SPEAKER LOCALIZATION AND IDENTIFICATION

April 2012

Hendrik Tómasson
Master of Science in Electrical Engineering

SPEAKER LOCALIZATION AND IDENTIFICATION

Hendrik Tómasson
Master of Science
Electrical Engineering
April 2012
School of Science and Engineering
Reykjavík University

M.Sc. RESEARCH THESIS

ISSN 1670-8539

Speaker localization and identification

by

Hendrik Tómasson

Research thesis submitted to the School of Science and Engineering
at Reykjavík University in partial fulfillment of
the requirements for the degree of
Master of Science in **Electrical Engineering**

April 2012

Research Thesis Committee:

Jón Guðnason, Supervisor
Assistant professor, Reykjavik University

Yngvi Björnsson
Associate professor, Reykjavik University

The undersigned hereby certify that they recommend to the School of Science and Engineering at Reykjavík University for acceptance this research thesis entitled **Speaker localization and identification** submitted by **Hendrik Tómasson** in partial fulfillment of the requirements for the degree of **Master of Science** in **Electrical Engineering**.

_____

Date

_____

Jón Guðnason, Supervisor
Assistant professor, Reykjavik University

_____

Yngvi Björnsson
Associate professor, Reykjavik University

Speaker localization and identification

Hendrik Tómasson

April 2012

Abstract

Recognizing and locating sounds is a crucial part of human awareness and communications. Humanoid robots should be as aware as humans or better and their artificial auditory system should have better speech separation than humans are capable of doing. The humanoids should be able to recognize who is speaking and have the ability to turn their head such that a visual information of that speaker could be obtained.
This project is separated into three parts.
First: speaker recognition is done on the YOHO database comparing three different feature extraction methods:

1. Mel frequency cepstrum coefficients (MFCC)
2. Reversed mel frequency cepstrum coefficients (RMFCC)
3. MFCC on voice source obtained with Iterative adaptive inverse filtering (IAIF)

Each of the features are trained using Gaussian mixture models (GMM). The misclassification rate for each of the methods were found to be: 10.13% for MFCC, 30.96% for RMFCC and 62.04% for IAIF. Also by mixing MFCC and RMFCC methods the traditional MFCC method is improved by 13% and a misclassification rate of 8.81% is obtained.
Second: the locations and speaker identification for a new database which was recorded with a Kinect sensor are estimated. Generalized cross correlation with phase transform (GCC-PHAT) for time difference estimation was used to locate speakers, and MFCC using GMM were used to recognize the speakers. A misclassification rate of 24.67% was obtained and a location accuracy of $3.09^\circ \pm 3.92^\circ$ without windowing.
Third: the misclassification rate and localization errors as a function of window size are estimated for the new database and the real time behaviour of the speaker recognition and localization methods obtained.

Staðsetning og greining hljóðgjafa

Hendrik Tómasson

Apríl 2012

Útdráttur

Hljóðgreining nýtist fólki ýmist í samskiptum eða til að bregðast við áreiti úr umhverfinu. Vélmenni ættu að vera jafn góð ef ekki betri en menn í því að greina umhverfi sitt. Einnig ættu þau að geta aðgreint hljóð betur en menn. Vélmenni ættu að geta greint staðsetningu hljóðgjafa, snúið höfðinu og fengið sýnilegar upplýsingar um hljóðgjafann. Þessu verkefni er skipt niður i þrjá hluta.
Fyrsti hluti: er hljóðgreining gerð á YOHO gagnagrunninn fyrir þrjár mismunandi aðferðir til séreinkenna öflunar.

1. Mel cepstum framsetning (MFCC).
2. Öfug Mel cepstrum framsetning (RMFCC).
3. Mel cepstrum framsetning á raddlind sem fundin er með "iterative adaptive inverting filtering" (IAIF).

Blönduð Gássísk líkön eru notuð á séreinkenna vigrana í öllum tilvikum. Villuhlutfall var 10.13% fyrir MFCC, 30.96% fyrir RMFCC og 62.04% fyrir IAIF. Með því að blanda saman MFCC og RMFCC þá er villuhlutfallið lækkað í 8.81%.
Annar hluti: felst í því að greina og finna staðsetningar hljóðgjafa fyrir nýjan gagnagrunn sem er tekinn upp með Kinect skynjara. Tímamunur er áætlaður með GCC-PHAT og samsvarandi snúningur metinn. Fyrir hljóðgreiningu er notað MFCC einkenni og Gássisk líkön, villuhlutfall var 24.67%. Fyrir staðsetningar þá var meðaltals villa $3.09^\circ \pm 3.92^\circ$ án þess að glugga merki.
Þriðji hluti: fjallar svo um rauntímagreiningu hljóðgjafa og staðsetning þeirra. Villuhlutfall og meðaltals staðsetninga skekkja sem fall af gluggastærð er sýnd.

Contents

List of Figures **vii**

List of Tables **ix**

1 Introduction **1**

2 Speaker Identification **3**

2.1 Theory 3

2.1.1 Mel Frequency Cepstrum Coefficients (MFCC) 3

2.1.2 Reversed Mel Frequency Cepstrum Coefficients (RMFCC) 5

2.1.3 Iterative Adaptive Inverting Filter (IAIF) 6

2.1.4 Gaussian Mixture Models (GMM) 6

2.1.5 Combinations 9

2.2 Implementation 9

2.2.1 Steps For Speaker Identification 9

2.2.2 Database Setup And Kinect Specifications 10

2.3 Speaker Identification Results 12

2.3.1 Speaker Identification Of The YOHO Database 12

2.3.2 Combinations 15

2.3.3 Speaker Identification Of The SiriusV310 Database 22

2.4 Summary 25

3 Speaker Localization **27**

3.1 Theory 28

3.1.1 Generalized Cross Correlation with Phase Transform (GCC-PHAT) 28

3.1.2 Sound Source Localization 29

3.1.3 Delay And Sum Beamforming 33

3.2 Implementation 34

3.3 Speaker Localization Results 34

3.4 Summary 37

4 Real-Time Simultaneous Speaker Identification and Localization 39
4.1 Implementation 39
4.1.1 Preparation 39
4.1.2 Identification And Localization 40
4.2 Results 41
4.2.1 Speaker Identification 41
4.2.2 Real-Time Speaker Localization 41

5 Summary Of Results 43
5.1 Speaker Identification 43
5.2 Speaker Localization 44

6 Conclusions, Discussion And Future Work 45
6.1 Conclusions 45
6.2 Discussion 46
6.3 Future Work 47

A Appendix 51
A.1 Speaker identification configurations 51
A.1.1 MFCC and RMFCC configurations 51
A.1.2 IAIF configurations 52
A.1.3 GMM configurations 52
A.2 Database descriptions 53
A.3 Computer specifications 53

List of Figures

2.1 Mel frequency filterbank. 4
2.2 Mel frequency cepstrum coefficients. 5
2.3 MFCC with delta and delta delta coefficients. 5
2.4 Reversed mel frequency cepstrum filterbank. 6
2.5 Feature extraction of voice source. 6
2.6 Three contours. 8
2.7 Marginal probability contour. 8
2.8 Distribution surface. 9
2.9 Kinect sensor. 10
2.10 Distance between Kinect microphones. 11
2.11 Measure speaker locations. 11
2.12 Confusion matrix for MFCC features. 12
2.13 Confusion matrix for RMFCC features. 14
2.14 Confusion matrix for IAIF features. 15
2.15 Misclassification rate as function of weighting for combination of MFCC and RMFCC. 16
2.16 Misclassification rate as function of weighting. 17
2.17 Misclassification rate as function of weighting. 18
2.18 Combination of MFCC, RMFCC and 0% IAIF. 19
2.19 Combination of MFCC, RMFCC and 10% IAIF. 19
2.20 Combination of MFCC, RMFCC and 20% IAIF. 20
2.21 Combination of MFCC, RMFCC and 30% IAIF. 20
2.22 Combination of MFCC, RMFCC and 40% IAIF. 21
2.23 Confusion matrix using MFCC features. 22
2.24 Speakers 101, 107 and 108 comparison. 23
2.25 Confusion matrix using RMFCC features. 24
2.26 Misclassification as a function of weighting for combination of MFCC and RMFCC. 25

