Institutionen för systemteknik

Department of Electrical Engineering

Examensarbete

Speaker Recognition: Current State and Experiment

Master thesis performed in *Information Coding*
by Pol Lari Jarque

Report number
Linköping June 2011

TEKNISKA HÖGSKOLAN
LINKÖPINGS UNIVERSITET

Department of Electrical Engineering
Linköping University
S-581 83 Linköping, Sweden

Linköpings tekniska högskola
Institutionen för systemteknik
581 83 Linköping

Speaker Recognition: Current State and Experiment

Master thesis in Information Coding

at Linköping Institute of Technology
by Pol Lari Jarque

....
LiTH-ISY/ERASMUS-A--11/001--SE

Supervisor: Viiveke Fåk

Examiner: Viiveke Fåk
Linköping June 2011

Presentation Date June 9, 2011 **Publishing Date (Electronic version)**	**Department and Division** Department of Electrical Engineering Information Coding	Linköpings universitet

Language X English Other (specify below)	**Type of Publication** **X** Master's Thesis Licentiate thesis Degree thesis Thesis C-level	**ISBN (Licentiate thesis)** **ISRN:** **Title of series (Licentiate thesis)**

URL, Electronic Version
http://www.ep.liu.se

Publication Title
Speaker Recognition: Current State and Experiment

Author
Pol Lari Jarque

Abstract
In this thesis the operation of the speaker recognition systems is described and the state of the art of the main working blocks is studied. All the research papers looked through can be found in the References.

As voice is unique to the individual, it has emerged as a viable authentication method. There are several problems that should be considered as the presence of noise in the environment and changes in the voice of the speakers due to sickness for example. These systems combine knowledge from signal processing for the feature extraction part and signal modeling for the classification and decision part.

There are several techniques for the feature extraction and the pattern matching blocks, so it is quite tricky to establish a unique and optimum solution. MFCC and DTW are the most common techniques for each block, respectively. They are discussed in this document, with a special emphasis on their drawbacks, that motivate new techniques which are also presented here.

A search through the Internet is done in order to find commercial working implementations, which are quite rare, then a basic introduction to Praat is presented. Finally, some intra-speaker and inter-speaker tests are done using this software.

Number of pages: 53

Keywords

Speaker Recognition, Praat, MFCC, DTW, Wavelet Packets.

Abstract

In this thesis the operation of the speaker recognition systems is described and the state of the art of the main working blocks is studied. All the research papers looked through can be found in the References.

As voice is unique to the individual, it has emerged as a viable authentication method. There are several problems that should be considered as the presence of noise in the environment and changes in the voice of the speakers due to sickness for example. These systems combine knowledge from signal processing for the feature extraction part and signal modeling for the classification and decision part.

There are several techniques for the feature extraction and the pattern matching blocks, so it is quite tricky to establish a unique and optimum solution. MFCC and DTW are the most common techniques for each block, respectively. They are discussed in this document, with a special emphasis on their drawbacks, that motivate new techniques which are also presented here.

A search through the Internet is done in order to find commercial working implementations, which are quite rare, then a basic introduction to Praat is presented. Finally, some intra-speaker and inter-speaker tests are done using this software.

Acknowledgements

When I began my Telecommunications degree at Universitat Politècnica de Catalunya (UPC), in Barcelona, I never had thought that one day I would carry out and present my Final Project abroad, at that time I did not use to travel so much yet.

One year ago, I had the opportunity to apply for a position in order to develop my Project in a foreign University, and then I was sure that I wanted to come to Sweden. A country, where I could “survive” with my English and which has always been mentioned for its good things in the South European countries.

Chance made that the University of destination were Linköpings Universitet, by that time I did not know anything about this city nor its University. But now, I can state that it has been a great opportunity.

I would like to express my gratitude to Viiveke, my supervisor at the Department, who has always been actively helpful with all my issues around this thesis and also in the development of this report, giving a valuable feedback. Thanks for providing me the opportunity to write the thesis with the Information Coding Division.

I would like to hereby express that this thesis has been carried out in several different places, at the working room kindly offered by the Department, on planes, buses, cars, several libraries and at my home in Vilanova during a short visit that I did in April.

I would like to thank all the people that I have been able to meet during my stay in Sweden, with whom I have shared lots of good moments and we have learned aspects of each other’s own cultures.

A special thanks to Rubén, who also is my opponent, and Sergi from UPC, for the moments we have shared in Sweden.

Moreover, I would like to thank the people from Vilanova i la Geltrú, my hometown, who have given me a significant support and encouragement, and they have also been following my “adventures” during these months living in Sweden.

And finally, I would especially like to thank my family, who from Vilanova has given to me all the facilities to push forward this thesis. Particularly to my mother, for all the help during my whole education.

For all these reasons, I would like to dedicate my thesis to all of you, hoping you will enjoy it.

“El meu país és tan petit, que quan el Sol se’n va a dormir, mai no està prou segur d’haver-lo vist”

Lluís Llach, catalan songwriter.

My country is so small, that when the Sun goes to sleep, it is never sure if it has seen it.

Table of contents

List of figures

List of tables

1. INTRODUCTION

Voice has emerged as a viable authentication method, because just like a fingerprint or iris, voice is unique to the individual.

A speaker's voice is extremely difficult to forge for biometric comparison purposes, since too many qualities are measured ranging from spectral magnitudes to pitch. The vibration of a user's vocal chords and the patterns created by the physical components resulting in human speech are as distinctive as fingerprints.

Attempts to impersonate a voice or provide voice recordings to gain fraudulent authentication fail due to the distinctive details of the voiceprint used for comparison. While voice impersonations may sound like an exact match to the human ear, detailed mathematical analysis of the print tends to reveal vast differences. Likewise, voice recordings that sound like an exact match to the human ear most often reveal distortions caused in the recording process when measured for biometric authentication purposes.

1.1. Overview

Speech processing is a diverse field with many applications; Figure 1 shows a few of these areas and how speaker recognition relates to the rest of the field. [1]

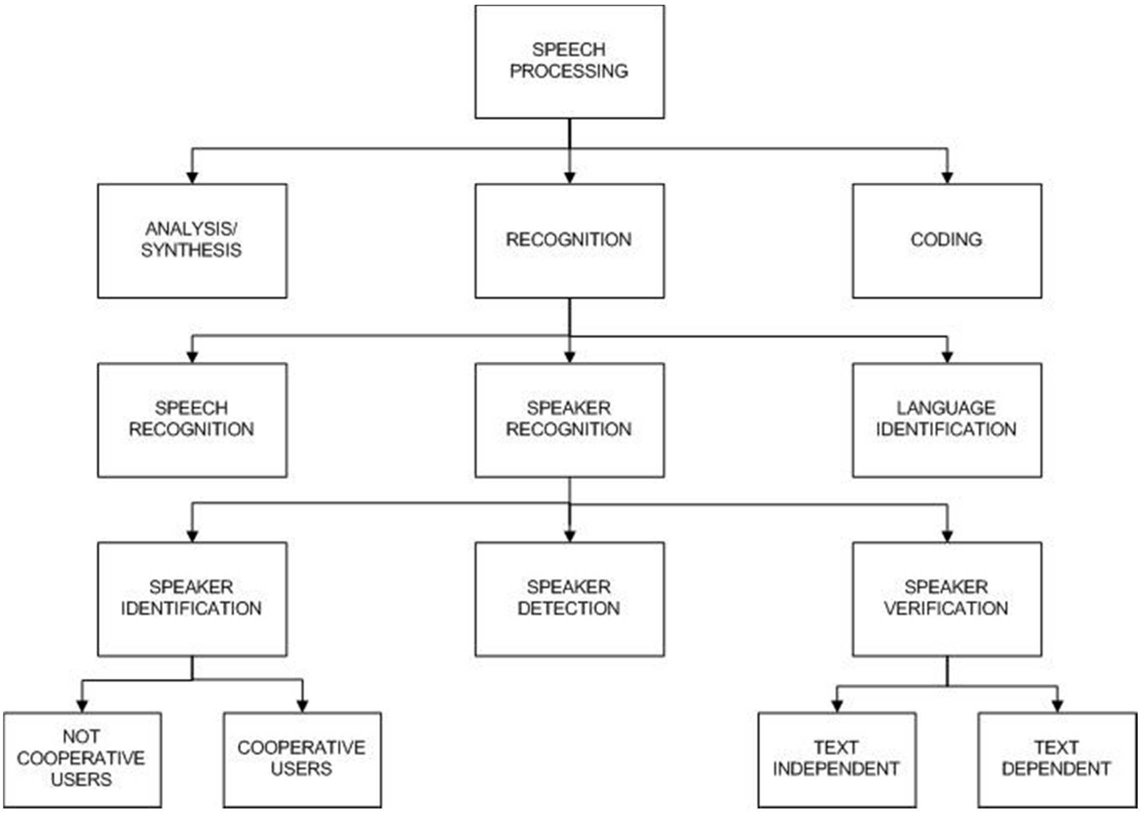

Figure 1: Speech processing fields.

1.2. Problem and Purpose

This master thesis project is an academic study suggested by the Information Coding Division of the Electrical Engineering Department of Linköping University. The aim of this project is to carry out a study of the current state of the art of Speaker Recognition Systems.

A company interest is created in order to give a suitable frame and scope for the study; it is assumed that they want to implement an Automatic Speaker Verification System for its own service. They pretend to control worker's access with this system, as this is a natural (voice cannot be forgotten or misplaced) and economical solution (the cost mainly is for software).

Then, a whole theoretical study of the state of the art is done, but only methods widely used and which can be implemented with small resources are of interest, as a possible solution for the imagined company.

There is also the intention of looking for some real implementations in order to compare its features and if it is possible to test its performance. All this work should be done during approximately 4 months.

So, the first step will be to know the exactly definition of these systems and their main features and then, study the general blocks diagram of this system.

1.3. Speaker Recognition System

Speaker or voice recognition is a biometric modality that uses the human voice for recognition purposes. The speaker recognition process relies on features influenced by both the physical structure of an individual's vocal tract and the behavioral characteristics of the individual.

We should distinguish clearly between two operation modes: *Speaker Identification* and *Speaker Verification*. The first one is the task of determining who is talking from a set of known voices or speakers. As the unknown person makes no identity claim the system must perform an 1:N classification.

Generally, it is assumed that the unknown voice comes from a fixed set of known speakers, thus the task is often referred as a closed-set identification.

Speaker Verification (also called authentication or detection) is the task of determining whether the person is who he/she claims to be (by entering an employee number or a word, which is known to the system), then an accept/rejection decision is taken.

Generally, it is assumed that the imposters are not known to the system, this is referred as an open-set task.

Depending on the grade of speaker's cooperation there are two modalities:

- *Text-dependent*; the individual presents either a fixed word or prompted phrase that is programmed into the system and can improve performance especially with cooperative users. Fixed words require higher resistance against recordings while prompted random words make analysis for forgeries simpler.
- *Text-independent*; these systems do not have previous knowledge of the presenter's phrasing and are more flexible in situations where the individual submitting the sample may be unaware of the collection or unwilling to cooperate, which presents a more difficult challenge.

There are two main features (error probabilities) that characterize the performance of these systems:

- *False acceptance of an invalid user (FA) or incorrect impostor acceptance.* It takes a pair of subjects to make a false acceptance error: an impostor and a target. These errors are the ultimate concern of high-security speaker-verification applications. However, they can be traded off for false rejection errors.
- *False rejection of a valid user (FR) or incorrect customer rejection*. This fact of course denies users of their rights and prevents them from performing duties.

At a high level, all speaker recognition systems contain two main modules: feature extraction and feature matching.

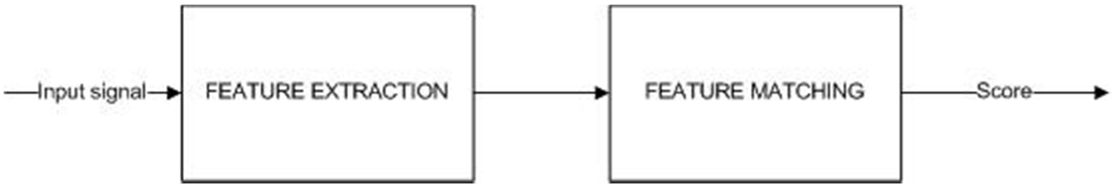

Figure 2: General Block diagram for Speaker Recognition Systems.

1.4. Background

Speaker verification has co-evolved with the technologies of speech recognition and speech synthesis because of the similar characteristics and challenges associated with each.

Original speaker recognition systems used the average output of the several analog filters to perform matching, often with the aid of humans. In 1976, Texas Instruments built a prototype system that was tested by the U.S. Air Force. In the mid 1980s, the NIST developed the NIST Speech Group. Among those who have researched and designed several generations of speaker recognition systems are AT&T; ITT; Massachusetts Institute of Technology Lincoln Labs; Nippon Telegraph and Telephone. The majority of ASV research is directed at verification over telephone lines. [1]

Last years' progress has focused on the understanding on how the distributions of training data are best represented and on what generalizations should be made.

So, there is potential in order to find the best signal representation for the acoustic speech signals that best retain the needed information for the recognition process, especially in noisy environments while suppressing irrelevant information. The general trend shows accuracy improvements over time with larger tests (enabled by larger data bases).

Nowadays, there is considerable speaker recognition activity in industry, national laboratories and universities. Some of the companies who are working in Speaker Verification Systems are Microsoft, Loquendo, Recognition Technologies, Google, Nokia and Apple.

Figure 3 shows a sampling of the chronological advancement in speaker verification. The following terms are used to define the columns in the figure: *"Org"* is the company or school where the work was done, *"Year"* when it was done, *"Features"* are the signal measurements, *"Input"* is the type of input speech, *"Text"* indicates whether a text-dependent or text-independent mode of operation is used, *"Method"* is the heart of the pattern-matching process, *"Pop"* is the population size of the test, and *"Error"* is the equal error percentage for speaker-verification systems *"V"* or the Recognition error percentage for speaker identification systems *"I"* given the specified duration of the test speech in seconds.

Org	Year	Features	Method	Input	Text	Pop	Error
AT&T	1974	Cepstrum	Pattern Match	Lab	Dependent	10	I: 2% -0.5 s V: 2% -1s
AT&T	1981	LP	Pattern Match	Telephone	Dependent	10	V: 0.2% -3s
ITT	1983	LP Cepstrum	Pattern Match	Lab	Independent	11	I: 21% -3 s I: 4% -10 s
TI	1985	Filter-Bank	DTW	Lab	Dependent	200	V: 0.8% -6s
AT&T	1985	LP	VQ	Telephone	10 isolated digits	100	I: 5% -1.5 s I:1.5% -3.5s
ITT	1986	Cepstrum	DTW	Lab	Independent	11	V:10%-2.5 s I:4.5% -10s
ITT	1991	LP-Cepstrum	DTW	Office	Dependent	186	V: 1.7%-10s
AT&T	1991	LP	HMM	Telephone	10 isolated digitis	100	V: 2.8% -1.5 s V:0.8% -3.5s
MIT-LL	1995	Mel-Cepstrum	HMM(GMM)	Office	Dependent	138	I:0.8%-10s V:0.12%-10s
MIT-LL	1996	Mel-cepstrum Mel delta-cepstrum	HMM(GMM)	Telephone	Independent	416	V:11%-3s V:6%-10s V:3%-30s

Figure 3: Selected chronology of speaker verification progress.

