LPC-derived delta features outperform the subband-based delta parameters (filterbank, FFT-cepstrum).

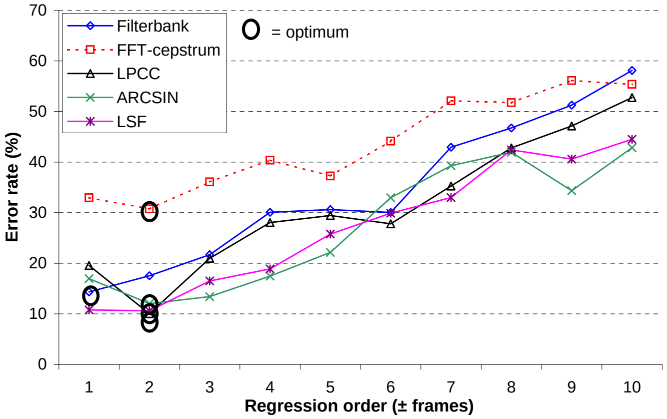

Figure 7.17: Performance of the delta features on the Helsinki corpus using the *linear regression* method (codebook size $K = 64$).

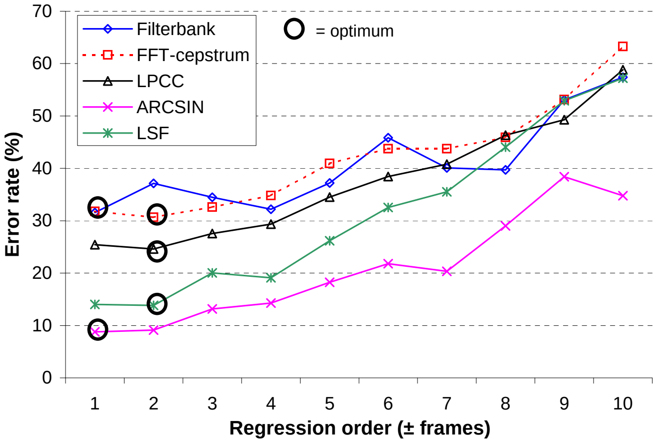

Figure 7.18: Performance of the delta features on the TIMIT corpus using the *linear regression* method (codebook size $K = 64$).

Next, we studied the linear regression method by varying the regression order M from $M = \pm 1$ to ± 10 frames. The results for codebook size $K = 64$ are shown in Figures 7.17 and 7.17. For both corpora, the optimum regression order is between 1-2 frames. This makes setting the control parameters more easier than for the differentiator method. Also, for both corpora the delta-LSF parameter set gives the smallest error rate (10.6 and 14.0 % for the Helsinki and TIMIT corpus, respectively).

Next, we study the differences between the two delta computation methods. For this, we selected the delta-LSF and delta-ARCSIN methods from the previous figures since these gave the best results. The comparison of the methods is shown in Figures 7.19 and 7.20 for codebook size $K = 64$. Both corpora and both feature sets show that the differentiator method gives smaller error rates when the order is selected correctly. This is a pretty surprising result, and we are forced to conclude that our test hypothesis was wrong. The regression method probably oversmooths the parameter trajectories, discarding rapid spectral changes that the differentiator method is able to capture. The ordering of the methods may be opposite in noisy conditions, but for clean speech our results suggest quite clearly that the differentiator method is better. The advantage of the regression method based on these results is that its parameters are more easier to set - the optimal regression order is 1 or 2 frames for all feature sets and both corpora with the frame rate used here.

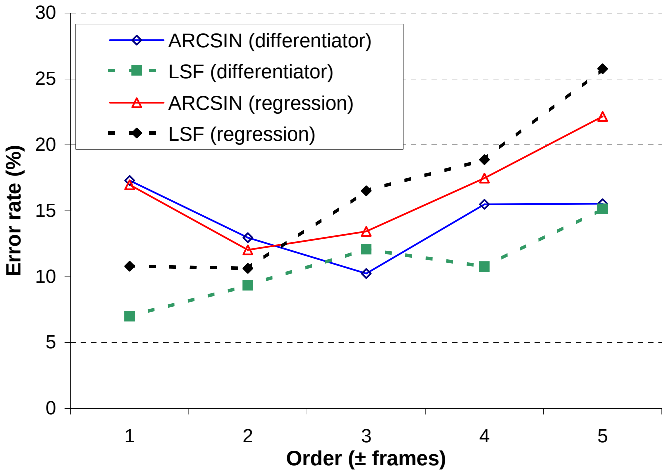

Figure 7.19: Comparison of the differentiator and regression method on the Helsinki corpus.

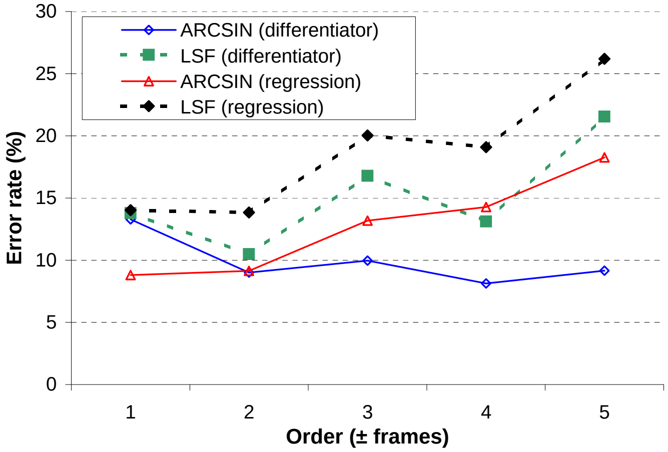

Figure 7.20: Comparison of the differentiator and regression method on the TIMIT corpus.

Next, we compared the static and dynamic features by selecting the best results for the codebook size $K = 64$. The results are given in Table 7.13. We observe that the dynamic features give higher error rates, which was expected. The delta parameters of the LPC-derived features (LPCC, ARCSIN, LSF) outperform the delta parameters of the subband-based features (filterbank, FFT-cepstrum). Based on these results, the LPC-derived feature sets along with their delta-parameters seem to be the best choice.

Higher order delta parameters (delta-deltas) are sometimes also used. They should not be ignored even though we have excluded them from our experiments because of lack of time. In fact, joint optimization of the delta- and delta-delta orders would be an interesting additional test to carry out in future in order to evaluate their usefulness in practice.

Since the static and dynamic features encode different information, it is expected that they could be efficiently combined [149]. However, we did not study the joint performance of static and dynamic features, since this raises new design issues that are out of the scope of this thesis. First, we should decide the fusion method: (1) input fusion (vector concatenation) or (2) classifier fusion (separate codebooks for static and dynamic features as in [149]). For the input fusion, we should do some feature normalization since the dynamic features have different range than the original features. For the classifier fusion, we should decide the method of combining the classifier output scores, as well as to give the combination weights for each classifier output.

Table 7.13: Comparison of static and dynamic features (codebook size $K = 64$).

Corpus	**Feature type**	**Feature set**				
		FB	FFT-cep	LPCC	ARCSIN	LSF
Helsinki	Static	0.00	0.35	0.06	0.00	0.00
	Dynamic	14.35	27.90	10.05	10.23	6.99
TIMIT	Static	4.84	4.84	0.25	0.00	0.05
	Dynamic	30.08	25.98	21.88	8.12	10.49

7.5 Concluding Remarks

7.5.1 Cepstrum Revisited

The previous experiments have shown that different forms of cepstrum are a powerful presentation. Throughout the experiments, we have used a linear scale and logarithmic filter output compression in the FFT-cepstrum computation. The linear frequency scales was selected since we felt that controlling the parameters was easier than for the nonlinear frequency scales. The logarithmic compression was selected since it is typically used in the mel-cesptrum computation. However, the cube root compression gave good results with the filterbank presentation. For this reason, we wanted to make a comparison between the logarithmic and cubic nonlinearities. The following parameters were fixed: frame length 30 milliseconds, frame overlap 25 %, and Hamming window function. We used the lowest 15 cepstral coefficients, excluding $c[0]$ as before. We varied the codebook size from $K = 16$ to 256. After some experimenting, we decided to use a test segment length of 100 vectors in the classification experiments so that differences between methods could be seen (for the previously used 350 vector segment, majority of the error rates were already 0.00 %).