3.1 Speaker localization. 29
3.2 Difference between phase shifted waves. 30
3.3 The precision of GCC-PHAT for 16 KHz sampling frequency. 31
3.4 Arc sine. 32
3.5 The precision with cross structure. 33
3.6 Localization of speakers of the SiriusV310 database using mic 1 and 4. . . 34
3.7 Localization of speakers of the SiriusV310 database using mic 1 and 2. . . 35
3.8 Angular errors between measured and estimated for microphones 1 and 4. 35
3.9 Angular errors between measured and estimated for microphones 1 and 2. 36
3.10 Confusion matrix using mfcc features on beamformed data. 37

4.1 Typical signal. 40
4.2 Misclassification as a function of identification period. 41
4.3 Mean errors and standard deviation. 42

6.1 Eigenmike by MH acoustics (www.mhacoustics.com). 47

List of Tables

2.1 Results from TIMIT, NTIMIT, Switchboard from Reynolds [8] and YOHO. 13

3.1 Summary of average errors, maximum errors and error standard deviation for localization without windowing. 36

5.1 Speaker identification summary for the YOHO and SiriusV310 database. . 43
5.2 Lowest misclassification rate of combined methods for the YOHO database. 43
5.3 Accuracy of the speaker localization methods. Summary of errors. 44

A.1 MFCC and RMFCC configurations for the YOHO database 51
A.2 MFCC and RMFCC configurations for the SiriusV310 database 52
A.3 IAIF configurations 52
A.4 GMM configurations 52
A.5 MFCC and RMFCC configurations for the YOHO database 53

Chapter 1

Introduction

Hearing is one of the greatest human senses and is crucial for awareness and communications. When people meet they hear each others voices and respond to what the hear. For example, if a speaker calls a person's name from behind then the person will typically turn towards the source of the sound. If the person has heard that speaker before then a model of that speaker has been made and the person recognizes the speaker. At the same time the person hears the music on the radio and the air conditioning. The person is able to recognize which sound is which, locate where each sound is coming from and recognize which sounds can be ignored, all simultaneously.

Mimicing the behaviour of human hearing in a computer or a robot is interesting. Humanoid robots should have the capability to distinguish between different speakers and sounds coming from different direction and be even more aware of the environment than humans. They should be able to turn their head and seek visual information of interesting speakers and respond to all information obtained from and about each speaker.

An artificial auditory system could also be used to help deaf people locating important sound sources around them, for example, ambulances or fire alarms. If such a system would be made for deaf people a speech separation application should also be implemented giving them the ability to "listen" to multiple speakers.

These are all very difficult topics for machine learning and the main issue is the noise which humans are able to filter out. Humans are able to do these things with two ears but a humanoid robot could have many ears. Computers can recognize fairly well what is said around them but if two or more people speak at the same time the speech recognition becomes harder. Even for a human it can be difficult to recognize what two or more people say at the same time. Humanoid robots can not respond to human interaction unless the robot knows what is said, what it means, who said it and how to respond.

The aim of this project is to develop and experiment with computerized methods to recognize who is speaking and where the speakers are located. This project is separated into three different parts.

The YOHO database is used for **speaker identification** and different methods of feature extraction are compared. The methods are *mel frequency cepstrum coefficients* (MFCC), *reversed mel frequency cepstum coefficients* (RMFCC) and MFCC used on the voice source which is obtained by *iterative adaptive inverting filtering* (IAIF) the sound signal. The combinations of these feature extraction methods is then used to improve the traditional MFCC method.

Speaker localization is performed on a new database, made with Icelandic sentences in a closed office room. The locations of each speaker is estimated and the accuracy of the localization. Also an identification by beamforming using previous locations is obtained.

A **real-time simultaneous speaker identification and localization** is performed, by windowing the new database and the corresponding localization errors and misclassification rate as a function of window size (identification period) are obtained.

Chapter 2

Speaker Identification

Humans recognize sounds because they have heard them before or because they have heard similar sounds. To give a computer the ability to recognize sound the characteristics for that sound needs to be identified. To find these characteristics a feature extraction is applied to the sound waves and a feature vector is made. This feature vector is like the fingerprints of the sound, some kind of data which tells the difference between person A or B. These fingerprints are then used to make a model for each speaker, which is the basis for speaker recognition. In this chapter, three different feature extraction methods will be described. The model and the steps for speaker identification are explained and experimental results presented for the methods.

2.1 Theory

2.1.1 Mel Frequency Cepstrum Coefficients (MFCC)

For speech and speaker recognition the most common way to find the features of each utterance is the mel frequency cepstral coefficents (MFCC). This feature extraction is an approximation to the human hearing system. The first step in obtaining MFCC is spectrum analysis of a small window of speech. The next step is to apply a mel-scaled filterbank to the spectrum. The mel filterbank is linear for lower frequencies but logarithmic at higher frequencies. This means that there is more information at lower frequencies than higher ones. The mel frequency scale is given by [6]

$$f_{mel} = 2595 \cdot log_{10}\left(1 + \frac{f_{linear}}{700}\right). \tag{2.1}$$

The number of filters is typically between 22 - 29. To compute the mel frequency cepstrum coefficients the following procedure was used [4]. First the speech signal is windowed using Hamming window to 30 ms parts. The window is then Fourier transformed and the magnitude for the new spectrum found. A logarithm is applied to each energy and the frequencies warped according to the mel scale in Equation 2.1. An inverse Fourier transform is applied to the log-energies and the MFCC coefficient are obtained. Figure 2.1 shows the MFCC filterbank.

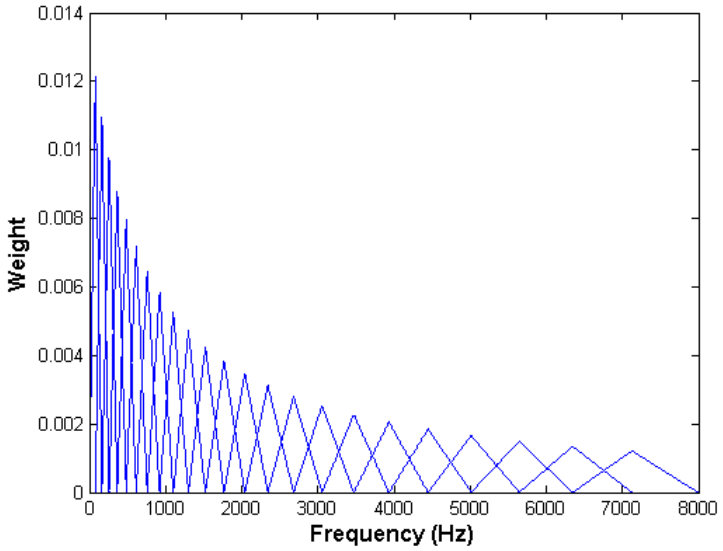

Figure 2.1: Mel frequency filterbank.