The results in error probabilities that we can obtain with a Speaker Verification System will depend on which techniques we choose on each block. It should be noted that is difficult to make meaningful comparisons between the text-dependent and the generally more complex text-independent tasks. In order to make the best selection we should know about the scenario where the system has to work, this involves discussing about noise, as the system will never be used in clean conditions.

1.5. Methodology

The first part of the work consists of a study of research papers published mainly in the last decade, selected from different authors in order to get different points of view of the state of the art. It is not possible to set only one solution for the problem, as there are many working environments. After this, some statistical parameters are used in order to study the performance of the software found.

1.6. Structure

Chapter one is an introduction of the thesis work where we can find an overview, the purpose of the thesis, a brief definition and approach of Speaker Verification Systems. Chapter two deals with the working scenario, there the working conditions and some blocks of the system are discussed. In chapter three, several feature extraction techniques are presented, as this system block is the best suitable for a signal processing approach. In chapter four, some practical implementations found are presented. In chapter five, some tests using Praat software are shown. In chapter six, the thesis conclusions are shown, and some points of view for a future work are presented. At the end, there is also an appendix, with some information about the voiceprints used in the tests, a glossary and the references consulted during the project.

2. THE WORKING SCENARIO

In this chapter we will discuss the working conditions of the system; some characteristics of the speech signal, the sources of verification error, mainly the noise and the pattern matching and decision blocks.

2.1. The input signal: some characteristics of speech production and modeling

There are two main sources of speaker specific production characteristics of speech: Physical and Learned. [1]

Vocal tract shape is an important physical distinguishing factor of speech. The vocal tract is generally considered as the speech production organs above the vocal cords. This includes the following:

Laryngeal pharynx, Oral pharynx, Oral cavity, Nasal pharynx and Nasal Cavity

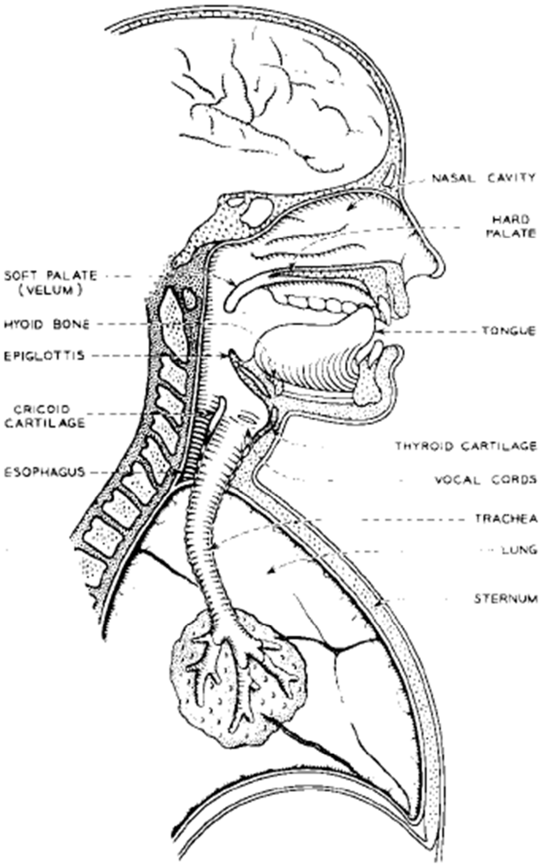

Figure 4: Human vocal system, extracted from "Speech Analysis and Perception".

An adult male vocal tract is approximately 17 cm long. The vocal folds are stretched between the thyroid cartilage and the arytenoids cartilages. The area between the vocal folds is called the glottis.

As the acoustic wave passes through the vocal tract its frequency content (spectrum) is altered by the resonances of the vocal tract. Vocal tract resonances are called formants. Thus, the vocal tract shape can be estimated from the spectral shape of the voice signal.

Voice verification systems typically use features derived only from the vocal tract. The human vocal mechanism is driven by an excitation source, which also contains speaker-dependent information. The excitation is generated by airflow from the lungs, carried by the trachea through the vocal folds. The excitation can be characterized as *phonation, whispering, frication, compression, vibration* or a combination of these.

Phonation occurs when air flow is modulated by the vocal folds, another name used to call the vocal cords. When the vocal folds are closed, pressure builds up underneath them until they blow apart. Then the folds are drawn back together again by their tension, elasticity and the Bernoulli Effect. This pulsed air stream, arising from the oscillating vocal folds, excites the vocal tract. The frequency of oscillation is called the fundamental frequency and it depends on the length, tension and mass of the vocal folds. Thus, fundamental frequency is another distinguishing characteristic that is physically based.

Whispered excitation is produced by airflow rushing through a small triangular opening between the arytenoids cartilages at the rear of the nearly closed vocal folds, this result in turbulent airflow, which has a wide-band noise characteristic. The arytenoids cartilages are two in number and they are of a pyramidal shape and situated at the upper border of the cricoids cartilage at the back of the larynx.

Frication excitation is produced by constrictions in the vocal tract. The place, shape and degree of constriction determine the shape of the broad-band noise excitation. As the constriction moves forward, the spectral concentration generally increases in frequency. Sounds generated by frication are called *fricatives* or *sibilants*. Examples of them may be "f" and "z" in English.

Compression excitation results from releasing a completely closed and pressurized vocal tract. This results in silence (during pressure accumulation) followed by a short noise burst. If the release is sudden, a stop or *plosive* sound is generated, the sounds spelled by "p" and "b" found in several languages might be examples of them. If these sounds are at the beginning of a word, it is possible to see clearly the generation process. If the release is gradual, an *affricate* is formed, sounds spelled with "ch" and "j" in English are clear examples of them.

Vibration excitation is caused by air being forced through a closure other than the vocal folds, especially at the tongue, as for example in trilled "r".

Speech produced by phonated excitation is called *voiced*, speech produced by phonated excitation plus frication is called *mixed voiced* and speech produced by other types of excitation is called *unvoiced*.

For these differences in the manner of production, it is reasonable to expect some speech models to be more accurate for certain classes of excitation than others.

Other physiological speaker-dependent properties include the thoracic area (which plays a role in the resonance properties of the vocal system), vital capacity (maximum volume of air one can blow out after maximum intake), maximum phonation time (the maximum duration a syllable can be sustained) and glottal air flow (amount of air going through vocal folds).

Due to the nature and properties of the input signal, there are some human factors which contribute to the verification error, these are *aging* (the vocal tract can drift away from models with age, *sickness* (colds can alter the vocal tract), extreme *emotional states* (for example stress or duress) and misread or misspoken prompted phrases. All of them may be sources of error of the own signal and generally are outside the scope of the algorithms.

2.2. Noisy environments

Verification systems are not used in ideal acoustic conditions and some environmental noise is often mixed with the speech signal. Therefore, it is necessary to distinguish the types of noise that we have to deal with. [2]

- *Continuous noise*, which has a stationary property, thus it is simple to define a quantitative model.
- *Sudden Short-time noise*, which appears discontinuously and exists only for a short time. It has the dynamic property that the beginning of the noise has large amplitude and the succeeding part decays with time. The quantitative model of this type of noise is difficult to obtain. Impulsive noise would be a clear example of that.

We also should consider other external annoying effects as the presence of other speakers near the system, poor or inconsistent room acoustics (presence of multipath and reverberation) and the channel mismatch (for example using different microphones for enrollment and verification or time varying microphone placement). All of them will contribute to increase the misrecognition rate, but they can be taken into account in algorithms.

Once we have reached this point, it is necessary to know which strategies can be followed to deal with noise problems. It is possible to discern two main different ways to approach the problem:

- *Signal Processing*, where we try to clean the input signal.
- *Input Signal*, where we try to get a clean input signal.

2.2.1. Signal Processing

There are two different ways to proceed:

- **Signal Filtering Enhancement**

It is possible to use generic noise reduction techniques to enhance the quality of the original time-domain signal prior to the feature extraction. The most interesting approach is how to deal with impact noise. There are some attempts using filters that separate the non stationary part from the stationary part of the input signal. [2]

This proposed system is composed of a nonlinear digital filter called stationary-non-stationary separating filter (S-NS) and a time-frequency masking. The former detects and separates the beginning part of the impact noise; the latter is for reducing the reverberation and it is only applied for the noisy part detected by the filter. It is supposed that we know which kind of the impact noise and the typical duration is known beforehand.

The time-frequency masking is realized using a voice model and a noise model which are designed from typical human voice and typical noise characteristics which correspond to that contained in the input. The clean input part obtained by S-NS can also be used to create the voice model in addition to the voice data base.

Using signal enhancement becomes an additional step in the entire recognition process, so the computational load is increased.

- **Signal Transform Noise Robustness**

More and more new noise-robust characteristic features have been developed. The most common representation is MFC coefficients, but there are also others as PLP, Wavelets that try to improve noise robustness. They will be discussed in detail in the next chapter.

2.2.2. Input Signal

An interesting example of these techniques is the use of a throat microphone which captures the body-conducted signal (uncorrupted by background noise). [3]

These microphones have a good noise rejection and a high accuracy on speaker activity detection even in very noisy conditions. They rely on the fact that the human voice is not only transmitted through the air, it is propagated through bones and tissues as well, as long as interfering signals are not in contact with persons.

So, it is possible to take benefit of this fact in order to obtain an input signal with a high SNR for our ASV system.

This way of obtaining the signal limits the signal bandwidth, so high frequencies will be attenuated (where unvoiced sounds are found) and only the low-pass speech signal will be retained (voiced sounds).

For this reason it would be unsuitable for applications where broadband signals are needed, such as recognizers. Using a narrowband speech at the input, there will be matching discrepancies unless we train the recognizer with a large database of body-conducted data.

Using this method, the system complexity is increased because of the extensive training of the recognizer. It is not a good solution either because having these microphones working as standard equipment is uncomfortable for the users, needs technical support and also they may be stolen and damaged. In consequence, it is not a useful solution for our purpose.

2.3. Pattern Matching

The main idea behind this part of the system is to calculate a match score, which is a measure of the similarity of the input feature vectors to some model. To enroll users into the system, a model of the voice, based on the extracted features, is generated and stored. Finally, to authenticate the user, the matching algorithm compares the incoming speech signal with the model of the claimed user. [1]

There are two types of models:

- *Template model;* the pattern matching is deterministic.
- *Stochastic models*; the pattern matching is probabilistic.

2.3.1. Template models

The observation is assumed to be an imperfect replica of the template, and the alignment of observed frames to template frames is selected to minimize a distance measure.

This method and its corresponding distance measure is perhaps the most intuitive method. A simplest template model consists of a single template **x**, which is the model for a frame of speech. The match score between the template **x** for the claimed speaker and an input feature vector x_i from the unknown user is given by the distance d (x_i, **x**). The model for the claimed speaker could be the centroid (mean) of a set of N training vectors.

$$\boldsymbol{x} = 1/N \sum_{i=1}^{N} Xi$$

Many different distance measures between the vectors x_i and **x** can be expressed as:

d (x_i, **x**)= (x_i - **x**)^T **W** (x_i - **x**); where **W** is a weighting matrix. If it is an identity matrix, the distance is Euclidean.

The models can be dependent of time, as in *DTW*, or independent of time, as in *VQ* modeling. Template models dominated early work in text-dependent speaker recognition.

The most common used method is DTW. Suppose we have the two time series **X** and **Y** of length *m* and *n*, respectively:

X= X_1,X_2,…X_i...X_m

Y=Y_1,Y_2,…Y_j…Y_n

To align the sequences an *n* x *m* matrix is constructed. In general *n* is not equal to *m* because of timing inconsistencies in human speech. The (i^{th},j^{th}) element of the matrix contains the distance d (X_i,Y_j) between the two points X_i, Y_j. Typically the Euclidean distance d (X_i,Y_j) = (X_i - Y_j)^2 is used. Each matrix element (*i,j*) corresponds to the alignment between the points X_i,Y_j.

The warping path **W**, is a contiguous set of the matrix elements that define a mapping between X and Y. The k^{th} element of W is defined as w_k=$(i,j)_k$, where we have:

W=w_1,w_2,…,w_k; where max(m,n)<K<m+n-1

The warping path is typically subject to several constraints, so there are many warping paths which satisfy the imposed conditions, but the most interesting ones are those which minimize the warping cost, as the following example:

$$DTW(\boldsymbol{X},\boldsymbol{Y}) = \min\left(\frac{\sqrt{\sum_{k=1}^{K} Wk}}{K}\right)$$

If the warp signals were identical, the warp path (see: Figure 6) would be a diagonal line and the warping would have no effect.

The algorithm tries to explain variability in the Y-axis by warping the X-axis. This can lead to unintuitive alignments where a single point on one time series warps onto a large subsection of another time series. These drawbacks are called singularities.

Another problem is that the method may fail to find obvious or natural alignments in two sequences simply because a feature (i.e. valley, peak, inflection point) in one sequence is slightly higher or lower than its corresponding feature in the other sequence.

There is an improved technique called DDTW [4], which tries to solve both these problems. In DDTW the distance measure d(X_i,Y_j) is not Euclidean but rather the square of the difference of the estimated first derivatives of X_i and Y_j. The method for estimating derivatives is not fixed by DDTW.

With that approach if two data points X_i, Y_j have the same value, but X_i is part of a rising trend and Y_j is part of a falling trend, they will not be mapped ideally as in DTW. It is possible to prevent this problem because DDTW also considers the higher level feature of shape, not only the Y-values of the data points as DTW.

Empirically, the time complexity in DDTW is the same as in DTW. For this reason it should be taken into account for an hypothetical implementation.

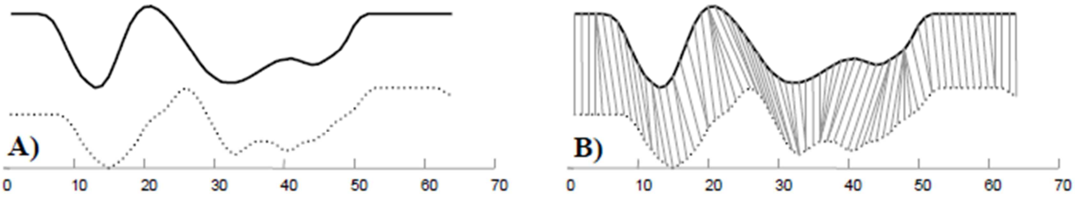

Figure 5: An example of Dynamic Time Warping, extracted from "Derivative Dynamic Time Warping". In A) there are two sequences, it should be noted that while the sequences have an overall similar shape, they are not aligned in the time axis. In B) may be seen the alignment found by DTW between the two sequences that allows a more sophisticated distance measure to be calculated.

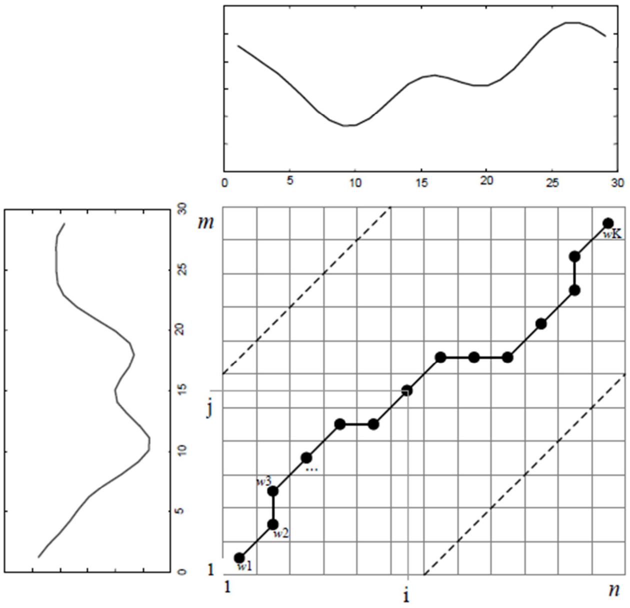

Figure 6: An example of a warping path, extracted from "Derivative Dynamic Time Warping".