Table 7.14: Classification error rates (%) for different FFT-cepstrum presentations on the Helsinki corpus (segment length = 100 vectors, codebook size $K = 64$).

Codebook size	**Non-linearity**	**Frequency warping**	
		Linear	Mel
16	log	10.29	6.85
	cube root	8.17	6.85
32	log	7.86	6.44
	cube root	7.27	6.44
64	log	6.55	5.75
	cube root	5.71	5.75
128	log	6.15	5.26
	cube root	8.28	5.26
256	log	6.06	5.68
	cube root	4.45	4.68

The classification results are shown in Tables 7.14 and 7.15. We observe that using the cubic compression decreases error rates in *all cases*. We conclude that if the raw subband outputs or subband-based cepstrum is used, then the filter outputs should be compressed using cube root instead of logarithm. Regarding the frequency warping, we observe that the linear scale is better for the TIMIT corpus, whereas the opposite is true for the Helsinki corpus. This is explained by the differences in the relative importances of different subbands as discussed before. We conclude that there is no globally optimal frequency warping method, but it must be tailored for each corpus.

Although the results show that the mel-scale is better than linear scale in some cases, the author prefers to use a linear-frequency filterbank. In this way, controlling of the important frequency bands is more easy and the implementation is also more simple. The problem reduces then to estimating the importances of the individual subbands, which can be done for instance using the F-ratio or other separability criterion. These values can then be used as weights in the distance or likelihood function. An interesting research topic would be to establish a connection between the linear and warped filterbanks. For instance, if we are using a linear frequency scale, how should be the filter outputs weighted so that the resulting outputs would approximate close to the outputs of the mel-frequency filterbank.

Table 7.15: Classification error rates (%) for different FFT-cepstrum presentations on the TIMIT corpus (segment length = 100 vectors, codebook size $K = 64$).

Codebook size	**Non-linearity**	**Frequency warping**	
		Linear	Mel
16	log	20.31	26.60
	cube root	14.18	17.49
32	log	14.56	22.35
	cube root	10.61	12.28
64	log	12.28	17.47
	cube root	8.38	10.61
128	log	10.74	16.69
	cube root	8.28	10.53
256	log	9.69	15.96
	cube root	7.59	8.89

Table 7.16: Classification error rates (%) for different FFT-cepstrum presentations on the Helsinki corpus (segment length = 100 vectors, codebook size $K = 64$).

Codebook size	**Non-linearity**	**Frequency warping**	
		Linear	Mel
16	log	20.31	26.60
	cube root	14.18	17.49
32	log	14.56	22.35
	cube root	10.61	12.28
64	log	12.28	17.47
	cube root	8.38	10.61
128	log	10.74	16.69
	cube root	8.28	10.53
256	log	9.69	15.96
	cube root	7.59	8.89

Finally, we compared the FFT- and LPC-based cepstral representations. It is often stated that the FFT-cepstrum is more accurate but the LPC-cepstrum is computationally more efficient. Since the main focus of this thesis is not the time complexity[3], we are interested in the recognition accuracy

[3] Besides, most of the computation time is not spent on the feature extraction but distance computations in the recognition phase.

only.

As before, the frame length and overlap were set to 30 milliseconds and 25 %, and the Hamming window was used as the window function. Since the FFT-cepstrum is based on non-parametric spectrum modeling, and the LPC-cepstrum is based on parametric (all-pole) modeling, it is hard to set parameters that are perfectly comparable. For the FFT-cepstrum, we chose to use 30 and 50 subbands, and both the log and cube root compressions. For the LPC-cepstrum, we used LPC-predictors of orders $p = 15$ and 30. For both feature sets, the frequency axis was linear (no warping), and the number of cepstral coefficients was set to 15. The test segment length was fixed to 100 vectors, and the codebook size was varied from $K = 15$ to 256. The results are shown in Figures 7.21 and 7.22.

We observe that for correct parameter selection, both methods give good results. However, a pretty surprising observation is that the LPC-cepstrum seems to outperform the FFT-cepstrum. This can be seen clearly in the case of the TIMIT corpus, and the Helsinki corpus also shows the same tendency. The performance of the FFT-cepstrum can be improved by using more subbands. However, notice that this applies only to the cube root compression - for the logarithm, increasing subbands increases errors in most cases. The performance of the LPC-cepstrum can be improved by increasing the LPC analysis order, but not in all cases.

For the codebook size $K = 256$, both corpora show the same grouping of the performances: the log FFT-cepstrum gives the poorest results and the LPC-cepstrum the best results, the cube root FFT-cepstrum being between these two. Since the ordering is consistent for both corpora, we conclude that the LPC cepstrum should be used. If the FFT-cepstrum is used, then the cube root should be used as already indicated clearly before. The results of this thesis do *not* show that the FFT-cepstrum would be more accurate. However, the advantage of FFT-cepstrum is that it is more simple to implement[4].

[4] The author faced several problems with the Levinson-Durbin recursion and the LPC → LPCC formula (5.19). On the other hand, this was the first the the author was working with the LPC model - next time the programming might be more easy.

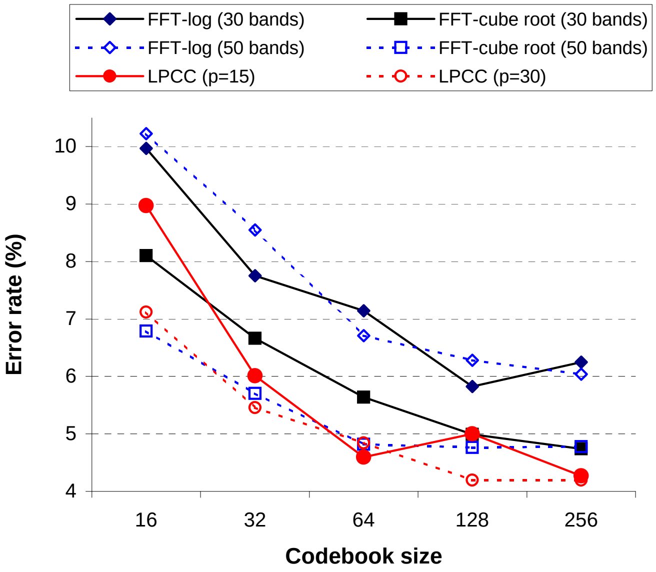

Figure 7.21: Comparison of the FFT- and LPC-cepstral presentations on the Helsinki corpus (segment length = 100 vectors) .

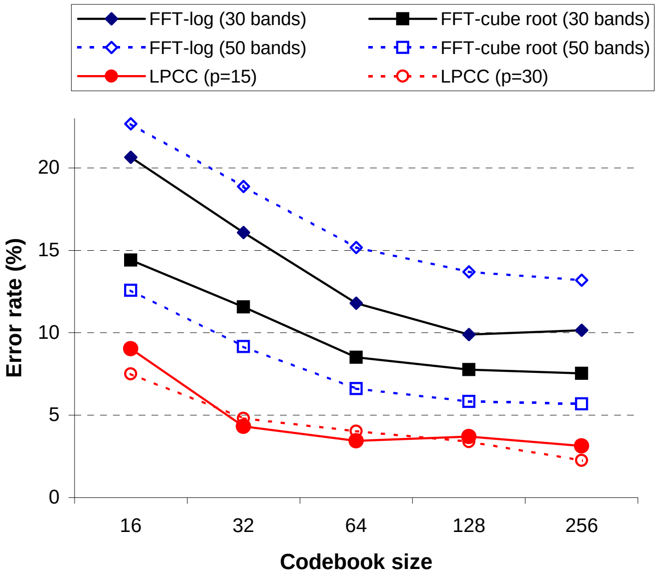

Figure 7.22: Comparison of the FFT- and LPC-cepstral presentations on the TIMIT corpus (segment length = 100 vectors).