Often, delta and delta-delta coefficients are included for more precision. That means that the first and second derivative of MFCC coefficients are added to them. This project used delta and delta-delta coefficients whenever the MFCC and reversed mel frequency cepstrum coefficents (RMFCC) are used. The RMFCC will be explained next. A typical picture of MFCC coefficients with and without deltas and delta deltas can be seen in Figures 2.2 and 2.3, respectively.

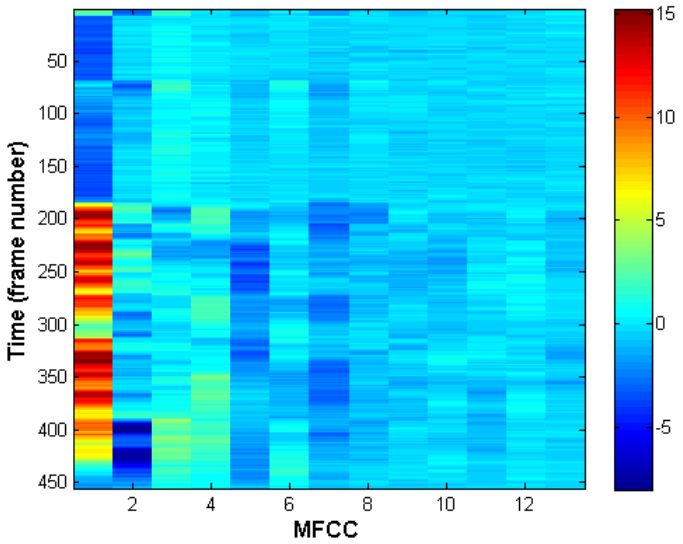

Figure 2.2: Mel frequency cepstrum coefficients.

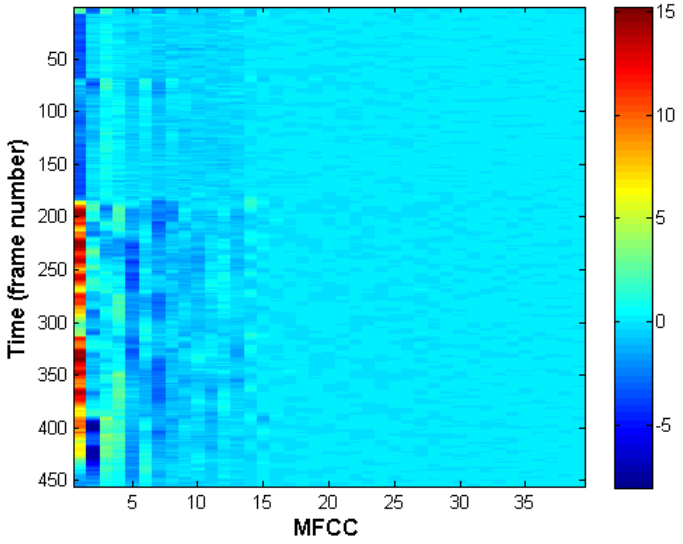

Figure 2.3: MFCC with delta and delta delta coefficients.

2.1.2 Reversed Mel Frequency Cepstrum Coefficients (RMFCC)

In the reversed mel frequency cepstrum coefficients (RMFCC) the filterbank is flipped about its center frequency. When the sampling frequency is 16kHz the center frequency would be 8000 Hz and the filterbank would scale linearly from 5000 Hz to 8000 Hz but logarithmically in the lower frequencies [6]. For speaker recognition it is thought that the mid to upper frequencies give more speaker characteristics than the lower ones [6]. It has

also been shown that non-uniform frequency features perform better than uniform ones [6]. The reversed mel frequency filterbank can be seen in Figure 2.4.

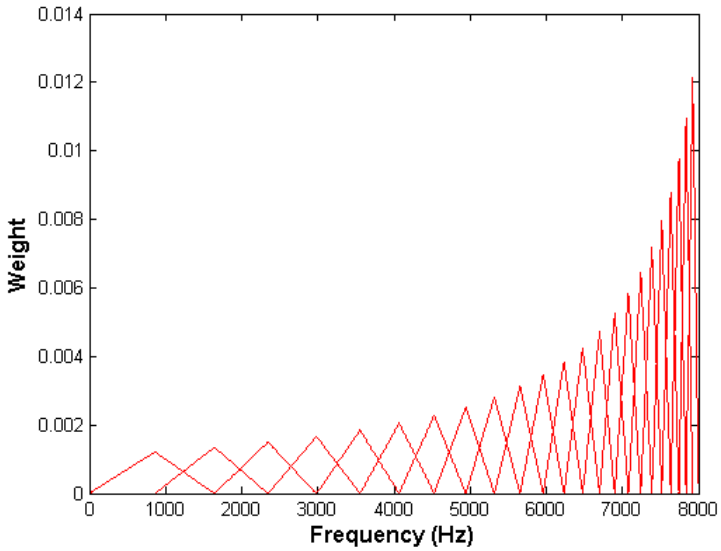

Figure 2.4: Reversed mel frequency cepstrum filterbank.

2.1.3 Iterative Adaptive Inverting Filter (IAIF)

Iterative adaptive inverting filtering (IAIF) extracts the voice source signal by inverse filtering the vocal tract from the speech signal. When the voice source has been extracted the mel frequency cepstrum coefficients are used for feature extraction. Figure 2.5 shows the feature extraction procedure. The aim is to extract the features from the vocal tract not the speech itself. The implementation of IAIF can be seen in [1].

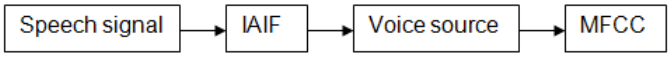

Figure 2.5: Feature extraction of voice source.

2.1.4 Gaussian Mixture Models (GMM)

Normal distribution (Gaussian) is often used to model natural phenomena. There are many different possibilities for model training but Gaussian mixture models (GMM) have proven to work well for speaker identification [8]. The advantages of using GMM include that they are computationally inexpensive and the Gaussian distribution is a well understood model in statistics [9].

The Gaussian distribution is written in the form:

$$N(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} exp\Big\{ -\frac{1}{2\sigma^2}(x-\mu)^2 \Big\}. \tag{2.2}$$

Where μ is the mean and σ^2 is the variance. If the $\mathbf{x}$ vector is a D dimensional vector then the form of the multivariate Gaussian distribution becomes,

$$N(\mathbf{x}|\mu,\mathbf{\Sigma}) = \frac{1}{(2\pi)^{D/2}\mathbf{\Sigma}^{1/2}} exp\Big\{ -\frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu) \Big\}, \tag{2.3}$$

where μ is a multidimensional mean vector of size D and $\mathbf{\Sigma}$ is a covariance matrix of size $D \times D$. The part in the exponential is called the Mahalanobis distance between $\mathbf{x}$ and μ [4]. The Gaussian mixture distribution is [2]:

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^{K} w_i N(\mathbf{x}|\mu_i, \mathbf{\Sigma}_i). \tag{2.4}$$

The mixture distribution is a sum of K Gaussian densities. Each mixture has its own mean vector μ_i, covariance matrix $\mathbf{\Sigma_i}$ and mixing weights w_i. The weights satisfy the constraint $\sum_{i=1}^{K} w_i = 1$ and $\lambda = (w_i, \mu_i, \Sigma_i)$ where i = 1,... , K . The log of the likelihood is defined as [2],

$$lnp(\mathbf{x}|\mathbf{w},\mu,\mathbf{\Sigma}) = \sum_{n=1}^{N} ln\Big\{ \sum_{i=1}^{K} w_i N(\mathbf{x}_n|\mu_i, \mathbf{\Sigma}_i) \Big\}. \tag{2.5}$$

For example, the data can sometimes be modelled by single Gaussian which is a Gaussian mixture model with one mixture. But that is not the case in most applications. Sometimes the data has many local maxima as can be seen in Figure 2.6, which shows three contours, each corresponding to a density of mixture component. Each of the distributions in the mixture has its own mean vector μ and covariance matrix $\mathbf{\Sigma}$.