In *VQ Source Modeling*, a VQ Codebook is designed by standard clustering procedures for each enrolled speaker using his training data. The pattern match score is the distance between an input vector and the minimum distance code word in the VQ codebook.

There is another method, called *Nearest Neighbors* (NN) which combines the strengths of the DTW and VQ. It does not cluster the enrollment training data to form a compact codebook. Instead, it keeps all the training data and can, therefore, use temporal information.

2.3.2. Stochastic models

Using these models results in a measure of the likelihood or conditional probability of the observation given the model. The observation is a random vector with a conditional PDF that depends upon the speaker. The conditional PDF for the claimed speaker can be estimated from a set of training vectors and given the estimated density, the probability that the observation is generated by the claimed speaker can be determined. These models were developed later and can offer more flexibility, examples of them are HMM and GMM.

GMM is used in speaker recognition systems due to its capacity of representing a large class of sample distributions. A powerful attribute of the GMM is its ability to form smooth approximations to arbitrarily shaped densities. It acts as a hybrid between the classical *uni-modal Gaussian model* which represents feature distributions by a position and a elliptic shape and a *Vector Quantizer* (VQ) or *Nearest Neighbor model* (NN) which represent a distribution by a discrete set of characteristic templates by using a discrete set of Gaussian functions, each one with its mean and covariance matrix. [5]

HMM is also very popular for modeling sequences. In conventional Markov models each state corresponds to a deterministically observable event, and then the transitional probabilities are the only parameters. In HMM, the state is not directly visible, but output, dependent on the state, is visible. Thus, the output of such sources in any given state is not random and lacks the flexibility needed here. In the HMM, the observations are a probabilistic function of the state. It can be considered a generalization of the GMM or from another point of view a GMM can also be viewed as a single state HMM.

2.4. Classification and decision

When the match score between the input speech feature vector and a model of the claimed speaker's voice is computed, a verification decision is made whether to accept or reject the speaker or to request another utterance.

The classification procedure is a sequential hypothesis-testing problem; it involves choosing between two hypotheses: that the user is the claimed speaker or that he is not the claimed speaker (an impostor). Let us assume they are called H_0 and H_1, respectively. [1]

As shown in Figure 7, the match scores of the observations form two different PDF's (Probability Density Function) according to whether the user is the claimed speaker or an impostor.

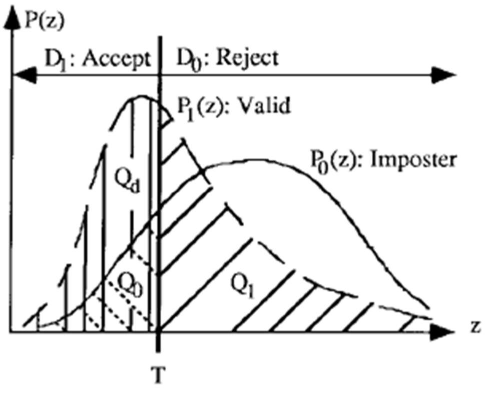

Figure 7: Valid and impostor densities, extracted from "Speaker Recognition: A tutorial".

Performance Probabilities	Decision D	Hypothesis H	Decision Result
Q_0	1	0	False Acceptance (FA)
Q_1	0	1	False Rejection (FR)
Q_d= 1- Q_1	1	1	True Acceptance
1-Q_0	0	0	True Rejection

Table 1: Probability Terms and definitions.

To find a given performance probability area, the hypothesis determines over which PDF to integrate, and the threshold determines which decision region forms the limits of integration. The probability of error, which is minimized by Bayes' decision rule, is determined by the amount of overlap in the two pdf's. The smaller the overlap between the two pdf's, the smaller the probability of error. The overlap in two Gaussian pdf's with different means and equal variance may be easily measured by the F-ratio as follows:

$$F = (\mu_1 - \mu_0)^2/\sigma^2$$

When the likelihood ratio for a speaker A, λ_A is determined, the classification problem can be stated as choosing a threshold ***T***, so that the decision rule is:

$$\lambda_A \geq \boldsymbol{T}\text{, choose } H_0$$

$$\lambda_A < \boldsymbol{T}\text{, choose } H_1$$

This threshold can be determined by:

- Setting ***T*** equal to an estimate of p_1/p_0 to approximate minimum error performance where p_i are the a priori probabilities that the user is an impostor and that the user is the true speaker, respectively.
- Choosing ***T*** to satisfy a fixed FA or FR criterion (Neyman-Pearson).
- Varying ***T*** to find different FA/FR ratios and then choosing ***T*** to give the desired FA/FR ratio.

A measure of overall system performance must specify the levels of both types of errors. The tradeoff between FA and FR is a function of the decision threshold. This is shown in the ROC curve, which plots the probability of FA versus the probability of FR (or FA rate versus FR rate).

In figure 8, an example of a hypothetical family of ROC is plotted on a log-log scale. The line of equal error probability is shown as a dotted diagonal line. The straight line in the figure indicates that the product of the probability of FA and the probability of FR is a constant for this hypothetical system, and it is equal to the square of what is referred to as the equal error rate (EER). [1]

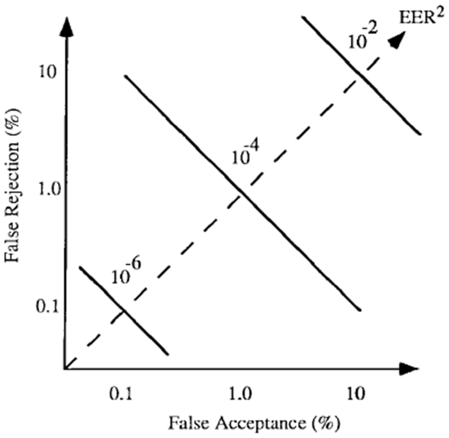

Figure 8: An example of a hypothetical ROC, extracted from "Speaker Recognition: A tutorial".

2.5. Summary

At the end of this chapter, as a conclusion it is possible to understand the following blocks diagram, which has more detail in comparison with the one showed in the introduction chapter.

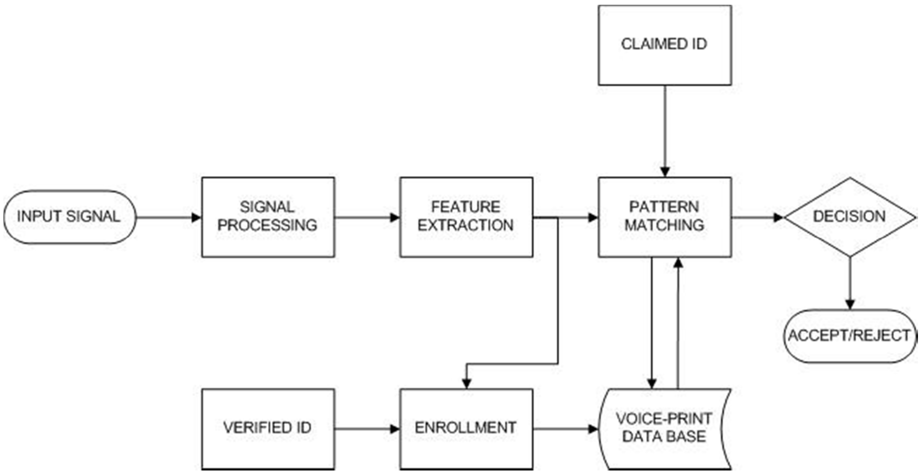

Figure 9: ASV block diagram.

3. FEATURE EXTRACTION

Speech is a complicated signal produced as a result of several transformations occurring at several different levels: semantic, linguistic, articulatory and acoustic. The feature analysis plays a crucial role in the overall performance of the system.

Speaker related differences are a result of a combination of anatomical differences inherent in the vocal tract and the learned speaking habits of different individuals. These differences should be used to discriminate between speakers. At this point, speaker recognition systems may benefit from speech recognition research in the field of speech signal representation.

Although Automatic Speaker Verification (ASV) systems are traditionally divided into three components: feature extraction and selection, pattern matching and classification, it is convenient from the perspective of designing system components, that they are not independent.

The purpose of the feature extraction module is to convert the speech waveform to some type of parametric representation. The speech signal is a slowly time varying signal, usually called quasi-stationary. When it is observed over a sufficiently short period of time (between 5 and 100 ms), its characteristics are fairly stationary. However, over long periods of time (on the order of 0.2 s or more) the signal characteristics change to reflect the different speech sounds being spoken.

Therefore, short-time spectral analysis is the most common way to characterize the speech signal. A wide range of possibilities exist for parametrically representing the speech signal for the speaker recognition task, such as LPC, MFCC and others.

The aim of this chapter is to discuss some of them, as this is the suitable module for a signal processing approach in these systems. The most used feature extraction techniques and some new approaches trying to improve some drawbacks will be discussed. Probably MFCC is the best known and most popular method, but it does not work as well for all the possible scenarios.

It is especially interesting to know how the knowledge of human hearing system and speech production may influence these techniques.

Even without the aid of grammatical, semantic and pragmatic information human listeners outperform today's best automatic speech recognizers. It is assumed that this ability is due to a superior human technique for processing sounds in general. This has led many researchers to attempt to reproduce characteristics of the human auditory system in the hope of obtaining improved systems. [8]

We could agree that speech has evolved taking account of the abilities and limitations of our hearing system. It would be surprising if there were features in the speech signal that could contribute usefully to automatic speech recognition and yet be imperceptible to humans: if they cannot be perceived it is unlikely that they would be controlled in the speech production process.

After this general overview, it is necessary to know which are the most important properties that useful feature extraction techniques should satisfy:

- High noise and distortion robustness
- High disguise and mimicry robustness
- High inter-speaker variation

- Low intra-speaker variation
- Easy to measure
- Maximally internally independent features

3.1. LPC

Linear Prediction Coding (LPC) was originally developed as an efficient method for coding of speech. Later LPC was also used for other speech related tasks such as speech recognition and speaker recognition. The appearance of MFCC reduced its relevance. LPC is based on a simple model of speech production. The vocal tract is modeled as a set of connected tubes with equal length and constant diameter. Under certain assumptions as no energy loss inside the vocal tract and nonlinear effects it can be shown that the transfer function of this model is an all-pole filter with Z-transform:

$$A(z) = \frac{1}{1 - \sum_{i=1}^{P} a_i z^{-i}}$$

Where P is the number of tube segments, and the coefficients a_i are directly related to the resonance frequencies of the vocal tract. [6]

It is possible to obtain these coefficients by minimizing the linear prediction error, taking into account that speech samples are approximated as a linear combination of past speech samples. Then, the following expression is used:

$$e(n) = s_n - \sum_{i=1}^{P} a_i s_{n-i}$$

There are different criteria for minimizing linear prediction error, usually the squared expectation value over a finite interval is chosen. It is called the autocorrelation method. Then a unique set of predictor coefficients is obtained.

3.2. MFCC

First of all, we should describe briefly what the Cepstrum representation of the signal is. The Cepstrum power was defined in 1963 by Bogert as:

$$= \left|\mathcal{F}\left\{\log(|\mathcal{F}\{f(t)\}|^2)\right\}\right|^2$$

It can be seen as information about the rate of change in the different spectrum bands. An important property of this representation is that a convolution of signals (in the time domain) becomes a sum in the Cepstral domain.
This is useful in voice identification because the low-frequency periodic excitation of the vocal cords and the formant filtering of the vocal tract; which convolve in the time domain, multiply in the frequency domain, and become additive in different regions of the Cepstral domain, then can be separated in this domain.

The difference between the Cepstrum and the Mel-frequency Cepstrum is that in MFC, frequency bands are spaced on the Mel-scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the Cepstral representation.

The Mel frequency scale is used in order to capture the phonetically important characteristics of speech. This scale has linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. See its expression:

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$$

As a result the following mapping between Hertz and Mel is achieved:

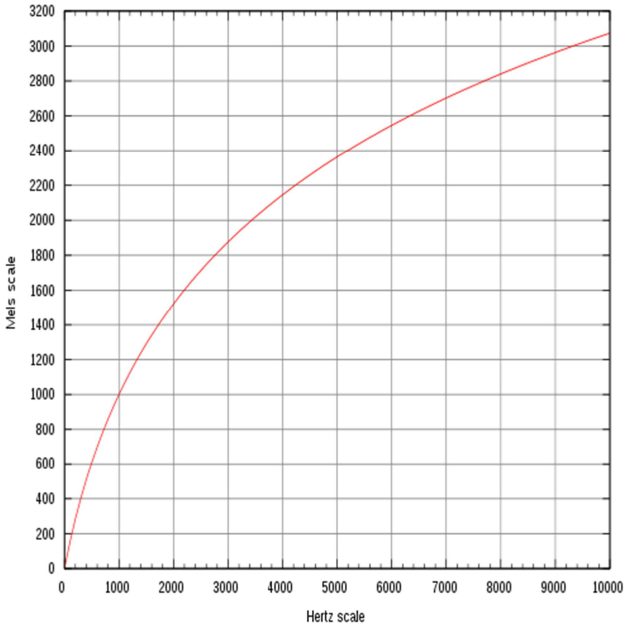

Figure 10: Plot of pitch Mel versus Hz.

So, there are five steps in order to obtain the MFC coefficients: [9]

- *Framing*; Short-time spectral analysis is the most common way to characterize the speech signal. For this purpose, it is necessary to divide the input signal in short length frames, values between 10 and 30 ms are useful to capture local spectral characteristics. Then, we may be sure that we have a stationary signal. There is a tradeoff between time and frequency resolution.
  It is usual to use frames of 20 ms of duration, with 50% overlapping.

- *Windowing*; The concept here is to minimize the spectral distortion by using the window to reduce the signal on both ends thus reducing side effects by signal discontinuity at the beginning and at the end due to framing. Each frame is windowed with a *Hanning* or *Hamming* window. The Hamming window is a "raised cosine" function optimized to minimize the maximum (nearest) side lobe, it is defined by the following formula:

$$w[n] = 0.54 - 0.46(1 - \cos(\frac{2\pi n}{N-1})); 0 \leq n \leq N-1$$

Then, the ouput after this block is the following:

$$y[n] = x[n] \cdot w[n] \text{ with } 0 \leq n \leq \text{N-1}$$

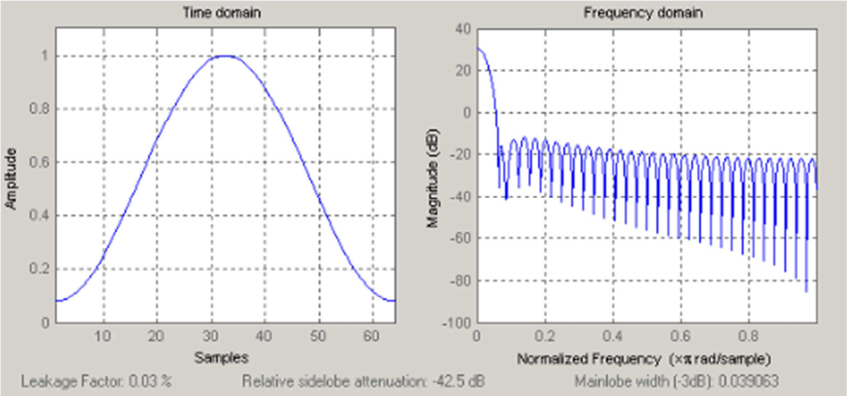

Figure 11: Representation of a Hamming Window extracted from MathWorks.