7.5.2 Implementation Viewpoint

We have been able to show that the commonly used feature sets differ in their performance. Specifically, and somewhat surprisingly, the LPC-based features seem to be competitive with the FFT-based features, and in many cases they seem to give clearly better performance. However, the recognition accuracy is only one issue in the selection of the feature extraction method for a specific applications. Other issues include the time- and memory consumption and ease of implementation.

We have not made a detailed time complexity analysis of the feature extraction methods since all of these include several steps that can be implemented in different ways. Summary of the computation steps of some of the methods are summarized in Table 7.17. The filterbank and FFT-cepstrum are straightforward to implement. Of the LPC-based features, the ARCSIN feature is the simplest one to implement, since the features are computed directly from the reflection coefficients that are a side product of the Levinson-Durbin recursion. The LPCC parameter set is also straightforward to compute from the predictor coefficients using the Atal's recursion formula [6]. Computation of the LSF feature set includes finding the roots of two complex polynomials. For this, there are very likely efficient algorithms since the roots of the two polynomials lie on the unit circle and they are interlaced with each other. The author used the Matlab software where computation of LSF parameters is a built-in procedure.

Table 7.17: Summary of steps needed in extraction some of the features evaluated in this thesis.

Feature set	**Steps**
Filterbank	FFT subband analysis
FFT-cepstrum	FFT subband analysis DCT
ARCSIN	Autocorrelation computation Levinson-Durbin recursion
LSF	Autocorrelation computation Levinson-Durbin recursion Finding the roots of two complex polynomials
LPCC	Autocorrelation computation Levinson-Durbin recursion Conversion of LPC's to LPCC's

Chapter 8

CONCLUSIONS AND FUTURE WORK

8.1 Discussion About Feature Extraction

This thesis presented a *learning-by-doing* project for the author. The main contribution of this thesis is that it summarizes, or at least tries to summarize, several different viewpoints to speaker recognition into one package. We have made an extensive literature review of the most commonly used (and some not so commonly used) feature extraction methods in speaker recognition. Understanding *what the features measure* necessary requires understanding the way the speech spectrum is composed from the interaction of various articulators articulatory movements.

The term *feature extraction* is a widely used expression, although it is somewhat misleading in the context of speech processing. As pointed out by Picone [125], it somehow implies that we *know* what we are looking for in the signal. Let us compare speaker (or speech) recognition to fingerprint recognition [99], since the term *voiceprint* [69] remains to stay so popular. The history of fingerprints starts from the 17th century (1684), when first scientific studies about the ridge, furrow, and pore structures were reported by Nehemiah Grew [99]. In current state-of-the-art fingerprint recognition systems, it is actually quite well known what features should be measured from the fingerprints[1]. On the other hand, the sound spectrograph that enabled for the first time systematic study of the acoustic features, wasn't invented until the year 1946 [67]! In this sense, the fingerprint features have been studied almost 300 years longer than speech features. The author does

[1] This is the author's general image, and a fingerprint expert might have a different opinion.

not believe that the human speech production organs would be any less individual than fingerprints. The voice production system is a very complex time-varying system, but after all, the acoustic speech signal is a reflection of the dimensions of the articulators. With the current knowledge, we do not know *what* to measure and *how* to measure.

8.2 Experiments

Based on our experiments, we can make a one general conclusion. Our results indicate that in addition to the smooth spectral shape, a significant amount of speaker information is included in the *spectral details*, as opposed to speech recognition where the smooth spectral shape plays more important role. This was indicated indirectly by the following observations:

1. The number of subbands should be high (around 30-50 for the corpora used in this thesis)

2. Increasing the number of cepstral coefficients does not degrade recognition. The number of coefficients should be clearly higher than for speech recognition (at least 15-20, depending on the type of the cepstrum).

3. The LPC analysis order should be clearly higher than in speech recognition. The results did not degrade even for order 30.

4. For the LPC-derived formants, the results did not degrade even for 15 formants.

5. The differentiator method outperformed the linear regression method in delta feature computation, indicating that the fast spectral changes are more important than the smoothly varying spectral features.

This observation is consistent with intuition. In speech recognition, the vocal tract parameters that are mostly coded in the spectral envelope, are the information one wants to extract. However, for speaker recognition we want to take advantage of all of the parts of the speech production organs, especially including the voice source.

Regarding the different feature sets that we evaluated, we conclude that the differences between the best candidates are after all pretty small. By adjusting the parameters correctly, we can reach the error rate of 0.00 % for the most promising features on both corpora (filterbanks, FFT-cepstrum, LPCC, ARCSIN, LSF), given that there is enough training material and that the length of the test sequence is long enough. For a few of the experiments, we made the test segment shorter in order to see better differences between the different feature sets and the effect of adjustable parameters. If we use the full test sequence and correctly selected parameters, we can reach the zero error rate easily with any of the features listed above. Therefore, the method of choice depends on other factors like the ease of implementation and the time consumption.

Regarding both the filterbank and FFT-cepstrum, we recommend to use the cube root compression instead of the logarithm in all cases. Since these two methods give essentially the same performance, we recommend to use the cepstrum, since it is needs less features. For the LPC-based features, we recommend the ARCSIN feature set due to its simple implementation and virtually as good performance as the other LPC-based features. However, the LSF and LPCC features give also very good results, and selection of the method seems to depend also on the implementer's personal preference.

8.3 Future Work

8.3.1 Short-Term Goals

A few important issues had to be left out from this thesis due to lack of time and space. Originally, we planned to include testing of the features with degraded speech and mismatched acoustical conditions. We also planned to evaluate the simple feature compensation methods such as the widely used *cepstral mean subtraction* (CMS). Robustness against noise and distortion is the hottest topic in speaker recognition research currently. This is due to the growing demands to get working applications outside of laboratory environments. We consider testing of the most promising features in noise the most important future work in short term. Specifically, the simple noise compensation methods reviewed by Mammone & al. [100] should be systematically tested.

Throughout the experiments, we have used the vector quantization (VQ) based speaker modeling. The features that we have evaluated here, should be tested using other modeling techniques as well, specifically the Gaussian mixture model (GMM) [137, 136] since it is considered the state-of-the-art modeling technique. We believe that the results obtained using the VQ approach generalize to the GMM approach.

One noticeable thing is that throughout all experiments, we did not apply any weighting or normalization of the coefficients, but we used the most simple Euclidean distance measure. However, for instance the F-ratios of the subbands indicate clearly that certain subbands could be given larger/smaller weight. We have shown previously that subband weighting based on F-ratios improves recognition accuracy [71]. We either did not apply weighting of the cepstral coefficients (termed *cepstral liftering*) which might improve accuracy in some cases [167].

8.3.2 Long-Term Goals

Because of lack of time, we did not consider the correlations between the feature sets. If the different feature sets provide uncorrelated information, they can be combined in order to give a joint decision [73]. For instance, some feature set might discriminate good on average but might be poor for a certain speaker. For this pathological speaker, a different feature set that is in general non-discriminative, might be good. As an analogue, consider a person's hair color as a feature for person authentication. While on average this feature might not be very discriminative, for a neon-yellow haired person this feature is very discriminative (assuming artificial hair coloring is not possible).