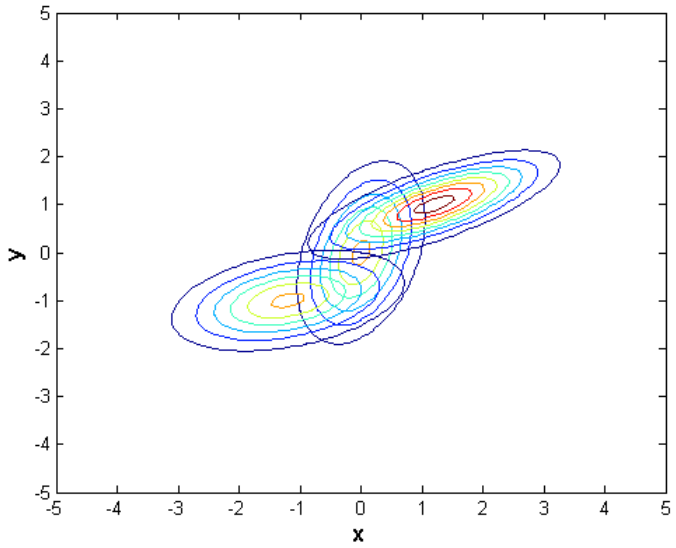

Figure 2.6: Three contours.

The marginal probability density of the combined mixtures can be seen in Figure 2.7. This probability density would be calculated with Equation 2.4. Each of the mixture is a Gaussian distribution.

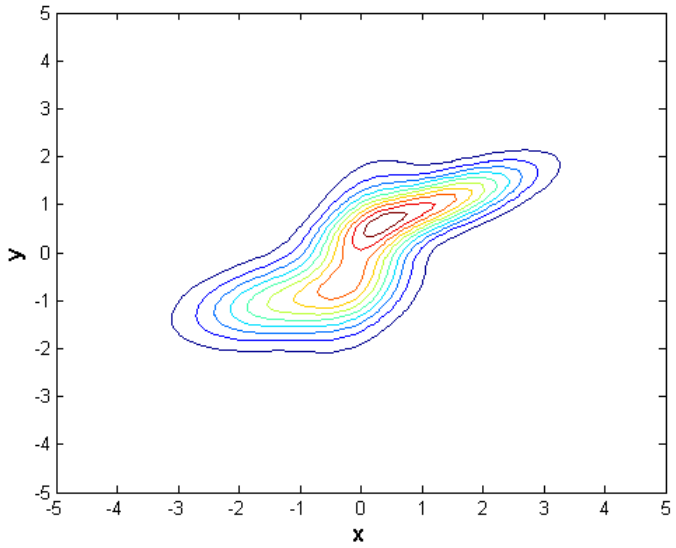

Figure 2.7: Marginal probability contour.

The corresponding distribution surface is shown in Figure 2.8,

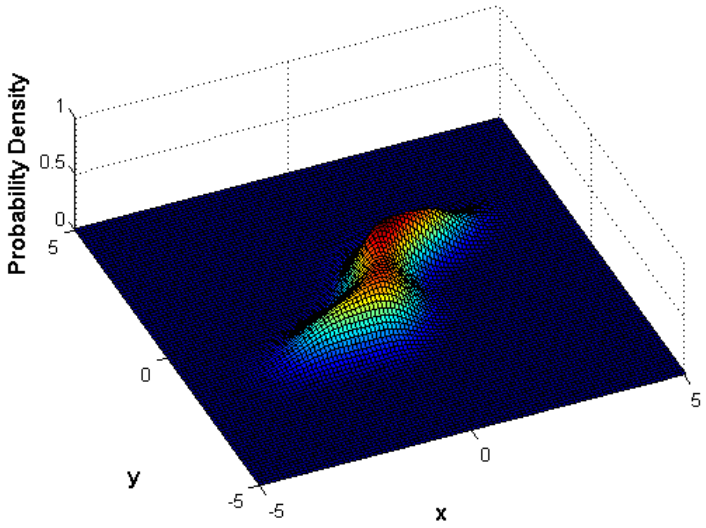

Figure 2.8: Distribution surface.

The combined surface still has similar abilities as a single Gaussian distribution, that is, the further away from the means the lower probability is obtained. The method used to find the maximum likelihood solution for models is called expectation-maximization algorithm (EM algorithm) [5].

2.1.5 Combinations

One idea to combine different likelihood models is to use weighting. The form of the weighting function is:

$$L = wL1 + (1 - w) \cdot L2 \tag{2.6}$$

where L is the final likelihood, L1 is a likelihood vector of some method, L2 is a likelihood vector from another method and w is the weight number. The aim of the weighting function (2.6) is to check if improvements can be made by combining different methods.

2.2 Implementation

2.2.1 Steps For Speaker Identification

The following steps are used for speaker identification.

1. **Database obtained.** The data base is separated into two groups. ENROLL and VERIFY. ENROLL is used for training. VERIFY is used to test the recognizer.

2. **Configuration files loaded.** The configuration files contain all configurations of file locations and parameters for all the methods used in the whole process.

3. **File lists made.** The locations of all features are stored in the feature list. Similarly, the locations for the speech files are stored in the speech list.

4. **Feature extraction.** The Feature extraction code runs on the speech list and for each utterance found there, a feature file is made and saved in a location from the feature list. In this project the feature types are either a MFCC, RMFCC or MFCC on voice source estimated with IAIF.

5. **Model training.** Mean and covariance is found for each mixture for each person in the ENROLL folder and the corresponding model made from that values. Expectation maximization (EM) algorithm is used for training [5]. The Configuration file stores the folder locations for the models.

6. **Log likelihoods obtained.** Log likelihoods of each utterance from the VERIFY folder corresponding to the models found with GMM are obtained using Equation 2.5. The likelihood vector for each speaker is stored corresponding to location given by the configuration file.

7. **Confusion matrix and misclassification rate found.** The highest likelihood from the likelihood folder and the corresponding model for that likelihood value is found. The person is classified as that model and the corresponding column value increases by one in the confusion matrix. The misclassification rate is the number of classifications which are not on the diagonal of the confusion matrix divided by the whole sum of the confusion matrix.

2.2.2 Database Setup And Kinect Specifications

A database was made to test speaker localization and identification simultaneously. The database consists of five males and five females. Five locations were used and for each location ten sentences were spoken by each speaker.

The Kinect sensor shown in Figure 2.9 was used because Microsoft has given an open source code for the Kinect audio system.

Figure 2.9: Kinect sensor.

The distance between the microphones of the Kinect sensor can be seen in Figure 2.10. Note that the microphone to the left will be called microphone 1 and the rest will be called microphone 2, 3 and 4. Also note that the microphone combination of mic 1 and 3 is not used because the sample difference will be the same as either mic 1 and 4 or mic 1 and 2 combinations.

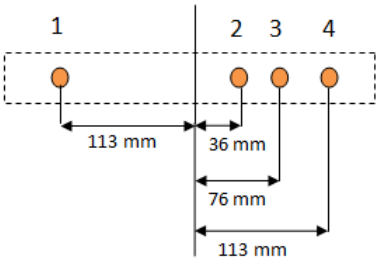

Figure 2.10: Distance between Kinect microphones.