- *Take the FFT;* Once we get the framed and windowed speech signal samples, the frame of N samples is converted from the time to the frequency domain using the FFT algorithm.

  $$Y\,[k] = \sum_{n=0}^{N-1} y[n] \cdot e^{\frac{-j2\pi kn}{N}}$$

  where X [n] is the n-th sample of the input windowed frame and with k between 0 and N-1.

- *Mel-Frequency Warping;* As it was mentioned, the Mel Scale is used in order to copy the human perception of the frequency contents. The formula given above is used to compute the Mels for a given frequency *f* in Hz.

  Then, a filter bank with triangular shape and overlapping is applied. The filters are arranged linearly in the Mel frequency domain, but when we look at the filter arrangement in the ordinary frequency domain, they are arranged as shown in the following figure:

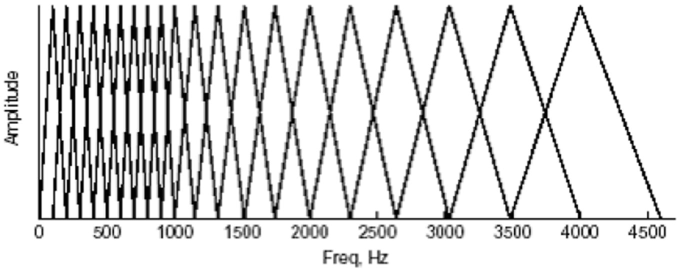

Figure 12: MFCC Filter banks extracted from "Modified Mel-Frequency Cepstral Coefficients".

For an 8 KHz sampling rate 20 filters are used over a frequency range of 0 - 4 KHz. Of the 20 filters, in the ordinary frequency range the first ten are arranged linearly in the range 0 - 1 KHz. The next 5 filters are arranged logarithmical in 1 KHz – 2 KHz range. The last 5 filters are also arranged logarithmical in the frequency range of 2 – 4 KHz.

The base of each triangular filter is defined by the center frequency of the neighboring filters. [10]
Then, the output at the end of this block is the following:

$$Z[k] = \sum_{j=1}^{N} Y[j] \cdot H\,[k-j]$$

Where $H\,[k-j]$ is the sampled response of the MFC filter bank of M filters.

- *Log compression and Discrete Cosine Transformation;* As we want the coefficients to have the speaker specific vocal tract characteristics in them, the Cepstral representation discussed before is chosen. For this purpose and in order to get the Cepstral characters in the features, it is necessary to take the logarithm of the filter bank outputs followed by the DCT to convert spectrum back to the time domain. The 20 outputs will become the Mel Frequency Cepstral Coefficients (MFCC).

  It is necessary to exclude the first coefficient that we get, because it represents the mean value of the input signal, which carries little specific information. Generally, for speaker recognition tasks the first 12 coefficients are used. Then, we obtain a set of coefficients, usually called *acoustic vector,* which may be used to represent or recognize the voice characteristics of the speaker. [10]
  Then, the mathemathical expression at the output of this block is:

$$C[n] = \sum_{k=1}^{M} \log(Z[k]) \cdot \cos\frac{\pi n}{N}(k - \frac{1}{2})$$

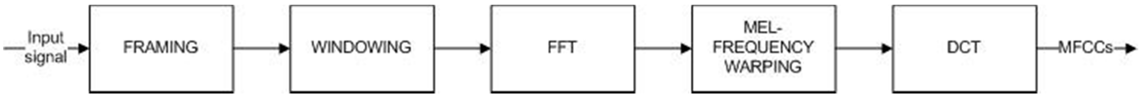

Figure 13: Block diagram of MFCC extraction technique.

Originally it was developed for speech recognition systems, but nowadays it is widely used by speaker verification systems.

As it is a logarithmic representation it degrades in presence of noise. In order to deal with that problem it is common to normalize their coefficients. It works well in clean conditions. There are several modifications that improve noise robustness, which will be discussed in what follows.

3.3. Modifications on MFCC

All of these modifications try to reduce the noise effects, as it is the weakness of the MFCC representation.

- *Cepstral Mean Normalization and Spectrum Mean Normalization*. [11]
  Named as CMN-SMN-MFCC, it is based on the general noisy speech model. This method uses SMN to suppress additive noise and CMN to suppress the effect of

convolution noise. Theoretical analysis shows that the combination of CMN and SMN can inhibit additive and convolution noise at the same time. Speaker Recognition tests performed using this technique achieve 10.5 % and 9.6 % relative improvements in comparison to conventional MFCC and Delta MFCC features, respectively.

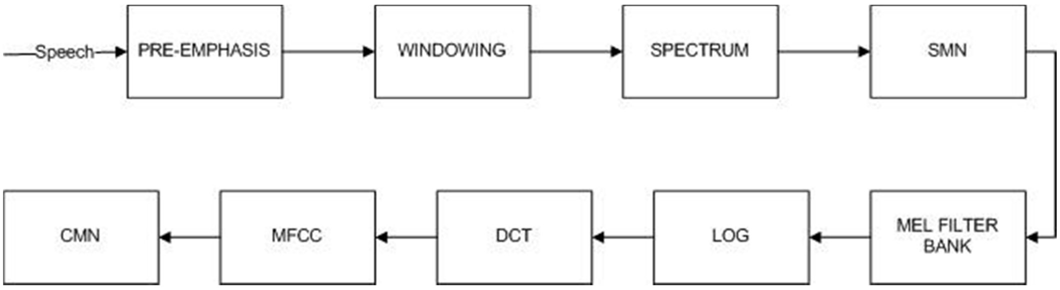

Figure 14: Extraction approach of the CMN-SMN-MFCC.

- *Additive Cepstral Distortion Model (ACDM)*. [7]
  ACDM is developed for achieving a statistical minimum mean-square error (MMSE) estimation of Mel-frequency coefficients. The estimator works entirely in the Cepstral domain, so there is no need for an inversion of the Discrete Cosine Transform (DCT).

  This proposed estimator is developed using a novel approach to modeling the interaction between speech and noise. As a result it models the noise distortion as additive in the Cepstral domain leading to a closed-form solution to the estimation problem.

  The success of the estimation algorithm depends mainly on the quality of three components: the a priori noise power estimates, the a priori speech power estimates and the Cepstral prior model.

- *Autocorrelation MFCC (AMFCC)*. [7]
  Its motivation is that higher-lag autocorrelation coefficients are less affected by noise than the original signal. This algorithm uses the full autocorrelation expression and assumes the cross terms are zero. This assumption is commonly used during the design of noise robust speech recognition feature extraction algorithms. The author investigated the validity of the assumption and demonstrated that it is fair for five different tested noises.
  The algorithm has a lot of steps in common with the MFCC algorithm and the main difference can be found in the method of estimating the speech spectrum.

3.4. Dynamic Cepstral Features

Examples such as Delta and Delta-delta Cepstral have been shown to play an essential role in capturing the transitional characteristics of the speech signal. So, they have also been introduced into the Speaker Recognition systems. They improve the recognition accuracy by adding a characterization of temporal dependencies. The n-th Delta-Cepstral feature is typically defined as:

$$\Delta C_k[n] = C_{k+m}[n] - C_{k-m}[n]$$

Where m is typically 2 or 3 and refers to the frames index, then it is possible to obtain the n-th Delta Delta-Cepstral feature:

$$\Delta^2 C_k[n] = \Delta C_{k+m}[n] - \Delta C_{k-m}[n]$$

The derivatives are done for each feature vector separately. [12]

There are other related features as delta Cepstral energy (DCE) and Delta-delta Cepstral energy (DDCE), which have been tested too. [11]

3.5. PLP and Rasta

This method was developed mainly by Hinek Hermansky and published in 1990 [14]. Here, several well-known properties of human hearing are simulated by practical engineering approximations and the resulting auditory spectrum is approximated by an autoregressive all-pole model of 5th order.

The three used concepts of the psychophysics of hearing are:

- *Critical-Band Spectral Resolution*
  This block is similar to the Mel-frequency warping in MFCC, but here is considered the Bark Scale, a bit different in comparison to the Mel Scale.
- *Equal-loudness curve*
  This step tries to approximate the non equal sensitivity of human hearing at different frequencies.
- *Intensity-loudness power law.*
  This is added in order to approximate the power law of hearing, which enunciates that there is a nonlinear relation between the intensity of a sound and its perceived loudness.

The author achieves more or less the same results as with common methods, but using a minor order model.

A later and improved version, called PLP-RASTA [15], was presented in 1994. Its purpose is to deal with additive and convolution noise using spectral subtraction.

The two main theoretical principles used are the relative unsentivity of human hearing to slowly varying stimuli and the suppression of slowly varying components in the speech signal which makes good engineering sense.

The rate of change of non-linguistic components in speech is out of the typical rate of change of the vocal tract shape. RASTA removes the spectral components that change slower and faster than the typical range of change of speech. This becomes useful in order to deal with additive and convolution noise.

3.6. Wavelets

As we saw, the problem of Speaker Identification can be divided into two major stages: feature extraction and matching based on the extracted features. Although these two components may appear to be independent, they are highly coupled. To be effective, the features should be capable of separating the speakers from each other in its space.

The identification results can be as high as 99.5% for a noise free database. However, for the same data set transmitted over telephone channels, the identification accuracy can be reduced to 60%. In fact, most of the applications for speaker identification will work over a noisy channel with the presence of background noise and convolution noise.

As it was commented, MFCCs are not immune to noise; these reasons motivated to formulate parameters which are less sensitive to such environments, as for example the use of wavelets for feature extraction. [16]

It is not the intention to present a theoretical overview of Wavelet Transformation here, but mainly interesting properties of Wavelets for Speaker Recognition will be presented. [13]

Using Wavelets instead of STFT (in MFCC) allows for overcoming the resolution of the STFT, and then a multiresolution analysis is performed.

Subband Based Cepstral Parameters (SBC) and Wavelet Packet Parameters (WPP) are formulated in order to allow embedded denoising or enhancement in the feature extraction stage rather than filtering the speech for improved speaker identification.

Discrete time implementation of wavelets and wavelet packets is based on the iteration of two channel filter-banks, which are subject to certain constraints, such as low pass and/or high pass branches on each level followed by a subsampling-by-two-unit. Unlike the wavelet transform which is obtained by iterating on the low pass branch, the filter bank tree can be iterated on either branch at any level, resulting in a tree structured filter bank which can be called a wavelet packet filter bank tree. The resulting transform creates a division of the frequency domain that represents the signal optimally with respect to the applied metric while allowing perfect reconstruction of the original signal.

A 24 subband decomposition with wavelet packets is constructed by cascading the basic two channel filter banks into various levels in order to approximate the Mel-Scale frequency division.

For the extraction procedure frames of 24 ms of duration are used. It is different from the 20 ms of the MFCC, because the number of samples per frame should be divisible per 64. As it is assumed that speech is sampled at 8 KHz, there are 192 samples per frame. Then, the speech frame is windowed by a Hamming function and preemphasized. [16]

The resulting subband divisions finely emphasize frequencies between 0 and 500 Hz which normally contain large portions of the signal energy. Equal partitions are used between 500 and 1750 Hz, where each subband width is 125 Hz, the remaining frequency axis is virtually the same as a Mel-scale division.

Filters	**1**	**2**	**3**	**4**	**5**	**6**
MFCC	28	89	154	224	300	383
WPP	31	94	156	219	281	344
Filters	**7**	**8**	**9**	**10**	**11**	**12**
MFCC	472	569	674	787	910	1043
WPP	406	469	563	688	813	938
Filters	**13**	**14**	**15**	**16**	**17**	**18**
MFCC	1187	1343	1512	1694	1892	2106
WPP	1063	1188	1313	1438	1563	1688
Filters	**19**	**20**	**21**	**22**	**23**	**24**
MFCC	2338	2589	2860	3154	3472	3817
WPP	1875	2125	2375	2750	3250	3750

Table 2: Comparison of center frequency (Hz) of 24 uniformly spaced (in Mel-scale) MFCC filter-banks and WPP subbands.

The wavelet packet transform is computed for the given wavelet tree, which results in a sequence of subband signals or the wavelet packet transform coefficients, at the leaves of the tree.

Then, as in MFCCs the derivation of parameters is performed in two stages. The first one consists of computing filter bank energies, and the second one is the decorrelation of the log filter bank energies with a DCT to obtain the MFCCs, here called SBC. The derivation of the SBC parameters follows the same process as when the filter bank energies are derived using the Wavelet Packet transform rather than the Short-Time Fourier Transform, and these features outperform MFCCs. One of the main reasons is because of the computation of subband signals with smooth filters.

The DCT step decorrelates the filter bank energies. It has been shown that a wavelet transform is a better decorrelator in coding applications, and we know that the Gaussian Mixture densities typically used to model speakers for identification, have diagonal covariances assuming that the components of the feature vector are independent of each other. The degree to which this assumption is satisfied partly depends on the transform which makes de decorrelation. Thus, we hypothesize that using wavelets, instead of the DCT, may satisfy the assumption better. [16]

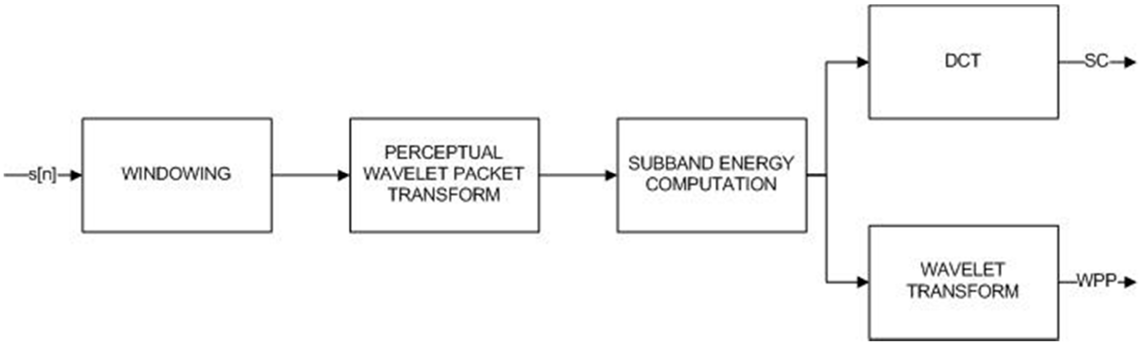
s[n]
WINDOWING
PERCEPTUAL WAVELET PACKET TRANSFORM
SUBBAND ENERGY COMPUTATION
DCT
SC
WAVELET TRANSFORM
WPP

Figure 15: Block diagram for Wavelet Packet Transform Features.