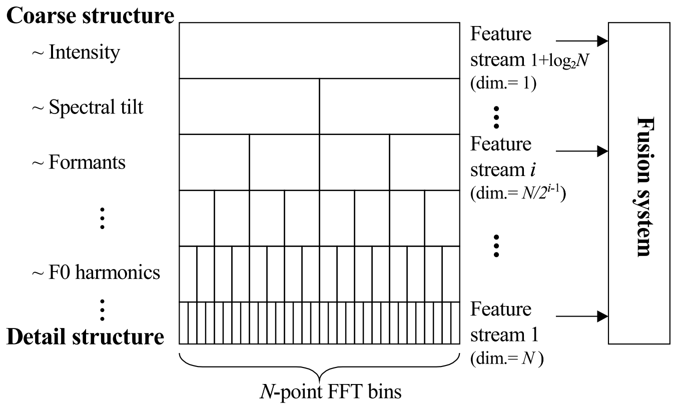

Figure 8.1: Idea of the multiresolution filterbank.

In long term, combination of several supplementary feature sets should be used, including spectral features as well as phonetical, prosodic, and lexical features. This includes several research topics:

- What feature sets to use?
- What modeling technique to use for each feature set?
- How to combine the cues from different feature sets?

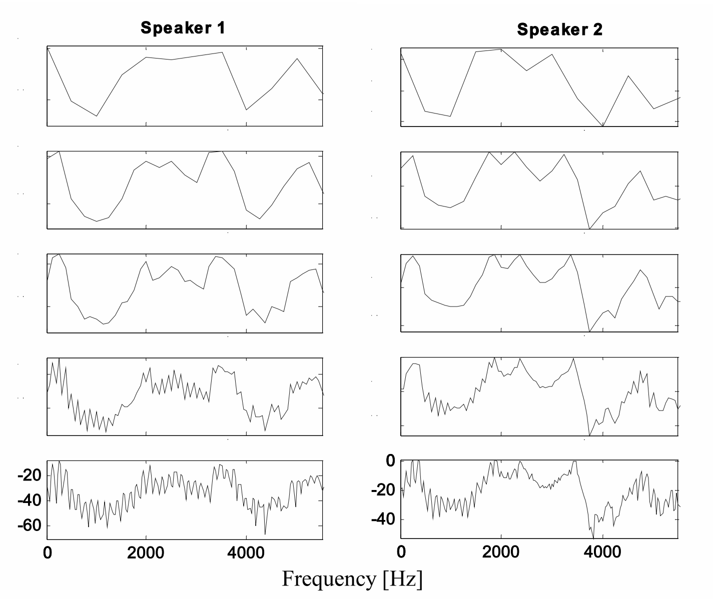

Figure 8.2: Example of multiresolution spectra of [i:] in the word "kiivasta" spoken by two males.

Story Will Continue

In order to end this thesis into an exciting point that requires a sequel, we sketch the idea of a new spectral feature set. The idea is illustrated in Fig. 8.1. Since it seems that the spectrum contains speaker information in several levels of resolutions (for instance, spectral tilt and formants represent the smooth changes, whereas F0 harmonics the fast varying details), we might want to decompose the spectrum into different levels and to model each level individually. For this, we might use a similar concept than in the case of the Haar wavelet [151]. The FFT output bins would be considered as the signal that we want to decompose. Then, the next (a coarser) level is formed by averaging the adjacent FFT bins, and this process is repeated recursively. In addition to averaging, we might want to use the differences of the adjacent bins. We could use the different resolution levels directly (with cubic root compression), or we could parametrisize them, for instance applying the DCT as in the case of cepstrum. At different levels of resolution, we might want to use different parametrizations. For the most coarse levels, we need a smaller number of cepstral coefficients, or coefficients with different indices than for the more detailed levels.

Figure 8.2 shows an example of the proposed idea applied to vowel [i:] in the word "kiivasta" spoken by two low-pitched male speakers. The first 5 decomposition levels are shown. We can see that the speakers differ in all resolution levels. The differences are smallest on the third level, which probably represents most of the phonetic information, i.e. the information that the vowel is [i:].

A classifier fusion approach that we have presented in [73] could be then applied so that the different levels of resolutions are considered as different feature sets. The combination weights of the individual feature sets can be estimated for instance from the individual classification error rates of the different resolution levels. Furthermore, it would be very interesting to see what levels are most robust against noise, voice disguise, mimicry, and other error sources. We believe that the multiresolution spectrum approach we have sketched here is potentially a promising approach for speaker recognition, since it is both intuitive, very simple to implement, and the control over different information in the spectrum is easy. This is the end of this thesis, but the story will continue in other occasions.

Bibliography

[1] Adami, A., and Hermansky, H. Segmentation of speech for speaker and language recognition conditions. In *Proc. 8th European Conference on Speech Communication and Technology (Eurospeech 2003)* (Geneva, Switzerland, 2003), pp. 841–844.

[2] Adami, A., Mihaescu, R., Reynolds, D., and Godfrey, J. Modeling prosodic dynamics for speaker recognition. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2003)* (Hong Kong, 2003), pp. 788–791.

[3] Ariyaeeinia, A., and Sivakumaran, P. Effectiveness of orthogonal instantaneous and transitional feature parameters for speaker verification. In *Proc. IEEE Int. Conf. on Security Technology* (1995), pp. 79–84.

[4] Ashour, G., and Gath, I. Characterization of speech during imitation. In *Proc. 6th European Conference on Speech Communication and Technology (Eurospeech 1999)* (Budapest, Hungary, 1999), pp. 1187–1190.

[5] Atal, B. Automatic speaker recognition based on pitch contours. *Journal of the Acoustic Society of America 52*, 6 (1972), 1687–1697.

[6] Atal, B. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *Journal of the Acoustic Society of America 55*, 6 (1974), 1304–1312.

[7] Atal, B. Efficient coding of lpc parameters by temporal decomposition. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1983)* (1993), pp. 81–84.

[8] Auckenthaler, R., Carey, M., and Lloyd-Thomas, H. Score normalization for text-independent speaker verification systems. *Digital Signal Processing 10* (2000), 42–54.

[9] Bartkova, K., D.L.Gac, Charlet, D., and Jouvet, D. Prosodic parameter for speaker identification. In *Proc. Int. Conf. on Spoken Language Processing (ICSLP 2002)* (Denver, Colorado, USA, 2002), pp. 1197–1200.

[10] Besacier, L., Bonastre, J., and Fredouille, C. Localization and selection of speaker-specific information with statistical modeling. *Speech Communications 31* (2000), 89–106.

[11] Besacier, L., and Bonastre, J.-F. Subband architecture for automatic speaker recognition. *Signal Processing 80* (2000), 1245–1259.

[12] Besacier, L., Grassi, S., Dufaux, A., Ansorge, M., and Pellandini, F. GSM speech coding and speaker recognition. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2000)* (Istanbul, Turkey, 2000), pp. 1085–1088.

[13] Bimbot, F., Blomberg, M., Boves, L., Genoud, D., Hutter, H.-P., Jaboulet, C., Koolwaaij, J., Lindberg, J., and Pierrot, J.-B. An overview of the CAVE project research activities in speaker verification. *Speech Communications 31* (2000), 155–180.

[14] Bimbot, F., Magrin-Chagnolleau, I., and Mathan, L. Second-order statistical measures for text-independent speaker identification. *Speech Communications 17* (1995), 177–192.

[15] the Biometric Consortium. WWW page, December 2003. `http://www.biometrics.org/`.

[16] Bishop, C. *Neural Networks for Pattern Recognition.* Oxford University Press, New York, 1996.

[17] Bonastre, J.-F., Bimbot, F., Boë, L.-J., Campbell, J., Reynolds, D., and Magrin-Chagnolleau, I. Person authentication by voice: a need for caution. In *Proc. 8th European Conference on Speech Communication and Technology (Eurospeech 2003)* (Geneva, Switzerland, 2003), pp. 33–36.