The measured locations for the speakers can be seen in Figure 2.11.

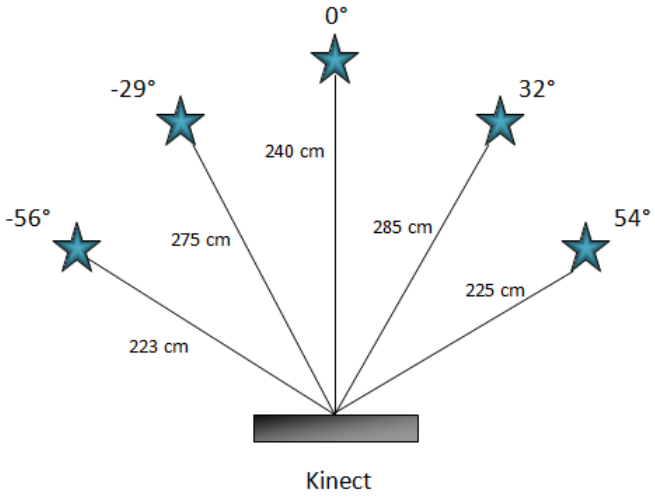

Figure 2.11: Measure speaker locations.

The measurements of the locations are not perfect and the measurement error could be few degrees. The recordings were made in an office room called Sirius and is numbered V310 so the name of the database was given SiriusV310. The ten sentences were put into two groups, ENROLL and VERIFY. The ENROLL data is used for model constructing and the VERIFY data is used to test the models. Seven sentences are used as the ENROLL data and three sentences are used as the VERIFY data. Two of these three sentences are the same for each speaker and the rest is unique. The reason why the speakers in the VERIFY folder are given the same sentences is because the aim is to recognize the speakers, not the speech.

2.3 Speaker Identification Results

2.3.1 Speaker Identification Of The YOHO Database

For the YOHO database a few different feature extraction methods were used as the platform for for GMM model constructing. RMFCC is the same as MFCC except the filterbank is flipped and IAIF is a method of extracting the voice source which is feature extracted with MFCC. To visualize the results a confusion matrix is generated. A confusion matrix tells how many utterances from the VERIFY folder are classified as which person. Each persons utterance is compared to all the models and the highest likelihood indicates which person is predicted. For example when the the actual speaker is 101 then we want it to be predicted as speaker 101 not someone else. For perfect classification then the diagonal would be the only one containing numbers.
For the YOHO database the number of utterances per speaker for classification are 40. That means than the highest value in the confusion matrix in each row can be maximum 40 and the sum of the numbers in each row is 40. Note that all configurations for GMM and feature extraction can be seen in Appendix 1.

MFCC

Figure 2.12 shows the confusion matrix obtained when MFCC is used for feature extraction and GMM for modelling

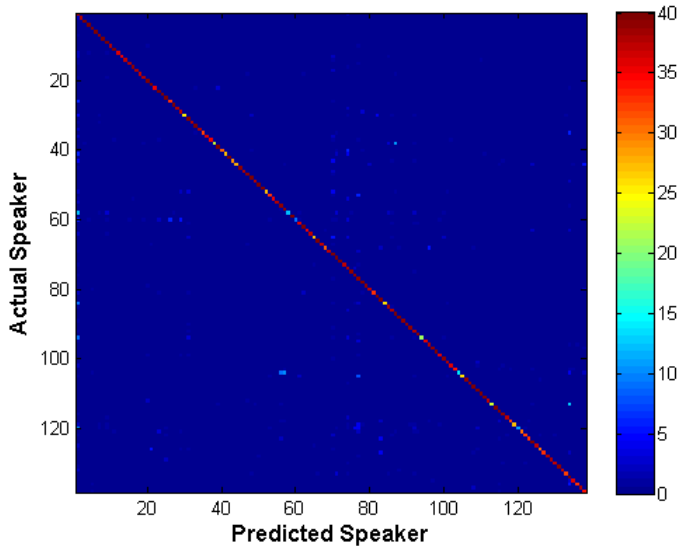

Figure 2.12: Confusion matrix for MFCC features.

As can clearly be seen in Figure 2.12 the diagonal has the highest values. The misclassification rate $MisC$ is defined as,

$$MisC = 1 - \frac{\sum_n^N D(n)}{\sum_j^N \sum_i^N C_m(i,j)} \tag{2.7}$$

N is the number of speakers, $D(n)$ is a diagonal value and $C_m(i,j)$ is confusion matrix value. The misclassification rate for MFCC feature extraction was estimated: $10.13\% \pm 0.41\%$.

By comparing these results to the results which Reynolds [8] found from the TIMIT, NTIMIT and Switchboard there is an indication that the code is working properly.
Table 2.1 shows the results from Reynolds [8] and the accuracy for the YOHO database. Note that information about all the databases can be seen in Appendix 2 and their differences.

Table 2.1: Results from TIMIT, NTIMIT, Switchboard from Reynolds [8] and YOHO.

Database	Accuracy
TIMIT	99.5%
NTIMIT	60.7%
Switchboard	82.8%
YOHO	89.87%

RMFCC

Figure 2.13 contains the confusion matrix obtained when the YOHO database was feature extracted with RMFCC, as proposed by Tashev et al [6], and trained using GMM.

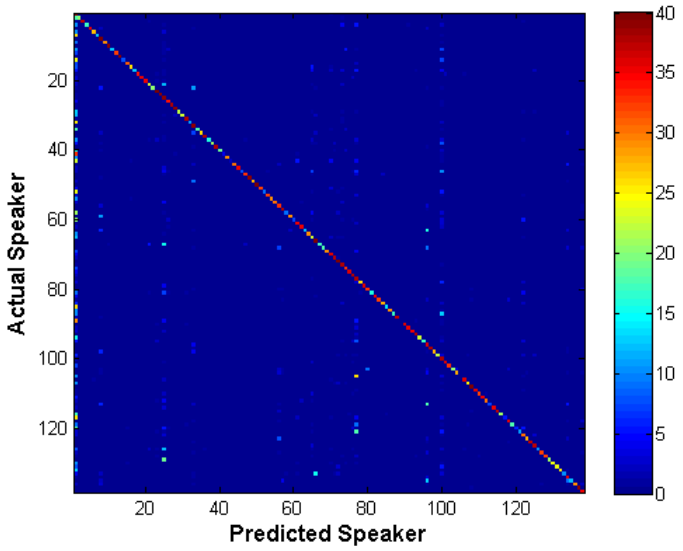

Figure 2.13: Confusion matrix for RMFCC features.

The diagonal in Figure 2.13 is not as strong as in Figure 2.12 but it's still much stronger than the non-diagonal. The misclassification rate for RMFCC feature extraction was estimated: $30.96\% \pm 0.62\%$. The RMFCC misclassification rate is much higher than the MFCC misclassification rate.

MFCC on voice source obtained by IAIF

Figure 2.14 contains the confusion matrix estimated when MFCC feature extraction was performed on the voice source obtained with IAIF on the speech.