3.7. Increasing feature space

For text-dependent speaker verification, Cepstral features exhibit a discriminative power that is, as of now, unsurpassed by any other feature representation for speech. Therefore, if we want to improve speaker verification systems beyond the discriminative limit of Cepstral features we must incorporate additional features that provide independent information. [17]

An easy feature to obtain that we can consider is the pitch, exploiting a set of novel speech features derived from a principal component analysis (PC) weakly correlated with Cepstral features.

A distance measure combining Cepstral and PC provides a discriminative power that cannot be achieved by Cepstral alone. Increasing the feature space of the system we are able to reduce the error probabilities: incorrect customer rejection and incorrect impostor acceptance.

The new considered characteristics are derived from the local structure of voiced sections of the speech signal. Further improvements can be achieved by using several components of each pitch class instead of the main principal component.

3.8. New trends using high level information

Recent research done in the last decade has found that considering more levels of information and combining all of them at the end to take the accept/reject decision could improve a system's accuracy. Each info-level entails a classifier, which are then all weighted in the final information fusion process. [18]

Humans can activate different levels of speech perception according to specific circumstances, by having certain processing layers compensate for others affected by noise. These new trends seek to mimic this process. For this purpose, four classifiers were implemented targeting abstract speech levels:

- The *spectral-acoustic level*, which is the lowest level of information, traditionally treated with Cepstrum Analysis.
- The *phonetic level,* which is based on counts of discrete acoustic units, actually we are not representing traditional phonetics, so this term is not strictly appropriate, but rather abstract acoustic units resulting from clustering the Cepstrum space.
- The *prosodic level.* Its feature set consists of histogram bins of pitch and energy raw values and corresponding transitional tokens
- The *idiolectal level* uses a feature set of frequencies of common words; it becomes the highest level layer.

Lower communication layers are normally constrained by the speakers' vocal-tract anatomy, while higher levels are more affected by behavioral markers.

The fusion of low and high level information is done at the end, using a linear combination of classifiers, employing a meta-learner to obtain the optimal weights for the respective component learners.

Although these systems are very complex, it is an interesting approach. They are more suitable when we are considering systems which have to identify a speaker during a conversation instead of only telling a word or a code number.

3.9. Chapter conclusions

We have discussed some feature extraction techniques with their pros and cons. Initially, it seemed that we can find some arguments suggesting that detailed copying of the human hearing system might be useful, but sophisticated auditory models have not generally been found to be better than conventional representations outside the laboratories in which they were developed.

First of all, we do not know enough about it as a whole, and copying some parts and not others may be counter-productive.

Second, we do not know much at all about how the output from the human hearing system is used to recognize speech.

Definitely there are some features which had an audition motivation (the use of Mel-scale, cube-root representation and PLP), but their properties can be better understood in purely signal processing terms or in some cases in terms of the acoustic properties of the speech production process. [8]

So, there is more to learn from studying and modeling human speech production than studying and modeling speech perception, in order to improve speech signal representations.

Therefore, for our purpose, in order to increase the noise robustness, it is a better direction to develop methods using combination of more techniques, new hybrid methods that will give improved performance in noisy environments.

4. PRACTICAL IMPLEMENTATIONS

After a wide research to have a good theoretical background, the thesis work evolved to look for any practical implementations of Speaker Verification Systems in order to test its performance.

The search was not very successful and during the available time, it was possible to find only two open-code developed programs:

- **PRAAT 5.2.21**, is developed by Paul Boersma and David Weenink, who work in the Phonetic Sciences group in Amsterdam University (Netherlands). [19]
  On its site web you might find the latest version as it is periodically updated. Used here is the version of 7th April 2011. It is possible to work with that program on Windows OS, and there are several beginner's manuals and tutorials written by other people.
  Initially, it does not seem a specific program for speaker verification. Speech analysis, speech synthesis, listening experiments, learning algorithms, programmability, speech manipulation, etc… can be done with this program.

- **BECARS** developed in 2005 by University of Balamand (Liban) and University of GET-ENST Paris (France), provides a C library and several tools that permit to set up of the modeling and scoring phases of a GMM based ASV System. The current version is 1.1.9. [20]

There are also some different found solutions developed by companies, for example:

- **Speaker Verification Engine 1.21**, developed in 2009 by United Research Labs, has a 3-day trial version on its site web, otherwise you should pay 800 $ to have it. It is a voice-based identification system component that allows you to write a program that claims to prevent any sort of unauthorized access to your PC.

- **RecoMadeEasy Speaker Recognition,** this is the state of the art language and text independent speaker recognition system by Recognition Technologies, Inc. It has been developed to work in different environments as over telephone lines or stand-alone environments and runs on most Linux distributions as well as Apple Macintosh OS.
  This engine operates in six different modalities: *Speaker Identification, Speaker verification, Speaker Classification and Event Detection, Speaker Detection, Speaker Tracking, Speaker Segmentation*. The second modality fits for our purpose.

  In this modality, the speaker has to enroll his voice. Once the enrollment process is done (recording of about 30 seconds of speech and obtaining a positive ID of the speaker), the speaker is added to the database. When the speaker returns, he makes a claim of his identity. He will also speak for a few seconds and the speaker's voice is matched against the database. His identity is either authenticated or he is rejected as an impostor. [21]

As Speaker Verification Engine does not satisfy the initial requirements, there is no trial version of RecoMadeEasy Speaker Recognition and it was not possible to execute Becars software, the following work in this project was chosen to familiarize with Praat software and try to make sure that a Speaker Verification test may be done with it.

4.1. Introduction to Praat

First of all, it is possible to record or read a sound in order to work with it. A sound is treated as an object for the program. [22]

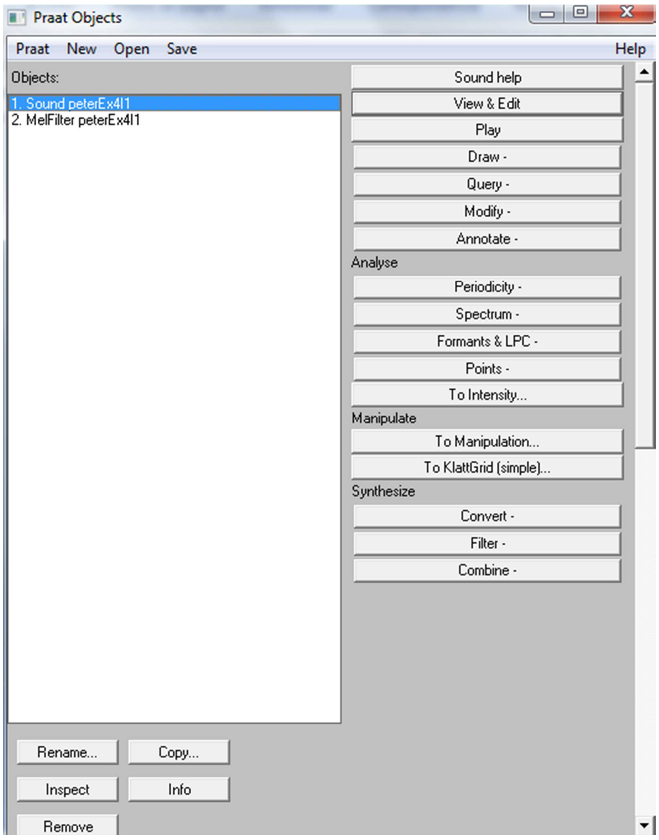

Figure 16: Capture of the main window of Praat where can be seen its tools and some objects ready to use.

The program shows the temporal representation and the spectrogram of the input signal at the same time (see Figure 15). There are some basic tools:

- *Spectral Analysis*
  Here it is possible to see the spectrogram. The time scale in the horizontal axis is the same as for the waveform and in the vertical direction there is frequency, usually between 0 Hz and 5 KHz.
  Darker parts in the spectrogram mean higher energy densities; lighter parts mean lower energy parts.
  We can set up the settings on how the spectrogram is obtained:

  -*View Range* (Hz)
  -*Window length*: the duration of the analysis window, it is necessary to be careful because it has direct influence on the bandwidth
  -*Dynamic Range* (dB)
  It is possible to view spectral slices of the spectrogram, called *spectrum*.

- *Pitch Analysis*
  There is a possibility to view a pitch contour in the spectrogram, drawn as a blue line or a sequence of blue dots. At the right of the window there is a floor value of pitch, usually 75 Hz and a top value, 600 Hz. When you click on the signal, the value of this point appears between the two bound values.

- *Formant Analysis*
  The formant contours of a sound as a function of time appear drawn as red speckles.

- *Intensity Analysis*
  The intensity contour of a sound as a function of time is drawn as a yellow or green line.

- *Manipulation of pitch, duration, formants and intensity*
  It is also possible to modify these contours in an easy and precise way, but we are not interested in that.

For each analysis we can get an object in the program list of objects in the main window and also we can "print" the representation in order to use in a word processor.

Below the list, there are two useful buttons, inspect and info. With the first one we can get all the information about the object and with the other one a summary. For example for an input sound, we can know the maximum and minimum amplitude values, the duration of the file, the sampling frequency and the number of samples.

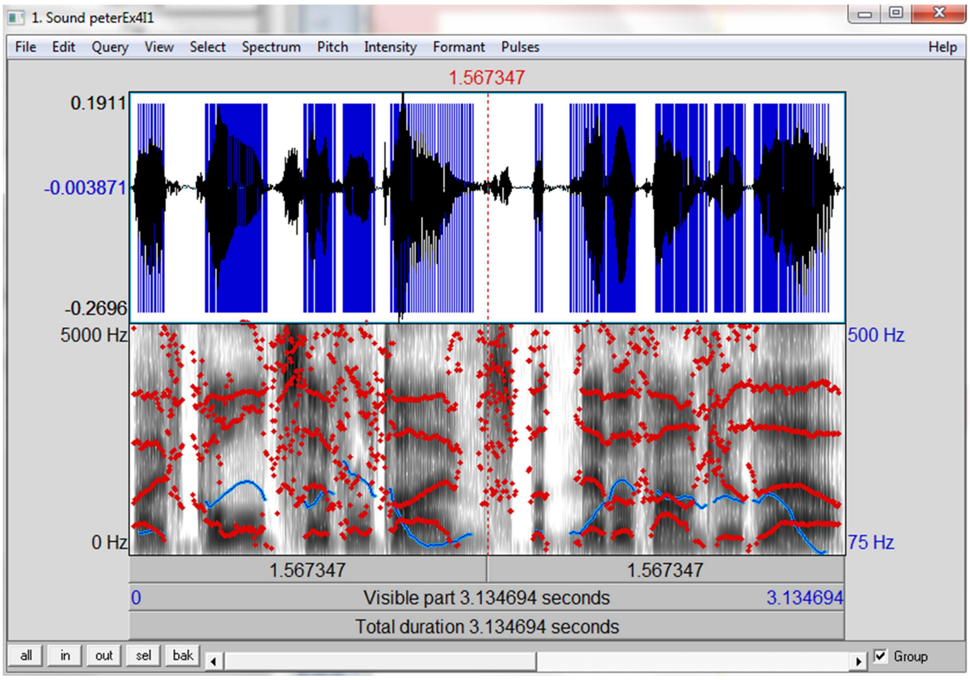
1. Sound peterEx4I1
File Edit Query View Select Spectrum Pitch Intensity Formant Pulses Help
1.567347
0.1911
-0.003871
-0.2696
5000 Hz
0 Hz
500 Hz
75 Hz
1.567347
1.567347
0
Visible part 3.134694 seconds
3.134694
Total duration 3.134694 seconds
all in out sel bak
☑ Group

Figure 17: Example of the signal representation in Praat, notice that here appear some of the analysis mentioned before.

4.2. Praat for Speaker Verification

After a brief introduction to the Praat program, we want to deal with our speaker verification problem. So, we need to know if we can get a Cepstral representation of the input signal or any one of the representations which were discussed in the chapter before, and how to obtain them.

As we saw, Praat works with objects. In the analysis menu in the main window, we may find some of the tools which we are looking for. We can create a MFCC object from an input sound and later perform a DTW comparison between two MFCC objects. It is possible to look up how the program implements the MFCC algorithm in its help.

Analysis performance

First, there should be performed a filter bank analysis on a Mel frequency scale. It is necessary to choose some parameters as the window length (seconds), time step (seconds), position of the first filter (Mel), the distance between filters (Mel) and the maximum Mel frequency. Then the Mel filter object is obtained from the input sound. This represents the power spectral density *P (f,t)* expressed in dB's. It is sampled into a number of points around equally spaced times t_i and frequencies f_i.

Afterwards, when the Mel filter object is chosen, the filter values are converted to Mel frequency Cepstral coefficients. Here we can choose how many coefficients we want. Later, we obtain the MFCC object in the main window. This one represents MFCC's as a function of time; the coefficients are represented in frames with constant sampling period. Then, with the Info button it is possible to know for example which minimum and maximum frequencies are considered for the analysis and the total number of frames. [22]

After selecting two objects with Cepstral coefficients we can perform a dynamic time warping, where the distance between Cepstral coefficients is calculated, and then is found the optimum path through the distance matrix with a Viterbi-algorithm. Here, it is possible to choose the boundary conditions, matching the beginning and/or ending positions of the input voice prints, set different slope constraints and different parameters referring to the energy of the signals.

When we have the DTW object in the main window of the program, selecting the info button we can access to a summary of the final comparison results, there are the domain for the prototype (pre-recorded sample) and candidate (claiming speaker) expressed in seconds, the number of frames for both, the path length (in frames), the global warped distance and distance along the diagonal (if both samples have the same length).

Selecting the inspect button we can access to the warping path matrix. Then, in order to obtain a graphic idea of the result, we can select the two input sounds objects and the DTW object and draw the input sounds and the warping path at the same time. Here, it is necessary to be careful with the time domains of the input sounds.

If the distance matrix has *m* cells along the x direction and *n* cells along the y direction, and the sum of distances along the minimum path is S, the weigthed distance is given by:

$$Weighted\ distance = \frac{S}{n+m}$$

At this point a first basic test is done in order to check that the program works as we expect. A recorded phrase of a man is used as inputs, and the result of the DTW algorithm is 0 and the graphic repsentation of the warping path is a diagonal line. Then, we distort progressively the signal with Praat and compare with the original one, so that the result of the comparison increases.

The values for the parameters mentioned in this chapter, which are necessary to perform tests will be discussed in the following chapter.

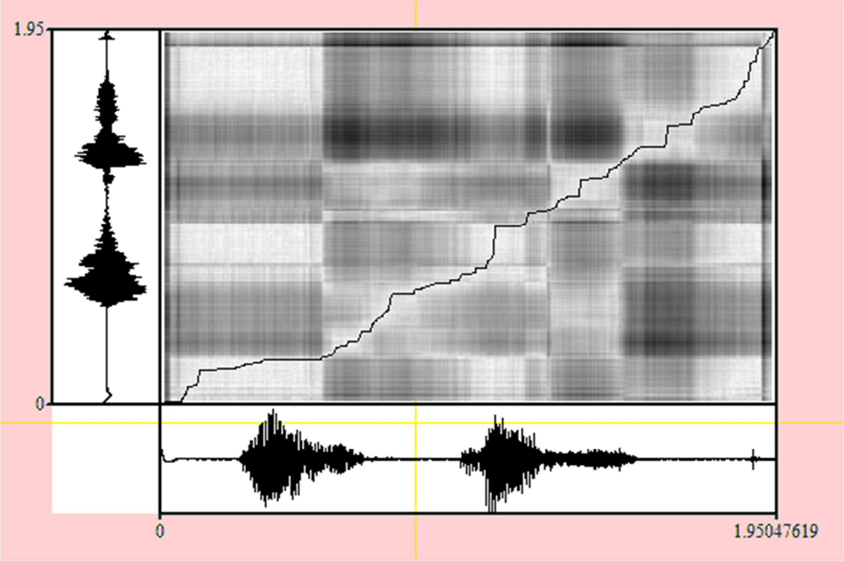

Figure 18: Example of a warping path obtained with Praat, using as inputs samples from the same speaker.