[18] Borden, G., and Harris, K. *Speech Science Primer. Physiology, Acoustics, and Perception of Speech*, second ed. Williams & Wilkins, Baltimore, 1984.

[19] Brunelli, R., and Falavigna, D. Person identification using multiple cues. *IEEE Trans. on Pattern Analysis and Machine Intelligence 17*, 10 (1995), 955–966.

[20] Campbell, J. Speaker recognition: a tutorial. *Proceedings of the IEEE 85*, 9 (1997), 1437–1462.

[21] Campbell, J., Reynolds, D., and Dunn, R. Fusing high- and low-level features for speaker recognition. In *Proc. 8th European Conference on Speech Communication and Technology (Eurospeech 2003)* (Geneva, Switzerland, 2003), pp. 2665–2668.

[22] Carey, M., Parris, E., Lloyd-Thomas, H., and Bennett, S. Robust prosodic features for speaker identification. In *Proc. Int. Conf. on Spoken Language Processing (ICSLP 1996)* (Philadelphia, Pennsylvania, USA, 1996), pp. 1800–1803.

[23] Clark, J., and Yallop, C. *Introduction to Phonetics and Phonology.* Basil Blackwell, Wiltshire, 1990.

[24] Damper, R., and Higgins, J. Improving speaker identification in noise by subband processing and decision fusion. *Pattern Recognition Letters 24* (2003), 2167–2173.

[25] Davis, S., and Mermelstein, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. Acoustics, Speech, and Signal Processing 28*, 4 (1980), 357–366.

[26] Dempster, A., Laird, N., and Rubin, D. Maximum likelihood from incomplete data. *Journal of Royal Statistical Society 39* (1977), 1–38.

[27] Dersch, D., and King, R. Speaker models designed from complete data sets: a new approach to text-independent speaker verification. In *Proc. 5th European Conference on Speech Communication and Technology (Eurospeech 1997)* (Rhodos, Greece, 1997), pp. 2323–2326.

[28] Doddington, G. Speaker recognition - identifying people by their voices. *Proceedings of the IEEE 73*, 11 (1985), 1651–1164.

[29] Doddington, G. Speaker recognition based on idiolectal differences between speakers. In *Proc. 7th European Conference on Speech Communication and Technology (Eurospeech 2001)* (Aalborg, Denmark, 2001), pp. 2521–2524.

[30] Doddington, G., Liggett, W., Martin, A., Przybocki, M., and Reynolds, D. Sheeps, goats, lambs and wolves: a statistical analysis of speaker performance in the nist 1998 speaker recognition evaluation. In *Proc. Int. Conf. on Spoken Language Processing (ICSLP 1998)* (Sydney, Australia, 1998).

[31] Duda, R., Hart, P., and Stork, D. *Pattern Classification*, second ed. Wiley Interscience, New York, 2000.

[32] Eatock, J., and Mason, J. A quantitative assesment of the relative speaker discriminating properties of phonemes. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1994)* (Adelaide, Australia, 1994), pp. 133–136.

[33] Eskelinen-Rönkä, P. Raportti automaattisen Puhujan Tunnistaja - tietokantaohjelman testauksesta. MSc Thesis,Department of General Phonetics, University of Helsinki, Helsinki, Finland, 1997.

[34] Eskelinen-Rönkä, P., and Niemi-Laitinen, T. Testing voice quality parameters in speaker recognition. In *Proc. The 14th Int. Congress on Phonetic Sciences (ICPhS 1999)* (San Francisco, California, USA, 1999), pp. 149–152.

[35] Fan, N., and Rosca, J. Enhanced VQ-based algorithms for speech independent speaker identification. In *Proc. Audio- and Video-Based Biometric Authentication (AVBPA 2003)* (Guildford, UK, 2003), pp. 470–477.

[36] Fant, G. *Acoustic Theory of Speech Production.* The Hague, Mouton, 1960.

[37] Farrell, K., Mammone, R., and Assaleh, K. Speaker recognition using neural networks and conventional classifiers. *IEEE Trans. on Speech and Audio Processing 2*, 1 (1994), 194–205.

[38] Faundez, M. On the model size selection for speaker identification. In *Proc. Speaker Odyssey: the Speaker Recognition Workshop (Odyssey 2001)* (Crete, Greece, 2001), pp. 189–193.

[39] Ferrer, L., Bratt, H., Gadde, V., Kajarekar, S., Shriberg, E., Sönmez, K., Stolcke, A., and Venkataraman, A. Modeling duration patterns for speaker recognition. In *Proc. 8th European Conference on Speech Communication and Technology (Eurospeech 2003)* (Geneva, Switzerland, 2003), pp. 2017–2020.

[40] Fränti, P., and Kivijärvi, J. Randomized local search algorithm for the clustering problem. *Pattern Analysis and Applications 3*, 4 (2000), 358–369.

[41] Fukunaga, K. *Introduction to Statistical Pattern Recognition*, second ed. Academic Press, London, 1990.

[42] Furui, S. Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech and Signal Processing 29*, 2 (1981), 254–272.

[43] Furui, S. Recent advances in speaker recognition. *Pattern Recognition Letters 18*, 9 (1997), 859–872.

[44] Furui, S. *Digital Speech Processing, Synthesis, and Recognition*, second ed. Marcel Dekker, Inc., New York, 2001.

[45] Genoud, D., and Chollet, G. Speech pre-processing against intentional imposture in speaker recognition. In *Proc. Int. Conf. on Spoken Language Processing (ICSLP 1998)* (Sydney, Australia, 1998).

[46] Gersho, A., and Gray, R. *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, Boston, 1991.

[47] Gish, H., and Schmidt, M. Text-independent speaker identification. *IEEE Signal Processing Magazine 11* (1994), 18–32.

[48] Gopalan, K., Anderson, T., and Cupples, E. A comparison of speaker identification results using features based on cepstrum and Fourier-Bessel expansion. *IEEE Trans. on Speech and Audio Processing 7*, 3 (1999), 289–294.

[49] Gravier, G., Mokbel, C., and Chollet, G. Model dependent spectral representations for speaker recognition. In *Proc. 5th European Conference on Speech Communication and Technology (Eurospeech 1997)* (Rhodos, Greece, 1997), pp. 2299–2302.

[50] Hardt, D., and Fellbaum, K. Spectral subtraction and RASTA-filtering in text-dependent HMM-based speaker verification. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1997)* (Munich, Germany, 1997), pp. 867–870.

[51] Harrington, J., and Cassidy, S. *Techniques in Speech Acoustics.* Kluwer Academic Publishers, Dordrecht, 1999.

[52] Harris, F. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE 66*, 1 (1978), 51–84.

[53] Haykin, S. *Neural Networks: a Comprehensive Foundation*, second ed. Prentice-Hall, Upper Saddle River, 1999.

[54] He, J., Liu, L., and Palm, G. A discriminative training algorithm for VQ-based speaker identification. *IEEE Trans. on Speech and Audio Processing 7*, 3 (1999), 353–356.

[55] Hermansky, H. Perceptual linear prediction (PLP) analysis for speech. *Journal of the Acoustic Society of America 87* (1990), 1738–1752.

[56] Hermansky, H. RASTA processing of speech. *IEEE Trans. on Speech and Audio Processing 2*, 4 (1994), 578–589.

[57] Higgins, A., Bahler, L., and Porter, J. Voice identification using nearest-neighbor distance measure. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1993)* (Minneapolis, Minnesota, USA, 1993), pp. 375–378.

[58] Huang, X., Acero, A., and Hon, H.-W. *Spoken Language Processing: a Guide to Theory, Algorithm, and System Development.* Prentice-Hall, New Jersey, 2001.