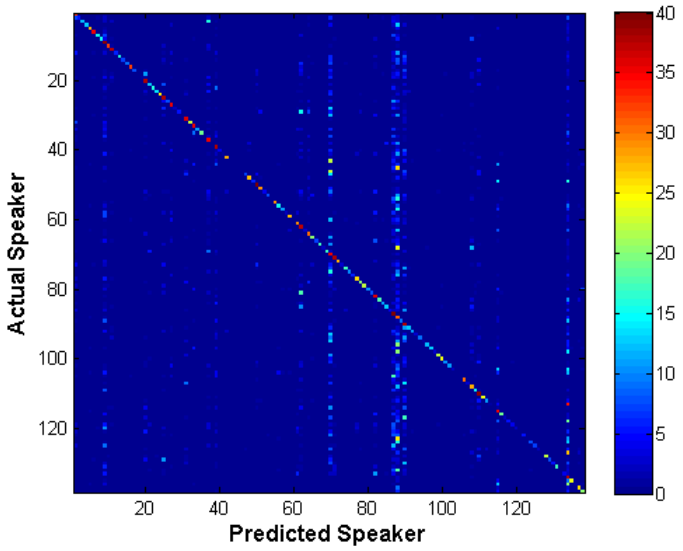

Figure 2.14: Confusion matrix for IAIF features.

The diagonal in Figure 2.14 is not as strong as for Figures 2.12 and 2.13. The misclassification of the confusion matrix in Figure 2.14 was estimated: $62.04\% \pm 0.65\%$. MFCC on the voice source obtained by IAIF gives much worse results than using RMFCC or MFCC on the signal it self.

2.3.2 Combinations

MFCC combined with RMFCC

Figure 2.15 shows the misclassification rate as function of the weighting when combining MFCC and RMFCC.

$$L_{final} = wL_{MFCC} + (1 - w) \cdot L_{RMFCC} \tag{2.8}$$

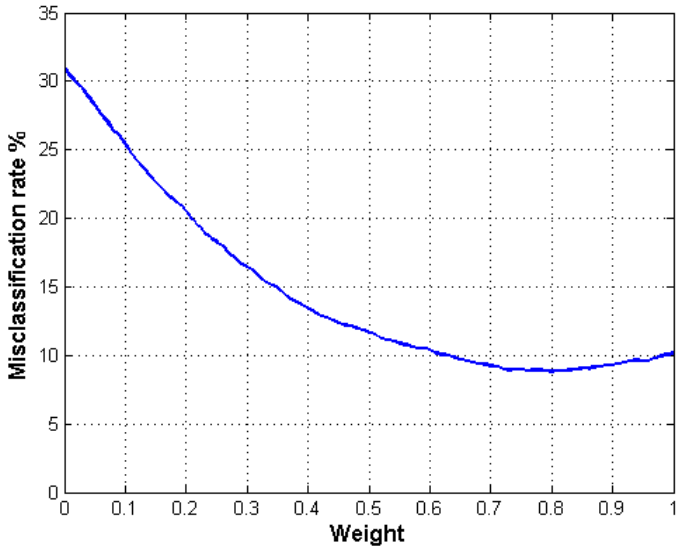

Figure 2.15: Misclassification rate as function of weighting for combination of MFCC and RMFCC.

As can be seen on Figure 2.15 the combination of MFCC and RMFCC does offer an improvement. The minimum misclassification rate obtained using the weighting as seen on Figure 2.15 is 8.81%, which occurs when the weighting ratio is 81% MFCC likelihoods and 19% RMFCC likelihoods. This is an improvement over the 10.13% misclassification rate when MFCC is used alone, or an 13.06% improvement.

MFCC combined with IAIF for voice source estimation

MFCC on the voice source obtained using IAIF could offer an improvement when MFCC is used on the speech directly. Figure 2.16 shows the misclassification rate as a function of weight for the implementation in Equation 2.9.

$$L_{final} = wL_{MFCC} + (1 - w) \cdot L_{IAIF} \tag{2.9}$$

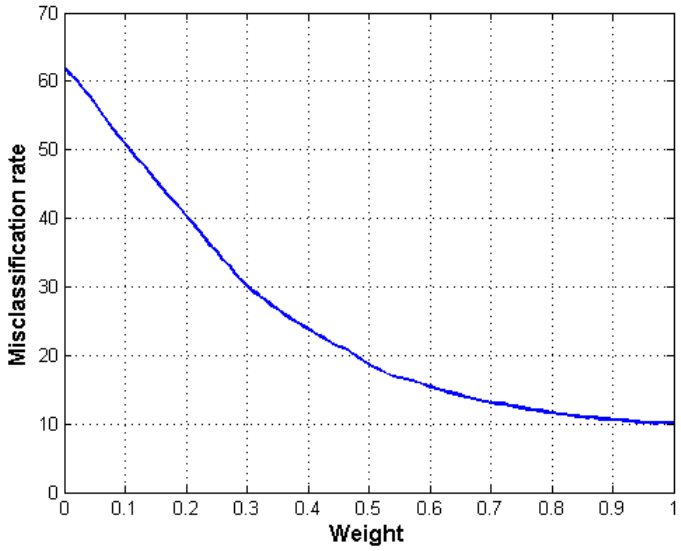

Figure 2.16: Misclassification rate as function of weighting.

The misclassification rate does not improve which is an indication that IAIF does not offer any extra information over what MFCC does.

RMFCC combined with IAIF

IAIF did not improve MFCC but RMFCC did. So the combination of RMFCC and MFCC on the voice source obtained with IAIF should also be checked. Figure 2.17 shows the misclassification rate as function of weighting for the combination of RMFCC and IAIF likelihoods.

$$L_{final} = wL_{RMFCC} + (1 - w) \cdot L_{IAIF} \tag{2.10}$$

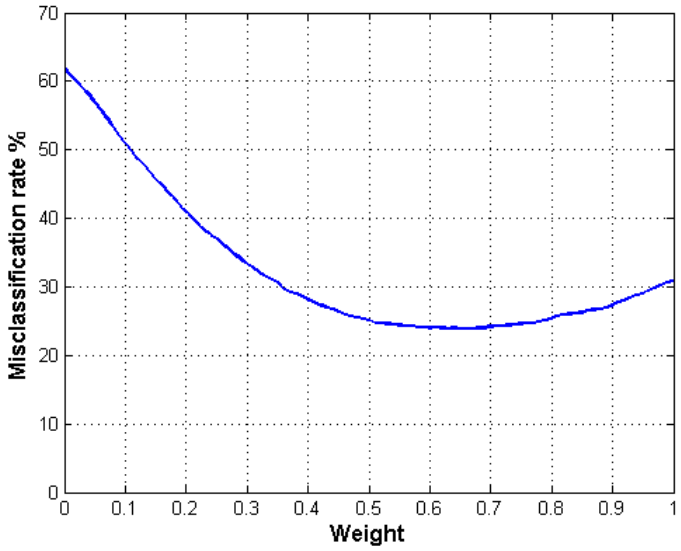

Figure 2.17: Misclassification rate as function of weighting.

As can be seen in Figure 2.17 the combination of RMFCC and IAIF improves RMFCC. Weighting factor of 0.62 gives best result for this combination the misclassificaton of 23.86%, which is far away from the quality which the MFCC/RMFCC combination give.

Combination of MFCC, RMFCC and MFCC on voice source (IAIF)

Using the knowledge from before that combination of RMFCC and MFCC made an improvement and that IAIF improves RMFCC a final case should be on the form

$$L = w_1 \cdot L_{MFCC} + w_2 \cdot L_{RFCC} + w_3 \cdot L_{IAIF} \tag{2.11}$$

Figures 2.18, 2.19, 2.20, 2.21 and 2.22 show the misclassification rate as function of MFCC combined with RMFCC. The legend for each pictures indicates the percentage of MFCC likelihoods for each line. Each picture corresponds to some percentage of IAIF likelihoods. The jumps between measurements are 10%, which is quite high, but the computation time to obtain the results were around 50 hours on computer 1 in appendix 2.