5. PERFORMANCE OF A TEST WITH PRAAT

In this chapter, tests using the software presented will be done. Before beginning the tests, the discussion group of Praat on the Internet was consulted in order to be sure that the procedure that will be followed is the correct one.

First of all, it is necessary to explain the conditions under which the tests are done. Then, the results will be presented and discussed in order to get some conclusions. From now, it is necessary to remind of the values for the Verification system performance, because some of them will be needed as input parameters for Praat.

5.1. Experiment settings

A HP microphone embedded in a laptop was used to record the voice samples. As it is possible to record sounds with Praat, for simplicity the same program is used for the purpose. The WAV format is chosen for it, with a sampling frequency of 44100 Hz and only one audio channel.

5.1.1. First tests to set parameters

First of all, some tests are done in order to set the parameters necessaries for the program. The parameters shown in the following tables are: global warped distance, path length (in frames) and the distance along the diagonal, all of them were described on the fourth chapter. The number of frames of the voiceprints is added in order to get an idea about the duration of the sample.

Test1. A window length of 20 ms and a time step of 10 ms are chosen, so there is an overlapping of 50% between frames. The position of first filter is 100 Mel, the distance between filters is 100 Mel and the maximum frequency 4000 Mel. These values are chosen because some researchers affirm that they are the optimal ones. [23]

Two voiceprints P_2 and P_3 are used in order to compare with P_1, all of them are from the same speaker, then S_1 and S_2 from a different speaker are compared with P_1. Different numbers of MFC Coefficients are considered; 12, because it is enough for speaker recognition tasks, 24 is proposed as an optimal value [23] and 36 is the maximum value able for the program.

	Frames		**12 MFCC**	**24 MFCC**	**36 MFCC**
P_1	201	**Global warped distance**	87.46	109.71	116.33
P_2	201	**Path length (frames)**	351	355	356
		Distance along diagonal	224.65	239.47	243.68

	Frames		**12 MFCC**	**24 MFCC**	**36 MFCC**
P_1	201	**Global warped distance**	89.67	112.08	118.35
P_3	196	**Path length (frames)**	352	354	359

	Frames		**12 MFCC**	**24 MFCC**	**36 MFCC**
P_1	201	**Global warped distance**	128.17	147.08	153.09
S_1	224	**Path length (frames)**	394	398	398

	Frames		**12 MFCC**	**24 MFCC**	**36 MFCC**
P_1	201	**Global warped distance**	114.41	135.25	141.48
S_2		**Path length (frames)**	420	425	423

Test2. As for default in the program, the value for window length is 15 ms and for the time step 5 ms, it is interesting to test them. The values for the frequencies are the same than in Test1.

	Frames		**12 MFCC**	**24 MFCC**	**36 MFCC**
P_1	403	**Global warped distance**	87.69	110.28	117.29
P_2	403	**Path length (frames)**	745	745	750
		Distance along diagonal	228.35	243.92	248.18

	Frames		**12 MFCC**	**24 MFCC**	**36 MFCC**
P_1	403	**Global warped distance**	88.94	113.19	119.88
P_3	394	**Path length (frames)**	732	734	734

	Frames		**12 MFCC**	**24 MFCC**	**36 MFCC**
P_1	403	**Global warped distance**	126.02	146.86	152.89
S_1	450	**Path length (frames)**	820	821	812

	Frames		**12 MFCC**	**24 MFCC**	**36 MFCC**
P_1	403	**Global warped distance**	111.75	133.68	140.16
S_2	505	**Path length (frames)**	867	867	867

Test3. This one is performed to compare the effect of a pre-emphasize filter.

	Frames		**12 MFCC**	**12 MFCC filter from 20 Hz**	**12MFCC filter from 50 Hz**
P_1	201	**Global warped distance**	87.46	87.71	87.46
P_2	201	**Path length (frames)**	351	353	351
		Distance along diagonal	224.65	226.20	225.58

	Frames		**12 MFCC**	**12 MFCC filter from 20 Hz**	**12MFCC filter from 50 Hz**
P_1	201	**Global warped distance**	128.17	128.50	128.73
S_1	224	**Path length (frames)**	394	394	394

For Test1, Test2 and Test3 the following parameters have been chosen in the DTW step: beginning and ending matched positions and no slope constraints.

The following tests are done in order to compare the setting parameters for the DTW algorithm. Moreover, at the end the warping paths are included in order to show how they may change according to the parameters chosen.

Test4. Here, the values of frequencies and times of Test1 are chosen, as well as 24 MFCC.

	Frames		**Beginning & Ending positions matched**	**No Beginning & Ending positions matched**
P_1	201	**Global warped distance**	109.71	83.31
P_2	201	**Path length (frames)**	355	236
		Distance along diagonal	239.47	239.47

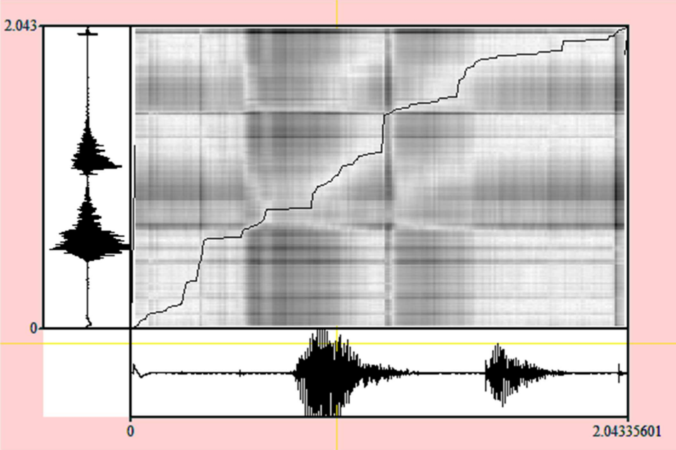

Figure 19: Capture of a warping path with Beginning & Ending positions matched.

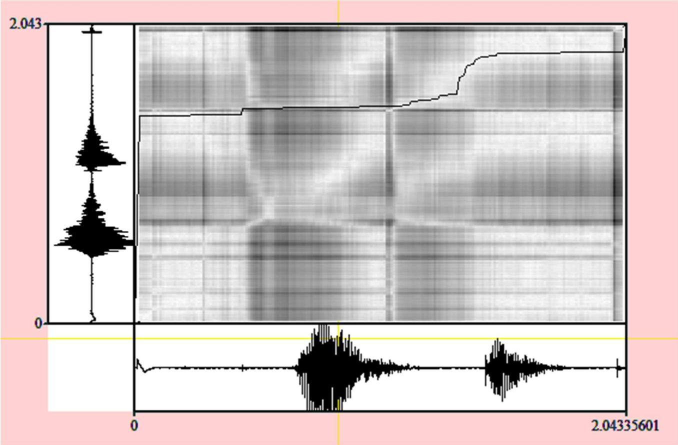

Figure 20: Capture of a warping path with no Beginning & Ending positions matched.

Test5. This test is performed changing the slope constraints. The resulting warping paths are shown later.

	Frames		**No constraints**	**1/3<Slope<3**	**1/2<Slope<2**	**2/3<Slope<3/2**
P_1	201	**Global warped distance**	109.71	123.99	141.25	170.88
P_2	201	**Distance along diagonal**	355	239.47	239.47	239.47
		Path length (frames)	239.47	277	251	235

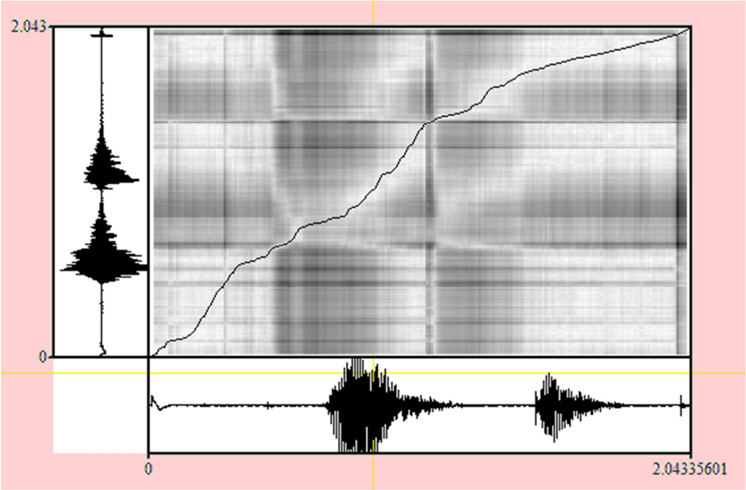

Figure 21: Capture of a warping path with the slope constraints: 1/3<Slope<3.

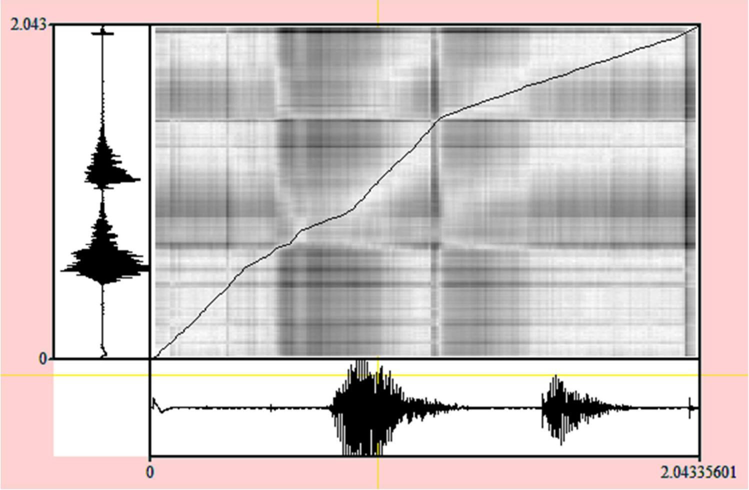

Figure 22: Capture of a warping path with the slope constraints: 1/2<Slope<2.

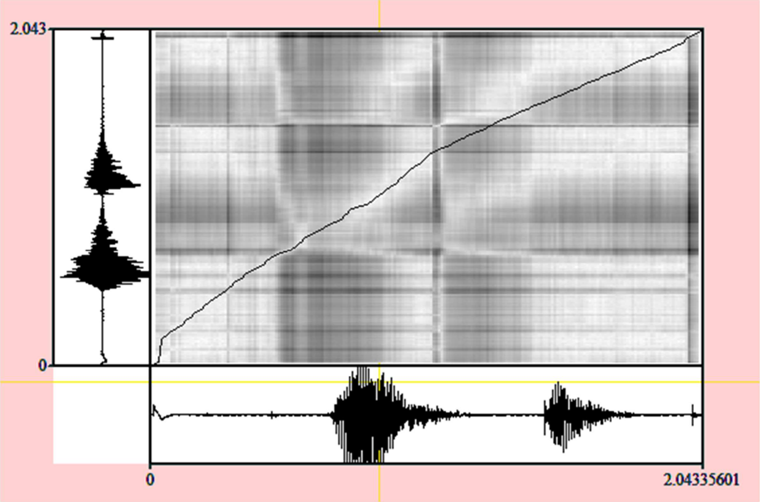

Figure 23: Capture of a warping path with the slope constraints: 2/3<Slope<3/2.

5.1.2. Parameters chosen and first conclusions

We will begin by discussing the parameters for the MFCC. The differences in the global warped distance between Test1 and Test2 are not significant, so for the Mel-warping step, the window length of 20 ms and a time step of 10 ms will be considered. With these parameters an overlapping of 50% between frames is achieved and with this window it is enough to have a quasi stationary signal.

On the other hand, the differences are higher in terms of the warped distance when different number of MFC coefficients are chosen. The results are lower when 12 MFCC is chosen instead of 24. The MFCC with 12 coefficients will be used in the following tests because it seems that we could distinguish better the results for same speaker tests and impostors claimings.

The use of the pre-emphasis filter does not seem to change significantly the global warped distance, as it can be seen in Test3, so in order to simplify the work this step will be omitted.

Finally, for the DTW algorithm; the slope constraints seem to be useful for a graphic result, as a more linear representation for the warping path may be obtained, but not in the numerical result as the warped distance increases.

The beginning and ending matching as boundary conditions will be used in the following tests, because it is one of the several constraints which the warping path is subject to. [4] It can be seen that when this setting is not selected the lowest value (in all the tests) for the warping path is obtained.

Step	**Parameters**	**Values**
Mel-Warping	*Window Length*	20 ms
	Time Step	10 ms
	Position (1st)	100 Mel
	Distance between filters	100 Mel
	Maximum frequency	4000 Mel
MFCC	*Number of coefficients*	12
DTW	*Boundary conditions*	Selected
	Slope Constraints	Omitted

Table 3: Summary with the parameters chosen.

5.2. Speaker tests

Now with all the parameters of the software discussed, it is possible to perform a more complete test. As it was mentioned that verification systems achieve higher accuracies when the number of comparisons increases, for this reason a larger number of samples will be used and four different speakers will participate in the test. They are assigned as “P”, “R”, “S” and “A”, respectively in the test results.

As we are working with several values obtained from comparisons between recordings, it is necessary to work out some statistical parameters, so the mean and the standard deviation of sample (all the values are considered and not only a population of them) are calculated with the following expressions:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} X_i$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i - \mu)^2}$$

5.2.1. Intra-speaker tests

Four speakers took part in the performance test with 8 voice prints each saying the numbers "one-two". These are the pre-recorded samples. In another session, one day later, 3 samples of their voices are compared with the pre-recorded ones. All the recordings were done under clean conditions in an empty room. The aim of this procedure is to set a main value and a rank between it for the acceptance of a claiming user. These values will depend on the user.

The following tables present the information of the comparisons. The details of the samples can be found in the Appendix. The duration of the voiceprints sometimes is the same, so the mean power of the voice sample is showed in order to distinguish them better.

Name of voice print	S_1E	S_2E	S_3E
S_1	103.65	108.57	115.53
S_2	98.24	97.06	105.29
S_3	102.51	110.30	113.43
S_4	101.21	102.39	107.09
S_5	92.97	94.04	95.50
S_6	97.26	95.30	101.21
S_7	94.68	100.70	102.94
S_8	95.16	96.99	100.60

Table 4: Comparison results for the first speaker. The global warped distance.

Mean and standard deviation obtained: μ_1 =101.36 σ_1 = 6.05

Name of voice print	P_1E	P_2E	P_3E
P_1	95.42	99.41	116.68
P_2	85.23	85.32	96.08
P_3	79.02	76.81	90.83
P_4	76.98	78.19	88.92
P_5	79.77	83.21	91.33
P_6	80.79	79.29	91.00
P_7	86.43	77.88	85.70
P_8	79.23	81.98	93.20

Table 5: Comparison results for the second speaker. The global warped distance.