[59] Huggins, M., and Grieco, J. Confidence metrics for speaker identification. In *Proc. Int. Conf. on Spoken Language Processing (ICSLP 2002)* (Denver, Colorado, USA, 2002), pp. 1381–1384.

[60] Hume, J. Wavelet-like regression features in the cepstral domain for speaker recognition. In *Proc. 5th European Conference on Speech Communication and Technology (Eurospeech 1997)* (Rhodos, Greece, 1997), pp. 2339–2342.

[61] Hyvärinen, A., Karhunen, J., and Oja, E. *Independent Component Analysis.* John Wiley & Sons, Inc., New York, 2001.

[62] Ifeachor, E., and Lewis, B. *Digital Signal Processing - a Practical Approach*, second ed. Pearson Education Limited, Edinburgh Gate, 2002.

[63] Iivonen, A., Harinen, K., Keinänen, L., Kirjavainen, J., Meister, E., and Tuuri, L. Development of a multiparametric speaker profile for speaker recognition. In *Proc. The 15th Int. Congress on Phonetic Sciences (ICPhS 2003)* (Barcelona, Spain, 2003), pp. 695–698.

[64] Jain, A., R.P.W.Duin, and J.Mao. Statistical pattern recognition: a review. *IEEE Trans. on Pattern Analysis and Machine Intelligence 22* (2000), 4–37.

[65] Jain, A., and Zongker, D. Feature selection: evaluation, application, and small sample performance. *IEEE Trans. on Pattern Analysis and Machine Intelligence 19* (1997), 153–158.

[66] Jang, G.-J., Lee, T.-W., and Oh, Y.-H. Learning statistically efficient features for speaker recognition. *Neurocomputing 49* (2002), 329–348.

[67] Jr., J. D., Hansen, J., and Proakis, J. *Discrete-Time Processing of Speech Signals*, second ed. IEEE Press, New York, 2000.

[68] Kajarekar, S., and Hermansky, H. Speaker verification based on broad phonetic categories. In *Proc. Speaker Odyssey: the Speaker Recognition Workshop (Odyssey 2001)* (Crete, Greece, 2001), pp. 201–206.

[69] Kersta, L. Voiceprint identification. *Nature 5*, 196 (1962), 1253–1257.

[70] Kersholt, J., Jansen, E., van Amelsvoort, A., and Broeders, A. Earwitness line-ups: effects of speech duration, retention interval and acoustic environment on identification accuracy. In *Proc. 8th European Conference on Speech Communication and Technology (Eurospeech 2003)* (Geneva, Switzerland, 2003), pp. 709–712.

[71] Kinnunen, T. Designing a speaker-discriminative filter bank for speaker recognition. In *Proc. Int. Conf. on Spoken Language Processing (ICSLP 2002)* (Denver, Colorado, USA, 2002), pp. 2325–2328.

[72] Kinnunen, T., and Fränti, P. Speaker discriminative weighting method for VQ-based speaker identification. In *Proc. Audio- and Video-Based Biometric Authentication (AVBPA 2001)* (Halmstad, Sweden, 2001), pp. 150–156.

[73] Kinnunen, T., Hautamäki, V., and Fränti, P. On the fusion of dissimilarity-based classifiers for speaker identification. In *Proc. 8th European Conference on Speech Communication and Technology (Eurospeech 2003)* (Geneva, Switzerland, 2003), pp. 2641–2644.

[74] Kinnunen, T., Kilpeläinen, T., and Fränti, P. Comparison of clustering algorithms in speaker identification. In *Proc. IASTED Int. Conf. Signal Processing and Communications (SPC 2000)* (Marbella, Spain, 2000), pp. 222–227.

[75] Kinnunen, T., and Kärkkäinen, I. Class-discriminative weighted distortion measure for VQ-based speaker identification. In *Proc. Joint IAPR International Workshop on Statistical Pattern Recognition (S+SPR2002)* (Windsor, Canada, 2002), pp. 681–688.

[76] Kinnunen, T., Kärkkäinen, I., and Fränti, P. Is speech data clustered? - statistical analysis of cepstral features. In *Proc. 7th European Conference on Speech Communication and Technology (Eurospeech 2001)* (Aalborg, Denmark, 2001), pp. 2627–2630.

[77] Kittler, J., Hatef, M., Duin, R., and Matas, J. On combining classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence 20*, 3 (1998), 226–239.

[78] Kittler, J., and Nixon, M., Eds. *4th Internation Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA 2003).* Lecture Notes in Computer Science. Springer-Verlag, Berlin, 2003.

[79] Kolano, G., and Regel-Brietzmann, P. Combination of vector quantization and gaussian mixture models for speaker verification. In *Proc. 6th European Conference on Speech Communication and Technology (Eurospeech 1999)* (Budapest, Hungary, 1999), pp. 1203–1206.

[80] Konig, Y., Heck, L., Weintraub, M., and Sonmez, K. Nonlinear discriminant feature extraction for robust text-independent speaker recognition. In *Proc. RLA2C-ESCA Speaker Recognition and its Commercial and Forensic Applications* (1998), pp. 72–75.

[81] Kuhn, R., Junqua, J.-C., Nguyen, P., and Niedzielski, N. Rapid speaker adaptation in eigenvoice space. *IEEE Trans. on Speech and Audio Processing 8* (2000), 695–707.

[82] Kwon, S., and Narayanan, S. Speaker change detection using a new weighted distance measure. In *Proc. Int. Conf. on Spoken Language Processing (ICSLP 2002)* (Denver, Colorado, USA, 2002), pp. 2537–2540.

[83] Kyung, Y., and Lee, H.-S. Text independent speaker recognition using micro-prosody. In *Proc. Int. Conf. on Spoken Language Processing (ICSLP 1998)* (Sydney, Australia, 1998).

[84] Lapidot, I., Guterman, H., and Cohen, A. Unsupervised speaker recognition based on competition between self-organizing maps. *IEEE Transactions on Neural Networks 13* (2002), 877–887.

[85] Laver, J. *Principles of Phonetics.* Cambridge University Press, Cambridge, 1994.

[86] Li, Q., Juang, B.-H., and Lee, C.-H. Automatic verbal information verification for user authentication. *IEEE Trans. on Speech and Audio Processing 8* (2000), 585–596.

[87] Li, X., Mak, M., and Kung, S. Robust speaker verification over the telephone by feature recuperation. In *Proc. 2001 Int. Symposium on Intelligent Multimedia, Video, and Speech Processing* (Hong Kong, 2001), pp. 433–436.

[88] Lieberman, P., and Blumstein, S. *Speech Physiology, Speech Perception, and Acoustic Phonetics.* Cambridge University Press, Cambridge, 1988.

[89] Linde, Y., Buzo, A., and Gray, R. An algorithm for vector quantizer design. *IEEE Transactions on Communications 28*, 1 (1980), 84–95.

[90] Linguistic data consortium. WWW page, December 2003. `http://www.ldc.upenn.edu/`.

[91] Liu, C.-S., Huang, C.-S., Lin, M.-T., and Wang, H.-C. Automatic speaker recognition based upon various distances of LSP frequencies. In *Proc. 25th Annual 1991 IEEE International Carnahan Conference on Security Technology* (1991), pp. 104–109.

[92] Liu, C.-S., Wang, H.-C., and Lee, C.-H. Speaker verification using normalized log-likelihood score. *IEEE Trans. on Speech and Audio Processing 4*, 1 (1996), 56–60.

[93] Liu, C.-S., Wang, W.-J., Lin, M.-T., and Wang, H.-C. Study of line spectrum pair frequencies for speaker recognition. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1990)* (Albuquerque, New Mexico, USA, 1990), pp. 277–280.

[94] Liu, D., and Kubala, F. Fast speaker change detection for broadcast news transcription and indexing. In *Proc. 6th European Conference on Speech Communication and Technology (Eurospeech 1999)* (Budapest, Hungary, 1999), pp. 1031–1034.