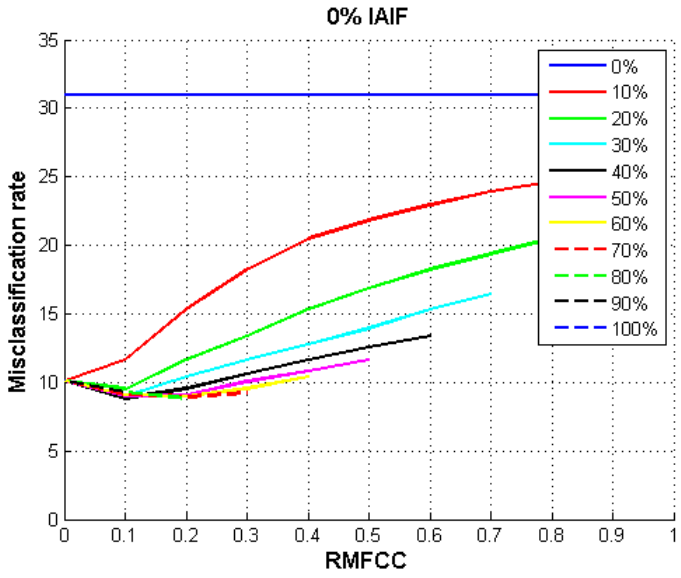

Figure 2.18: Combination of MFCC, RMFCC and 0% IAIF.

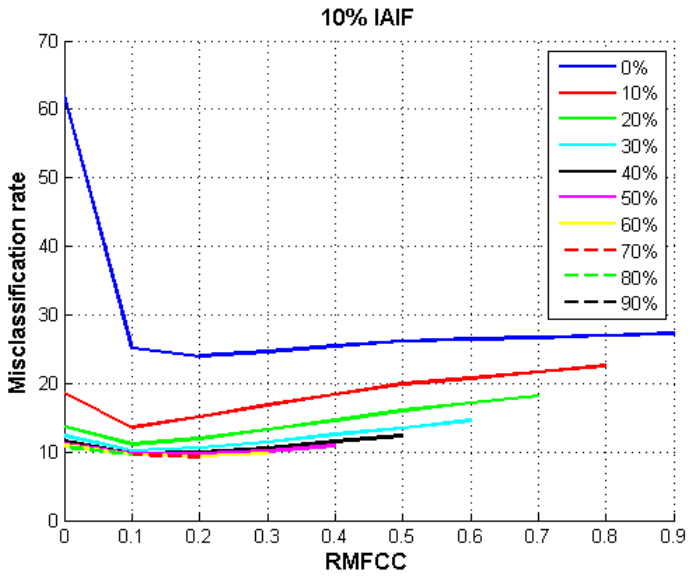

Figure 2.19: Combination of MFCC, RMFCC and 10% IAIF.

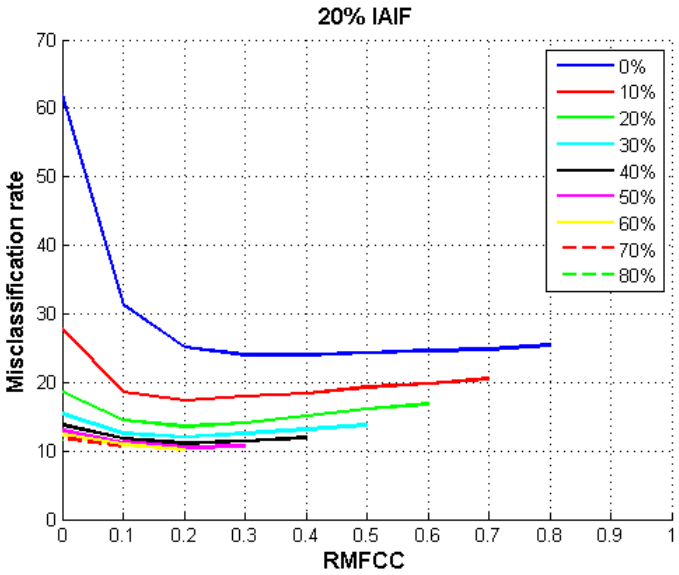

Figure 2.20: Combination of MFCC, RMFCC and 20% IAIF.

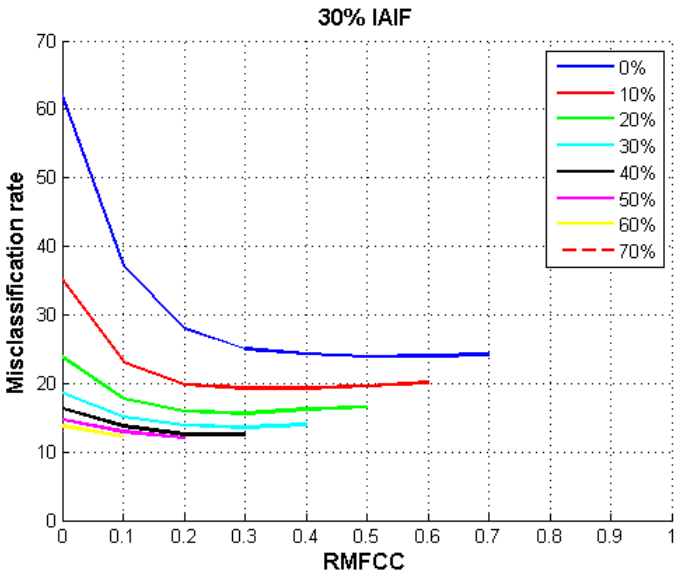

Figure 2.21: Combination of MFCC, RMFCC and 30% IAIF.

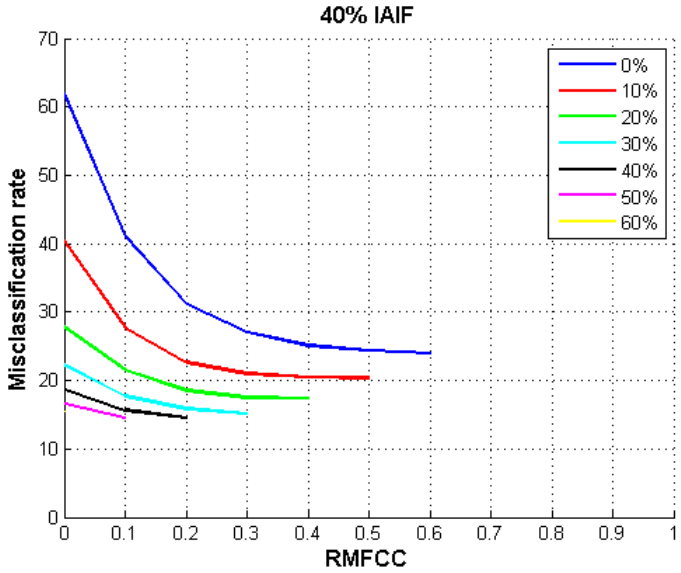

Figure 2.22: Combination of MFCC, RMFCC and 40% IAIF.

The lowest misclassification rate obtained are 8.81% for the two combinations $0.8\%L_{MFCC}+0.2\%L_{RMCC}+0\%L_{IAIF}$. . These results indicates that IAIF does not improve MFCC or RMFCC in any way for speaker recognition.

2.3.3 Speaker Identification Of The SiriusV310 Database

The basis for successful identification of the SiriusV310 database are the results from the YOHO database. The results from the YOHO database is a knowledge that method and its implementation are working properly. The same setup is used for the SiriusV310 and the YOHO database for feature extraction and model training. The frame size for feature extraction increases in samples for higher sampling frequency but the window size is of the same size.