Mean and standard deviation obtained: μ_2 = 86.61 σ_2 = 9.04

Name of voice print	R_1E	R_2E	R_3E
R_1	83.21	75.42	81.58
R_2	80.52	75.12	81.01
R_3	84.12	77.79	81.05
R_4	85.22	77.11	84.30
R_5	87.01	82.30	84.53
R_6	71.51	67.26	74.13
R_7	88.08	82.44	89.41
R_8	88.87	79.37	81.93

Table 6: Comparison results for the third speaker. The global warped distance.

Mean and standard deviation obtained: μ_3=80.97 σ_3= 5.40

Name of voice print	A_1E	A_2E	A_3E
A _1	69.27	72.92	81.12
A _2	70.44	78.40	85.61
A _3	74.71	78.72	83.70
A _4	95.38	102.47	90.78
A _5	80.10	83.47	90.64
A _6	75.67	74.86	87.45
A _7	85.65	83.45	86.74
A _8	83.43	79.53	89.18

Table 7: Comparison results for the fourth speaker. The global warped distance.

Mean and standard deviation obtained: μ_4 =82.65 σ_4 = 7.73

As it was said, and also can be noticed now, the values obtained depend on the user. Users with higher values of mean, have higher differences between their voiceprints. The values of deviation show the differences between the comparisons.

After that, another performance was done, the same speakers took part in the following test. Here, they say "un-dos" the same numbers, but spoken in Catalan. The number of voice-prints considered and the procedure is the same as before. The aim is to see how different are the values of each speaker saying another word.

Name of voice print	S_1CE	S_2CE	S_3CE
S_1C	122.70	122.70	125.09
S_2C	112.49	112.49	126.29
S_3C	105.32	122.10	120.65
S_4C	103.66	128.46	121.04
S_5C	110.36	117.78	124.99
S_6C	121.06	126.60	125.34
S_7C	133.09	133.65	129.28
S_8C	124.10	129.52	127.37

Table 8: Comparison results for the first speaker (Catalan). The global warped distance.

Mean and standard deviation obtained: μ_1 = 121.92 σ_1 = 7.81

Name of voice print	P_1CE	P_2CE	P_3CE
P_1C	91.72	95.82	98.12
P_2C	92.49	103.24	95.37
P_3C	83.73	90.37	84.07
P_4C	94.25	99.13	95.66
P_5C	85.26	91.33	90.04
P_6C	88.23	96.30	91.62
P_7C	87.32	94.84	91.80
P_8C	80.50	84.54	83.99

Table 9: Comparison results for the second speaker (Catalan). The global warped distance.

Mean and standard deviation obtained: $\mu_2 = 91.23$ $\sigma_2 = 5.57$

Name of voice print	R_1CE	R_2CE	R_3CE
R_1C	96.92	97.45	99.10
R_2C	103.67	99.79	107.09
R_3C	141.53	117.33	141.88
R_4C	102.65	94.99	103.64
R_5C	101.61	96.88	100.53
R_6C	98.60	94.79	100.92
R_7C	103.31	93.95	106.21
R_8C	116.63	102.75	119.55

Table 10: Comparison results for the third speaker (Catalan). The global warped distance.

Mean and standard deviation obtained: $\mu_3 = 105.91$ $\sigma_3 = 12.68$

Name of voice print	A_1CE	A_2CE	A_3CE
A _1C	87.93	89.67	92.95
A _2C	69.05	75.59	72.11
A _3C	93.37	95.75	91.43
A _4C	79.29	82.35	86.52
A _5C	92.14	83.09	84.43
A _6C	82.57	91.83	88.31
A _7C	76.80	79.44	73.18
A _8C	78.16	75.63	72.14

Table 11: Comparison results for the fourth speaker (Catalan). The global warped distance.

Mean and variance obtained: $\mu_4 = 83.07$ $\sigma_4 = 7.74$

With the results obtained, it is possible to visually confirm that the variability within the voiceprints for each speaker depend on the word which they are saying.

As a curious fact, a comparison of samples from the same speaker was done, but saying different words. For this purpose the pre-recorded samples of the speaker "P" saying "one-two" are compared to three of him saying "un-dos".

Name of voice print	P_1CE	P_2CE	P_3CE
P_1	129.37	134.56	135.36
P_2	117.80	126.02	120.26
P_3	116.83	126.50	123.59
P_4	116.53	124.42	118.87
P_5	113.33	120.91	118.71
P_6	111.11	119.23	116.57
P_7	114.43	115.52	116.08
P_8	119.89	123.20	121.55

Table 12: Comparison results of different words of same speaker. The global warped distance.

Mean and standard deviation obtained: $\mu_{2'} = 120.86$ $\sigma_{2'} = 6.04$

The mean for this speaker was $\mu_2 = 86.61$ when he said the combination "one-two", so here it is possible to see clearly that we are working with a text-dependent system.

5.2.2. Inter-speaker tests

The samples recorded for the previous test are used in this one. Here, voice printings of the different speakers are compared in order to find out if hypothetical impostors could achieve access, taking into account the values obtained of each speaker in the last section.

In the first example, speakers P, R and A claim to be the first one, S. Because of S has the highest mean value, more variability within his voiceprints, so is liable to be mistaken easier by other speakers. The three claiming voiceprints are compared with the eight pre-recorded samples of the first speaker.

Name of voice print	P_1E	P_2E	P_3E
S_1	119.24	106.63	121.77
S_2	107.45	95.56	112.62
S_3	124.80	109.18	131.55
S_4	115.58	109.37	121.65
S_5	112.91	106.36	119.27
S_6	111.83	100.49	110.68
S_7	112.27	101.62	114.48
S_8	114.66	101.88	114.40

Table 13: Comparison results first speaker compared with the second. The global warped distance.

Mean and deviation obtained: $\mu_{12} =$ 112.34 $\sigma_{12} = 8.12$

Name of voice print	R_1E	R_2E	R_3E
S_1	129.27	129.06	126.82
S_2	129.17	129.71	128.74
S_3	133.27	128.58	125.94
S_4	132.66	129.52	129.51
S_5	123.55	122.32	120.73
S_6	116.93	119.60	116.84
S_7	128.30	123.31	126.06
S_8	127.45	123.64	126.42

Table 14: Comparison results first speaker compared with the third. The global warped distance.

Mean and deviation obtained: $\mu_{13} =$ 126.14 $\sigma_{13} = 4.35$

Name of voice print	A_1E	A_2E	A_3E
S_1	109.28	109.67	121.50
S_2	100.55	108.58	121.36
S_3	103.92	105.05	119.78
S_4	99.88	101.82	117.41
S_5	94.02	102.83	113.87
S_6	91.17	97.90	112.79
S_7	96.02	104.43	115.98
S_8	97.30	102.74	112.25

Table 15: Comparison results first speaker compared with the fourth. The global warped distance.

Mean and deviation obtained: $\mu_{14} =$ 106.67 $\sigma_{14} = 8.61$

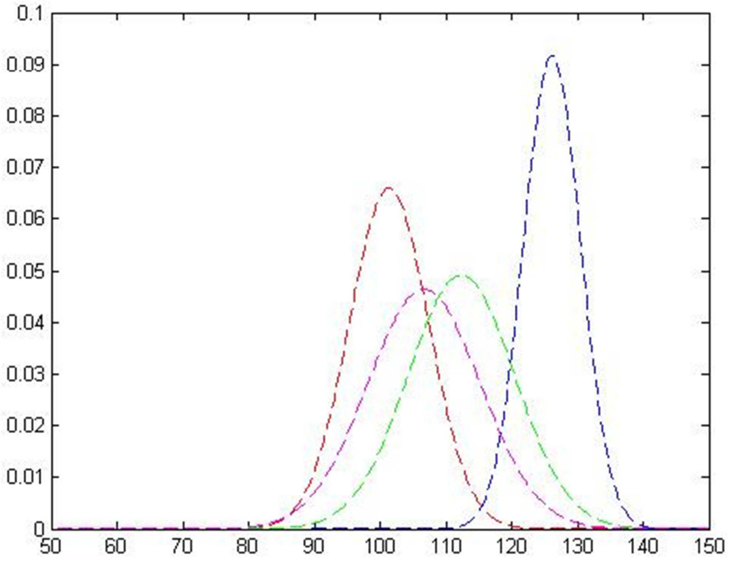

Figure 24: Representation of the different gaussian functions, obtained with Matlab. The claimed speaker corresponds to the red one, and the the impostors crossed with the claimed are magenta, green and blue.

Then, it should be taken into account that there is several ways to fix a threshold for the acceptances and rejections, mentioned in point 2.4. There are also shown the areas to integer in order to know the probabilites of False Rejection and False Acceptance.

Looking at the graphic result, it may be seen clearly that the fourth speaker (magenta) would create problems to an hypothetical system, as the mean obtained when his voiceprints are crossed with the first speaker is closer to the first speaker mean. Then, a threshold **T** is set in order to get a numerically idea of this fact.

Parameter	**Value (%) T=105**	**Value (%) T=106**	**Value (%) T=107**
False Rejection (1st speaker)	30.17	24.66	19.75
False Acceptance (2nd speaker)	19.96	23.59	27.54
False Acceptance (3rd speaker)	$9.91 \cdot 10^{-5}$	$3.02 \cdot 10^{-4}$	$8.75 \cdot 10^{-4}$
False Acceptance (4th speaker)	44.59	49.21	53.84

Table 16: Values of FR and FA obtained when different thresholds are considered.

The probabilites of False Rejection and False Acceptance are presented as a measure of overall system. There is a tradeoff between them, an ideal system would have zero values for both. In this case is hard to define which would be the best threshold because there is always a high value for the false acceptance of the fourth speaker.

It should be noticed that the procedure could be the other way round, first False Rejection and False Acceptance values are fixed, and with them the threshold value is worked out.

Now, the speaker with the lowest mean value is considered, this is the third one. Initially, it seems to be the favorable case. Then, the same procedure as it was done before is followed.

Name of voice print	S_1E	S_2E	S_3E
R_1	111.30	106.44	107.45
R_2	112.06	106.57	107.39
R_3	112.58	108.40	105.12
R_4	108.22	102.29	95.17
R_5	115.25	107.90	104.03
R_6	111.24	107.17	108.67
R_7	116.02	118.25	117.57
R_8	110.51	109.32	109.59

Table 17: Comparison results third speaker compared with the first. The global warped distance.

Mean obtained and deviation: $\mu_{31} = 109.10$ $\sigma_{31} = 4.40$

Name of voice print	P_1E	P_2E	P_3E
R_1	97.40	99.10	105.86
R_2	97.76	98.25	108.92
R_3	96.10	95.58	106.95
R_4	94.65	94.49	103.68
R_5	90.45	95.67	105.80
R_6	98.89	98.54	110.83
R_7	113.42	119.02	121.87
R_8	98.63	99.30	108.58

Table 18: Comparison results third speaker compared with the second. The global warped distance.

Mean and deviation obtained: $\mu_{32} = 102.49$ $\sigma_{32} = 7.90$

Name of voice print	A_1E	A_2E	A_3E
R_1	92.00	107.33	113.99
R_2	89.64	106.82	113.36
R_3	92.58	110.75	117.32
R_4	92.53	110.74	114.80
R_5	94.03	111.85	114.09
R_6	89.60	105.85	108.66
R_7	105.46	121.45	112.94
R_8	94.22	113.24	115.30

Table 19: Comparison results third speaker compared with the fourth. The global warped distance.

Mean and deviation obtained: $\mu_{34} = 106.19$ $\sigma_{34} = 9.74$

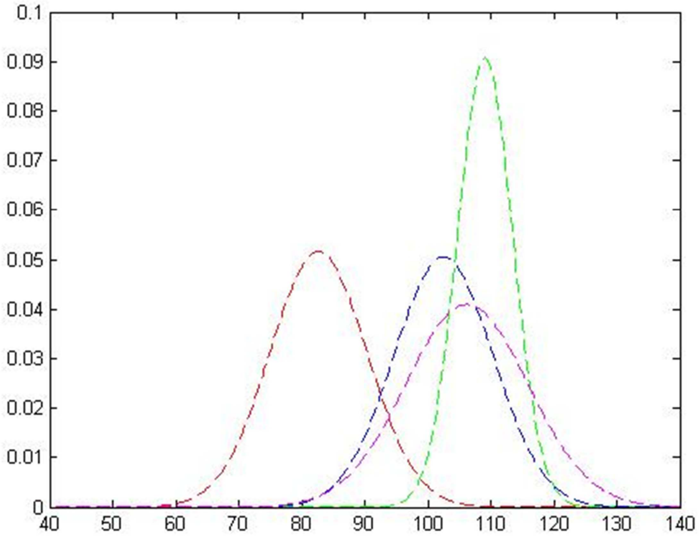

Figure 25: Representation of the different gaussian functions, obtained with Matlab. The claimed speaker corresponds to the red one, and the the impostors crossed with the claimed are magenta, green and blue.

Looking at the graphic result, it may be seen clearly that in this case the gaussians obtained when the speakers voiceprints are crossed are further than in the first presented case.

Parameter	**Value (%) T=89**	**Value (%) T=90**	**Value (%) T=91**
False Rejection (3rd speaker)	22.44	18.76	15.48
False Acceptance (1st speaker)	$4.02 \cdot 10^{-4}$	$1.14 \cdot 10^{-3}$	$3.05 \cdot 10^{-3}$
False Acceptance (2nd speaker)	4.99	6.44	8.2
False Acceptance (4th speaker)	4.32	5.35	6.57

Table 20: Values of FR and FA obtained when different thresholds are considered.

In this case, as it was expected, the values obtained for the False Rejection and False Acceptance are better, and closer to results presented in documents looked through during the thesis working. Perhaps, the tradeoff between the two probabilites would set the threshold in 90.

But, if the properties of the speech signal are remembered, it is known that there are several factors which contribute to change voice signal, so a Speaker Verification will never have a specific threshold for each speaker in its memory. This threshold is adaptative taking into account the environment.

Conclusively, it has been shown that is possible to reject hypothetical impostors in a Verification System using Praat, although within certain narrow limits and under low level noise conditions. As it was said, Praat is not an specific software for Speaker recognition.

It was expected to obtain lower values for the comparisons of voiceprints from the same speaker and higher values when voiceprints from different speakers are compared. Then, a larger confidence interval for the means could be obtained, but it was not like this.

5.3. Larger test

In order to find out if the number of comparisons is really a determinant factor, a final test was done. Here, the speaker named as “P” recorded hundred new voiceprints of the numbers “one-two”. The procedure followed is the same as in point 5.2. The information about the samples is omitted in this case, and the details of the comparisons are shown in the Appendix, as the main interest are the values for the mean and the deviation.

The values obtained are the followings: $\mu = 85.23$ $\sigma = 5.23$.

Taking into account the values obtained when 24 comparisons are done: $\mu_2 = 86.61$ $\sigma_2 = 9.04$, with 300 comparisons the mean value is reduced by a 1.59 % and the deviation by a 42.14 %. So, it is shown that with more comparisons, higher accuracy for the system may be achieved.

It should also be considered that recording so many voiceprints of the same speaker may become boring for him or her. In this case to get a hundred of recordings, around thirty minutes were necessary.

6. CONCLUSIONS AND FUTURE WORK

In general, Speech recognition is a knowledge field with high interest, and this may be confirmed with the amount of research papers published every year. Nowadays, there are also important companies which have research projects in this area.