[95] Liu, L., He, J., and Palm, G. A comparison of human and machine in speaker recognition. In *Proc. 5th European Conference on Speech Communication and Technology (Eurospeech 1997)* (Rhodos, Greece, 1997), pp. 2327–2330.

[96] Majewski, W., and Mazur-Majewska, G. Speech signal parametrization for speaker recognition under voice disguise conditions. In *Proc. 6th European Conference on Speech Communication and Technology (Eurospeech 1999)* (Budapest, Hungary, 1999), pp. 1227–1230.

[97] Makhoul, J. Linear prediction: a tutorial review. *Proceedings of the IEEE 64*, 4 (1975), 561–580.

[98] Mallat, S. *A Wavelet Tour of Signal Processing.* Academic Press, New York, 1999.

[99] Maltoni, D., Jain, A., Maio, D., and Prabhakar, S. *Handbook of Fingerprint Recognition.* Springer Verlag, New York, 2003.

[100] Mammone, R., Zhang, X., and Ramachandran, R. Robust speaker recognition: a feature based approach. *IEEE Signal Processing Magazine 13*, 5 (1996), 58–71.

[101] Markel, J., Oshika, B., and A.H. Gray, j. Long-term feature averaging for speaker recognition. *IEEE Trans. Acoustics, Speech, and Signal Processing 25*, 4 (1977), 330–337.

[102] Martin, A., and Przybocki, M. Speaker recognition in a multi-speaker environment. In *Proc. 7th European Conference on Speech Communication and Technology (Eurospeech 2001)* (Aalborg, Denmark, 2001), pp. 787–790.

[103] Mason, J., and Zhang, X. Velocity and acceleration features in speaker recognition. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1991)* (Toronto, Canada, 1991), pp. 3673–3676.

[104] Matsui, T., and Furui, S. A text-independent speaker recognition method robust against utterance variations. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1991)* (Toronto, Canada, 1991), pp. 377–380.

[105] Ming, J., Stewart, D., Hanna, P., Corr, P., Smith, J., and Vaseghi, S. Robust speaker identification using posterior union models. In *Proc. 8th European Conference on Speech Communication and Technology (Eurospeech 2003)* (Geneva, Switzerland, 2003), pp. 2645–2648.

[106] Miyajima, C., Watanabe, H., Kitamura, T., and Katagiri, S. Speaker recognition based on discriminative feature extraction - optimization of mel-cepstral features using second-order all-pass warping function. In *Proc. 6th European Conference on Speech Communication and Technology (Eurospeech 1999)* (Budapest, Hungary, 1999), pp. 779–782.

[107] Miyajima, C., Watanabe, H., Tokuda, K., Kitamura, T., and Katagiri, S. A new approach to designing a feature extractor in speaker identification based on discriminative feature extraction. *Speech Communications 35* (2001), 203–218.

[108] Mokhtari, P. *An Acoustic-Phonetic and Articulatory Study of Speech-Speaker Dichotomy.* PhD thesis, School of Computer Science, University of New South Wales, Canberra, Australia, 1998.

[109] Nguyen, P., Akagi, M., and Ho, T. Temporal decomposition: a promising approach to VQ-based speaker identification. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2003)* (Hong Kong, 2003).

[110] Niemi-Laitinen, T. *Puhujantunnistus rikostutkinnassa.* Licentiate's thesis, University of Helsinki, Department of Phonetics, Helsinki, Finland, 1999.

[111] Niemi-Laitinen, T. Personal communication, 2003.

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

[113] Openshaw, J., Sun, Z., and Mason, J. A comparison of composite features under degraded speech in speaker recognition. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1993)* (Minneapolis, Minnesota, USA, 1993), pp. 27–30.

[114] Oppenheim, A., and Schafer, R. *Digital Signal Processing.* Prentice Hall, Englewood Cliffs, 1975.

[115] Orman, D., and Arslan, L. Frequency analysis of speaker identification. In *Proc. Speaker Odyssey: the Speaker Recognition Workshop (Odyssey 2001)* (Crete, Greece, 2001), pp. 219–222.

[116] Ortega-García, J., and González-Rodríguez, J. Overview of speech enhancement techniques for automatic speaker recognition. In *Proc. Int. Conf. on Spoken Language Processing (ICSLP 1996)* (Philadelphia, Pennsylvania, USA, 1996), pp. 929–932.

[117] Paliwal, K., and Alsteris, L. Usefulness of phase spectrum in human speech perception. In *Proc. 8th European Conference on Speech Communication and Technology (Eurospeech 2003)* (Geneva, Switzerland, 2003), pp. 2117–2120.

[118] Pelecanos, J., Myers, S., Sridharan, S., and Chandran, V. Vector quantization based gaussian mixture modeling for speaker verification. In *Proc. Int. Conf. on Pattern Recognition (ICPR 2000)* (Barcelona, Spain, 2000), pp. 3298–3301.

[119] Peskin, B., Navratil, J., Abramson, J., Jones, D., Klusacek, D., Reynolds, D., and Xiang, B. Using prosodic and conversational features for high-performance speaker recognition: report from JHU WS'02. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2003)* (Hong Kong, 2003), pp. 792–795.

[120] Petry, A., and Barone, D. Speaker identification using nonlinear dynamic features. *Chaos, Solitons and Fractals 13* (2001), 221–231.

[121] Petry, A., and Barone, D. Text-dependent speaker verification using Lyapunov exponents. In *Proc. Int. Conf. on Spoken Language Processing (ICSLP 2002)* (Denver, Colorado, USA, 2002), pp. 1321–1324.

[122] Phan, F., and Micheli-Tzanakou, E. *Supervised and unsupervised pattern recognition: feature extraction and computational intelligence.* CRC Press, Boca Raton, 2000, ch. Speaker Identification Through Wavelet Multiresolution Decomposition and ALOPEX, pp. 301–315.

[123] Phythian, M., Ingram, J., and Sridharan, S. Effects of speech coding on text-dependent speaker recognition. In *Proceedings of IEEE Region 10 Annual Conference on Speech and Image Technologies for Computing and Telecommunications (TENCON'97)* (1997), pp. 137–140.

[124] Pickles, J. *An Introduction to the Physiology of Hearing.* Academic Press, London, 1982.

[125] Picone, J. Signal modeling techniques in speech recognition. *Proceedings of the IEEE 81*, 9 (1993), 1215–1247.

[126] Plumpe, M., Quatieri, T., and Reynolds, D. Modeling of the glottal flow derivative waveform with application to speaker identification. *IEEE Trans. on Speech and Audio Processing 7*, 5 (1999), 569–586.

[127] Praat: doing phonetics by computer. www page, December 2003. `http://www.praat.org/`.

[128] Prabhakar, S., Pankanti, S., and Jain, A. Biometric recognition: security and privacy concerns. *IEEE Security & Privacy Magazine 1* (2003), 33–42.

[129] Proakis, J., and Manolakis, D. *Digital Signal Prosessing. Principles, Algorithms and Applications*, second ed. Macmillan Publishing Company, New York, 1992.

[130] Quatieri, T., Reynolds, D., and O'Leary, G. Estimation of handset nonlinearity with application to speaker recognition. *IEEE Trans. on Speech and Audio Processing 8*, 5 (2000), 567–584.

[131] Rabiner, L., and Juang, B.-H. *Fundamentals of Speech Recognition.* Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[132] Ramachandran, R., Farrell, K., Ramachandran, R., and Mammone, R. Speaker recognition - general classifier approaches and data fusion methods. *Pattern Recognition 35* (2002), 2801–2821.

[133] Reynolds, D. Experimental evaluation of features for robust speaker identification. *IEEE Trans. on Speech and Audio Processing 2* (1994), 639–643.

[134] Reynolds, D. An overview of automatic speaker recognition technology. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2002)* (Orlando, Florida, USA, 2002), pp. 4072–4075.

[135] Reynolds, D., Andrews, W., Campbell, J., Navratil, J., Peskin, B., Adami, A., Jin, Q., Klusacek, D., Abramson, J., Mihaescu, R., Godfrey, J., Jones, D., and Xiang, B. The SuperSID project: exploiting high-level information for high-accuracy speaker recognition. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2003)* (Hong Kong, 2003), pp. 784–787.

[136] Reynolds, D., Quatieri, T., and Dunn, R. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing 10*, 1 (2000), 19–41.

[137] Reynolds, D., and Rose, R. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Trans. on Speech and Audio Processing 3* (1995), 72–83.

[138] Rodríguez-Liñares, L., García-Mateo, C., and Alba-Castro, J. On combining classifiers for speaker authentication. *Pattern Recognition 36* (2003), 347–359.

[139] Rose, P. *Forensic Speaker Identification.* Taylor & Francis, London, 2002.

[140] Rosenberg, A. Automatic speaker verification: a review. *Proceedings of the IEEE 64*, 4 (1976), 475–487.

[141] Rosenberg, A., and Sambur, M. New techniques for automatic speaker verification. *IEEE Trans. Acoustics, Speech, and Signal Processing 23*, 2 (1975), 169–176.

[142] Sambur, M. Selection of acoustic features for speaker identification. *IEEE Trans. Acoustics, Speech, and Signal Processing 23*, 2 (1975), 176–182.

[143] Schmidt-Nielsen, A., and Crystal, T. Speaker verification by human listeners: experiments comparing human and machine performance using the nist 1998 speaker evaluation data. *Digital Signal Processing 10* (2000), 249–266.

[144] Singh, G., Panda, A., Bhattacharyya, S., and Srikanthan, T. Vector quantization techniques for GMM based speaker verification. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2003)* (Hong Kong, 2003).

[145] Sivakumaran, P., Ariyaeeinia, A., and Loomes, M. Sub-band based text-dependent speaker verification. *Speech Communications 41* (2003), 485–509.

[146] Smith, S. *The Scientist and Engineer's Guide to Digital Signal Processing.* California Technical Publishing, `http://www.dspguide.com`, 1997.

[147] Sönmez, M., Heck, L., Weintraub, M., and Shriberg, E. A lognormal tied mixture model of pitch for prosody-based speaker recognition. In *Proc. 5th European Conference on Speech Communication and Technology (Eurospeech 1997)* (Rhodos, Greece, 1997), pp. 1391–1394.

[148] Soong, F., A.E., A. R., Juang, B.-H., and Rabiner, L. A vector quantization approach to speaker recognition. *AT & T Technical Journal 66* (1987), 14–26.

[149] Soong, F., and Rosenberg, A. On the use of instantaneous and transitional spectral information in speaker recognition. *IEEE Trans. on Acoustics, Speech and Signal Processing 36*, 6 (1988), 871–879.

[150] Story, B. An overview of the physiology, physics and modeling of the sound source for vowels. *Acoustical Science and Technology 23*, 4 (2002), 195–206.

[151] Strang, G., and Nguyen, T. *Wavelets and filter banks.* Wellesley-Cambridge Press, Wellesley, 1996.

[152] Sullivan, K., and Pelecanos, J. Revisiting Carl Bildt's impostor: would a speaker verification system foil him? In *Proc. Audio- and Video-Based Biometric Authentication (AVBPA 2001)* (Halmstad, Sweden, 2001), pp. 144–149.

[153] Toh, K.-A. Fingerprint and speaker verification decisions fusion. In *Proc. 12th Int. Conf. on Image Analysis and Processing (ICIAP'03)* (2003), pp. 626–631.

[154] Torres, H., and Rufiner, H. Automatic speaker identification by means of mel cepstrum, wavelets and wavelets packets. In *Proc. 22nd Annual EMBS International Conference (IEEE Engineering in Medicine and Biology Society)* (2000), pp. 978–981.

[155] van Leeuwen, D. Speaker verification systems and security considerations. In *Proc. 8th European Conference on Speech Communication and Technology (Eurospeech 2003)* (Geneva, Switzerland, 2003), pp. 1661–1664.

[156] Vuuren, S. Comparison of text-independent speaker recognition methods on telephone speech with acoustic mismatch. In *Proc. Int. Conf. on Spoken Language Processing (ICSLP 1996)* (Philadelphia, Pennsylvania, USA, 1996), pp. 1788–1791.

[157] Vuuren, S., and Hermansky, H. On the importance of components of the modulation spectrum for speaker verification. In *Proc. Int. Conf. on Spoken Language Processing (ICSLP 1998)* (Sydney, Australia, 1998), pp. 3205–3208.

[158] Wang, R.-H., He, L.-S., and Fujisaki, H. A weighted distance measure based on the fine structure of feature space: application to speaker recognition. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1990)* (Albuquerque, New Mexico, USA, 1990), pp. 273–276.

[159] Watkins, D. *Fundamentals of Matrix Computations*, second ed. Wiley-Interscience, Wellesley, 2002.

[160] Weber, F., Manganaro, L., Peskin, B., and Shriberg, E. Using prosodic and lexical information for speaker identification. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2002)* (Orlando, Florida, USA, 2002), pp. 141–144.

[161] Wolf, J. Efficient acoustic parameters for speaker recognition. *Journal of the Acoustic Society of America 51*, 6 (Part 2) (1972), 2044–2056.

[162] Woo, S., Lim, C., and Osman, R. Development of a speaker recognition system using wavelets and artifical neural networks. In *Proc. 2001 Int. Symposium on Intelligent Multimedia, Video and Speech Processing* (2001), pp. 413–416.

[163] Xiang, B. Text-independent speaker verification with dynamic trajectory model. *IEEE Signal Processing Letters 10* (2003), 141–143.

[164] Xu, L., Oglesby, J., and Mason, J. The optimization of percuptually-based features for speaker identification. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1989)* (Glasgow, Scotland, 1989), pp. 520–523.

[165] Yoshida, K., Takagi, K., and Ozeki, K. Speaker identification using subband HMMs. In *Proc. 6th European Conference on Speech Communication and Technology (Eurospeech 1999)* (Budapest, Hungary, 1999), pp. 1019–1022.

[166] Zetterholm, E. The significance of phonetics in voice imitation. In *Proc. 8th Australian Int. Conf. on Speech Science and Technology* (2000), pp. 342–347.

[167] Zhen, B., Wu, X., Liu, Z., and Chi, H. On the use of bandpass liftering in speaker recognition. In *Proc. Int. Conf. on Spoken Language Processing (ICSLP 2000)* (Beijing, China, 2000), pp. 933–936.

[168] Zhu, Q., and Alwan, A. On the use of variable frame rate in speech recognition. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2000)* (Istanbul, Turkey, 2000), vol. 3, pp. 1783–1786.

[169] Zilca, R., and Bistritz, Y. Speaker identification using LSP codebook models and linear discriminant functions. In *Proc. 6th European Conference on Speech Communication and Technology (Eurospeech 1999)* (Budapest, Hungary, 1999), pp. 799–802.

[170] Zilovic, M., Ramachandran, R., and Mammone, R. Speaker identification based on the use of robust cepstral features obtained from pole-zero transfer functions. *IEEE Trans. on Speech and Audio Processing 6*, 3 (1998), 260–267.

[171] Zwicker, E., and Terhardt, E. Analytical expressions for critical band rate and critical bandwidth as a function of frequency. *Journal of the Acoustic Society of America 68* (1980), 1523–1525.

The correct answer to “voiceprint identification” of Fig. 1.2: spectrograms B and E belong to the same speaker.