But, when specific theoretical solutions for Speaker verification are looked for, this amount is reduced. Even though, there is a common way to face up to the problem using MFCC and DTW techniques (longer tested, with high accuracy results working in clean conditions and not too much difficult to implement), there are not many avalaible practical solutions.

The research papers looked through, present interesting techniques in order to deal with the drawbacks of MFCC and DTW, and so it could be possible to achieve better results when worse working conditions are considered, mainly the presence of noise. The use of Wavelets instead of MFCC, seems to be a viable and easy solution to implement, as it is a wide used representation in other signal processing fields, for example image processing. Also the more complex hybrid methods using more features in order to increase the accuracy take shape as good approaches.

But, the results obtained by researchers, seem only to be possible to obtain under certain special conditions in their own research centers. They usually do not explain with much detail how to obtain them, so it is hard for oneself to implement and test these new techniques shown in papers.

Praat is not a specific program for Speaker Verification, but as a tool which pretends to offer the possibility to work with speech recognition, it allows us to perform tests using the MFCC and DTW techniques. It is quite easy to use, if someone does not want to worry with how the program implements the techniques.

It is hard to draw conclusions from the small tests done using the program. But, as it was shown, it is possible to reject impostors who claim to be another speaker, not obtaining too high values for the false acceptances probabilites.

So, where this thesis finishes, another one can begin, with a more “realistic” implementation of a Speaker Verification System using Praat. After a time working on Praat, one could be able to program scripts to run in the program, in the Praat’s Manual there is a whole chapter about this topic.

A more complete test might be done using a larger number of speakers and theirs samples. Different working situations can be considered to know the real accuracy of the program when it works in environments discussed in Chapter 2. Having a larger number of voiceprints and their corresponding comparisons may allow us to reject the worst samples when the statistical calculations are done.

From another point of view, it could be possible to develop a package for Praat, in order to test any of the feature extraction techniques discussed in this thesis, instead of MFCC. Functionalities can be extended in Praat by adding C or C++ code to it. It may also be possible to work with the other programs presented in this thesis.

APPENDIX

Name of voice print	Duration (s)	Number of frames	Mean power (dB)
S_1	2.275	224	75.52
S_2	2.554	252	76.92
S_3	2.229	219	77.49
S_4	2.229	219	76.07
S_5	2.267	233	74.89
S_6	2.461	243	74.03
S_7	2.229	219	74.25
S_8	2.236	210	73.98

Table 21: Information of the pre-recorded samples in English of the first speaker.

Name of voice print	Duration (s)	Number of frames	Mean power (dB)
S_1E	2.275	224	72.03
S_2E	2.267	215	73.14
S_3E	2.136	210	73.61

Table 22: Information of the claiming session in English of the first speaker.

Name of voice print	Duration (s)	Number of frames	Mean power (dB)
P_1	2.043	201	77.10
P_2	2.043	201	75.08
P_3	1.996	196	73.20
P_4	1.857	182	75.54
P_5	1.811	178	75.11
P_6	2.275	224	75.66
P_7	1.857	182	75.30
P_8	1.904	187	76.32

Table 23: Information of the pre-recorded samples in English of the second speaker.

Name of voice print	Duration (s)	Number of frames	Mean power (dB)
P_1E	1.950	192	74.14
P_2E	2.368	233	71.97
P_3E	1.718	168	73.63

Table 24: Information of the claiming session in English of the second speaker.

Name of voice print	Duration (s)	Number of frames	Mean power (dB)
R_1	2.043	201	73.88
R_2	1.904	187	72.06
R_3	1.997	196	74.11
R_4	1.997	196	72.09
R_5	2.043	201	72.8
R_6	1.904	187	71.6
R_7	1.997	196	73.2
R_8	1.997	196	73.09

Table 25: Information of the pre-recorded samples in English of the third speaker.

Name of voice print	Duration (s)	Number of frames	Mean power (dB)
R_1E	1.765	173	73.28
R_2E	1.904	187	70.79
R_3E	2.043	201	72.28

Table 26: Information of the claiming session in English of the third speaker.

Name of voice print	Duration (s)	Number of frames	Mean power (dB)
A_1	2.136	210	71.56
A _2	2.043	201	68.97
A _3	1.904	187	71.12
A _4	1.764	173	70.41
A _5	1.811	178	69.56
A _6	1.997	196	71.08
A _7	1.765	173	70.77
A _8	1.718	168	72.35

Table 27: Information of the pre-recorded samples in English of the fourth speaker.

Name of voice print	Duration (s)	Number of frames	Mean power (dB)
A _1E	1.625	159	72.62
A _2E	1.997	196	72.26
A _3E	1.950	192	71.74

Table 28: Information of the claiming session in English of the fourth speaker.

Name of voice print	Duration (s)	Number of frames	Mean power (dB)
S_1C	2.415	238	75.74
S_2C	2.740	270	75.85
S_3C	2.368	233	75.54
S_4C	2.183	215	74.19
S_5C	2.275	224	75.58
S_6C	2.183	215	74.82
S_7C	2.090	205	76.19
S_8C	2.275	224	76.1

Table 29: Information of the pre-recorded samples in Catalan of the first speaker.

Name of voice print	Duration (s)	Number of frames	Mean power (dB)
S_1CE	2.415	238	73.01
S_2CE	2.275	224	75.5
S_3CE	2.51	247	75.1

Table 30: Information of the claiming session in Catalan of the first speaker.

Name of voice print	Duration (s)	Number of frames	Mean power (dB)
P_1C	1.950	192	76.38
P_2C	2.136	210	75.64
P_3C	2.089	205	74.75
P_4C	1.857	182	75.20
P_5C	1.857	182	75.08
P_6C	1.857	182	75.18
P_7C	1.811	178	75.29
P_8C	1.950	192	75.37

Table 31: Information of the pre-recorded samples in Catalan of the second speaker.

Name of voice print	Duration (s)	Number of frames	Mean power (dB)
P_1CE	1.904	187	73.0
P_2CE	1.764	173	72.99
P_3CE	1.764	173	74.04

Table 32: Information of the claiming session in Catalan of the second speaker.

Name of voice print	Duration (s)	Number of frames	Mean power (dB)
R_1C	1.857	182	73.96
R_2C	1.857	182	72.92
R_3C	1.950	192	74.4
R_4C	1.997	196	73.19
R_5C	1.904	187	73.59
R_6C	1.997	196	73.07
R_7C	1.811	178	73.86
R_8C	1.904	187	73.98

Table 33: Information of the pre-recorded samples in Catalan of the third speaker.

Name of voice print	Duration (s)	Number of frames	Mean power (dB)
R_1CE	1.765	173	72.63
R_2CE	1.765	173	70.7
R_3CE	1.765	173	74.69

Table 34: Information of the claiming session in Catalan of the third speaker.

Name of voice print	Duration (s)	Number of frames	Mean power (dB)
A_1C	1.811	178	70.45
A _2C	2.182	215	71.01
A _3C	1.579	154	74.71
A _4C	1.579	154	72.57
A _5C	1.811	178	71.85
A _6C	1.625	159	72.21
A _7C	1.718	168	70.18
A _8C	1.671	164	72.2

Table 35: Information of the pre-recorded samples in Catalan of the fourth speaker.

Name of voice print	**Duration (s)**	**Number of frames**	**Mean power (dB)**
A _1CE	2.229	219	70.07
A _2CE	1.811	178	70.41
A _3CE	1.765	173	71.33

Table 36: Information of the claiming session in Catalan of the fourth speaker.

NAME OF VOICEPRINT	P_1E	P_2E	P_3E	**NAME OF VOICEPRINT**	P_1E	P_2E	P_3E
P1	82.48	86.66	83.98	P51	76.64	89.37	82.21
P2	87.38	89.02	92.17	P52	79.81	90.71	84.98
P3	86.31	96.31	92.03	P53	82.89	90.55	85.06
P4	84.51	89.39	85.96	P54	79.27	94.04	86.05
P5	91.18	97.56	95.69	P55	79.53	91.62	82.13
P6	83.69	79.70	84.13	P56	80.80	79.28	83.81
P7	80.63	89.59	85.84	P57	79.75	78.24	83.78
P8	84.11	90.72	86.43	P58	82.42	81.02	80.24
P9	85.62	83.12	90.55	P59	83.66	86.69	86.48
P10	83.43	85.90	90.21	P60	84.15	79.83	85.56
P11	85.62	96.44	92.31	P61	80.36	91.77	85.11
P12	82.39	80.81	89.56	P62	79.87	90.42	83.24
P13	85.11	90.51	86.83	P63	83.08	90.74	86.59
P14	79.62	81.93	81.82	P64	82.80	93.37	84.07
P15	84.50	94.94	86.73	P65	79.65	87.15	84.13
P16	84.07	93.07	87.98	P66	76.54	88.92	80.52
P17	87.08	93.26	90.25	P67	84.38	99.50	93.75
P18	81.31	93.66	87.30	P68	80.45	91.09	84.86
P19	86.06	96.69	90.48	P69	83.40	92.16	86.78
P20	80.70	95.04	85.76	P70	80.04	80.30	81.80
P21	85.94	93.48	90.66	P71	83.60	95.07	86.66
P22	81.27	90.71	83.76	P72	82.41	82.99	85.13
P23	81.70	90.54	83.91	P73	80.31	90.11	85.25

P24	76.35	89.97	82.93	P74	81.05	77.65	81.79
P25	79.85	83.57	83.13	P75	79.36	81.43	82.10
P26	80.74	98.16	88.40	P76	78.01	77.74	80.65
P27	76.01	88.68	83.78	P77	77.60	87.83	79.35
P28	80.06	90.71	88.30	P78	81.10	87.42	84.03
P29	77.76	88.75	86.59	P79	79.49	82.06	83.65
P30	77.19	88.26	86.57	P80	74.26	88.78	83.34
P31	75.82	84.46	81.58	P81	73.63	85.97	82.78
P32	79.05	78.73	84.67	P82	81.20	87.01	85.84
P33	75.28	83.79	82.84	P83	85.46	97.67	91.23
P34	79.47	93.43	87.47	P84	85.34	97.82	88.84
P35	84.44	94.29	88.45	P85	80.54	89.36	84.96
P36	78.35	92.62	88.54	P86	84.06	95.27	88.97
P37	77.78	85.40	83.71	P87	81.88	92.25	85.22
P38	78.26	89.67	84.10	P88	79.12	89.99	81.64
P39	79.29	92.85	85.19	P89	75.66	89.14	84.69
P40	76.92	90.95	83.73	P90	77.96	89.09	86.33
P41	82.01	75.79	81.55	P91	78.75	91.04	81.65
P42	79.72	91.28	81.86	P92	75.14	83.25	78.38
P43	85.66	96.40	90.10	P93	81.43	80.34	82.48
P44	78.44	89.38	85.89	P94	77.87	82.11	86.88
P45	80.31	88.41	86.88	P95	78.37	86.75	86.34
P46	81.46	93.14	85.21	P96	76.99	87.42	83.30
P47	83.77	94.20	85.31	P97	82.12	87.24	84.41
P48	79.48	94.24	85.59	P98	84.96	93.56	88.71
P49	82.72	84.11	83.19	P99	81.26	94.15	84.91
P50	82.09	92.13	88.45	P100	85.84	95.63	86.76

Table 37: Comparisons results of the test done in point 5.3.

GLOSSARY

ACDM: Additive Cepstral Distortion Model
AMFCC: Autocorrelation MFCC
ASV: Automatic Speaker Verification
AT&T: American Telephone & Telegraph
CMN: Cepstral Mean Normalization
DCT: Discrete Cosine Transform
DTW: Dynamic Time Warping
DDTW: Derivative Dynamic Time Warping
EER: Equal Error Rate
GMM: Gaussian Mixture Models
FA: False Acceptance
FFT: Fast Fourier Transform
FR: False Rejection
HMM: Hidden Markov Models
HP: Hewlett-Packard
ITT: International Telephone & Telegraph
LPC: Linear Prediction Coding
MFCC: Mel Frequency Cepstral Coefficients
NIST: National Institute of Standards and Technology
NN: Nearest Neighbors
PDF: Probability Density Function
PLP: Perceptual Linear Predictive
RASTA: Relative Spectral
SBC: Subband based Cepstral Parameters
SMN: Spectrum Mean Normalization
S-NS: Stationary-NonStationary
SNR: Signal Noise Ratio
STFT: Short-Time Fourier Transform
VQ: Vector Quantization
WAV: Waveform Audio Format

REFERENCES

[1] Joseph P. Campbell, “Speaker Recognition: A Tutorial”, 1997.

[2] Naoki Kioya and Kaoru Arakawa, “A method for impact noise-reduction from speech using a stationary-nonstationary separating filter”, 2009.

[3] Tomas Dekens, Werner Verhelst, Fransois Capman, Frédéric Beaugendre, “Improved Speech Recognition in noisy environments by using a throat microphone for accurate voice detection”, 2010.

[4] Eammon J. Keogh and Michael J. Pazzani, “Derivative Dynamic Time Warping”, 2001.

[5] Douglas Reynolds, “Gaussian Mixture Models”, 2001.

[6] Stefan Schacht, Jacques Koreman, Christoph Lauer, Andrew Morris, Dalei Wu and Dietrich Klakow, “Frame based Features”, 2007.

[7] Urmila Shrawankar and Dr. Vilas Thakare, “Feature Extraction for a Speech Recognition System in Noisy Environment”, 2010.

[8] Melvyn J. Hunt, “Spectral Signal Processing for ASR”, 1999.

[9] Md. Rashidul Hasan, Mustafa Jamil, Md. Golam Rabbani and Md. Saifur Rahman, “Speaker Identification using Mel Frequency Cepstral Coefficients”, 2004

[10] Goutam Saha and Ulla S. Yadhunandan, “Modified Mel-Frequency Cepstral Coefficient”, 2004.

[11] Wang Hong and Pan Jin’gui, “Modified MFCCs for robust speaker recognition”, 2010.

[12] Kshitiz Kumar, Chanwoo Kim and Richard M. Stern, “Delta-Spectral Cepstral Coefficients for Robust Speech Recognition”, 2010.

[13] Olivier Rioul and Martin Vetterli, “Wavelets and Signal Processing”, 1991.

[14] Hynek Hermansky, “Perceptual Linear Predictive (PLP) Analysis of Speech”, 1990.

[15] Hynek Hermanksy, “RASTA, Processing of Speech”, 1994.

[16] Ruhi Sarikaya, Bryan L. Pellom and John H. L. Hansen, “Wavelet Packet Transform Features with application to Speaker Identification”, 1998.

[17] R. M. Nickel, S.P. Oswal and A. N. Nyer, “Robust Speaker Verification with Principal Pitch Components”, 2004.

[18] Yosef A. Solewicz and Moshe Koppel, “Selective Fusion for Speaker Verification in Surveillance”, 2005.

[19] Paul Boersma and David Weenink , “Praat: doing phonetics by computer”
http://www.fon.hum.uva.nl/praat/

[20] ”Becars: Library and Tools for Speaker Verification”
http://www.tsi.enst.fr/becars/index.php

[21] Recognition Technologies: http://www.speakerverification.net

[22] “Praat Guide” http://latlcui.unige.ch/phonetique/easyalign/tutorialpraat.pdf

[23] Wei Han, Cheong-Fat Chan, Chiu-Sing Choy and Kong-Pang Pun, “An efficient MFCC extraction method in Speech Recognition”, 2006.