MFCC for SiriusV310

The confusion matrix for MFCC feature extraction of the SiriusV310 database can be seen in Figure 2.23. The SiriusV310 database is different from the YOHO database in that there are maximum of 15 utterances which can be classified for each speaker.

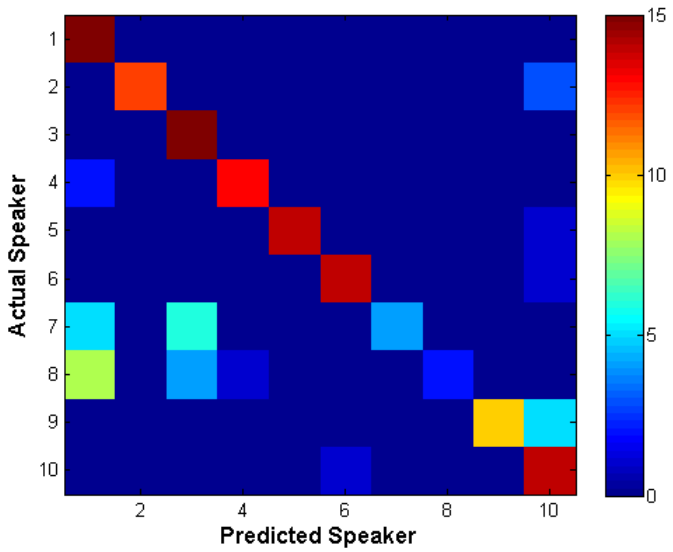

Figure 2.23: Confusion matrix using MFCC features.

The test set misclassification rate for the SiriusV310 database was found to be $24.67\% \pm 3.52\%$ which is a larger misclassification rate than the YOHO database gave.
One of the reason for the high misclassification is that there is considerable confusion between speakers 101, 107 and 108. Figure 2.24 shows the same sentence said by the three speakers 101,107 and 108.

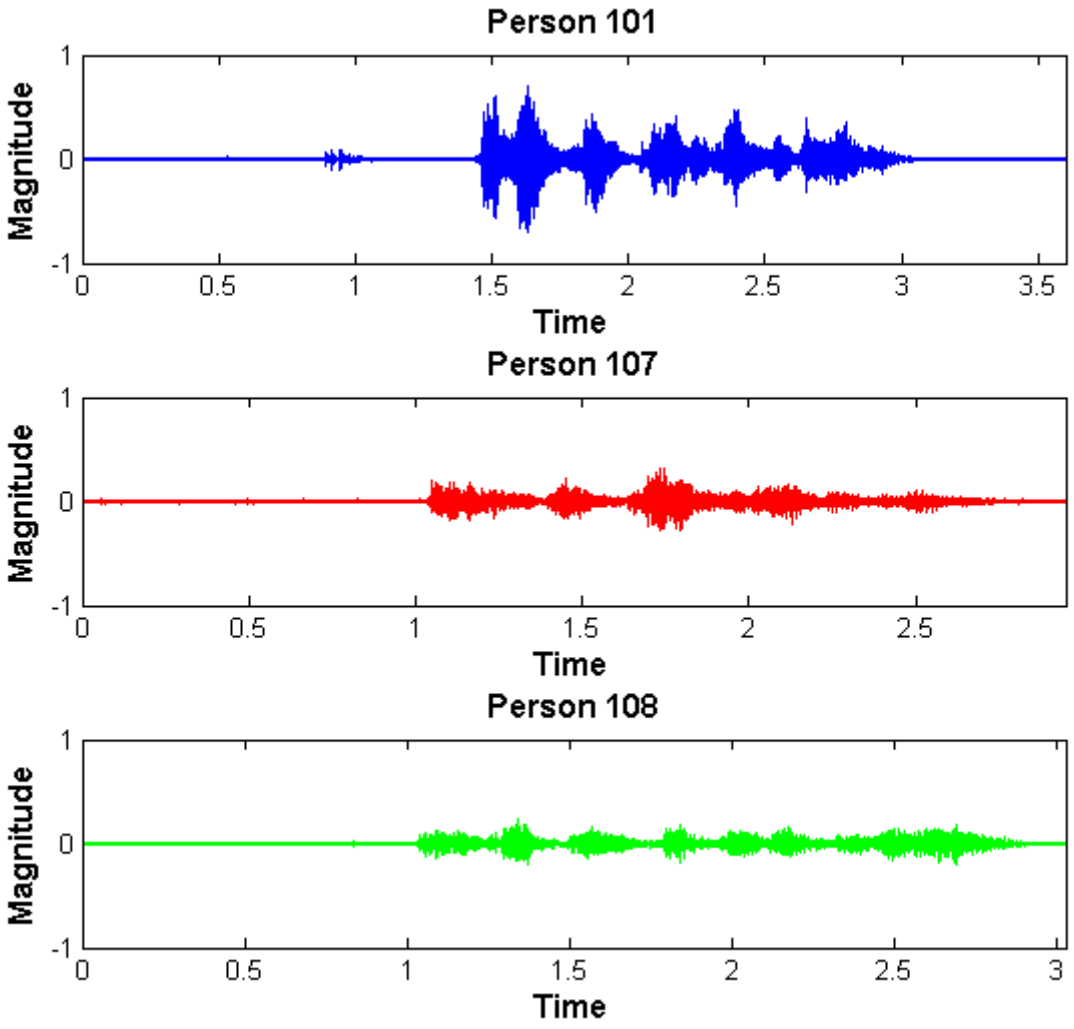

Figure 2.24: Speakers 101, 107 and 108 comparison.

RMFCC for SiriusV310

The combination of RMFCC and MFCC showed improvement in misclassification for the YOHO database. Therefore the RMFCC was used on the SiriusV310 database. Figure 2.25 shows the confusion matrix for RMFCC feature extraction on the SiriusV310 database.

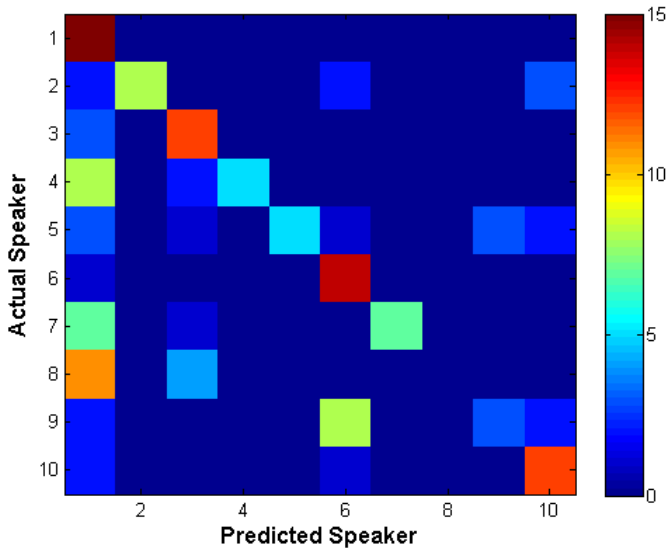

Figure 2.25: Confusion matrix using RMFCC features.

The misclassification rate for RMFCC for the SiriusV310 was found to be $46\% \pm 4.1\%$ which is very high and far away from the misclassification rate found using only MFCC. Next the combination of MFCC and RMFCC is performed on the SiriusV310 database. IAIF didn't show any good results for the YOHO database so it will be skipped for the SiriusV310 database.

Combinations of MFCC and RMFCC

The combination of MFCC and RMFCC gave the best results for the YOHO database. Figure 2.26 gives the misclassification for the weighting between MFCC and RMFCC using:

$$L_{final} = wL_{MFCC} + (1 - w) \cdot L_{RMFCC} \tag{2.12}$$

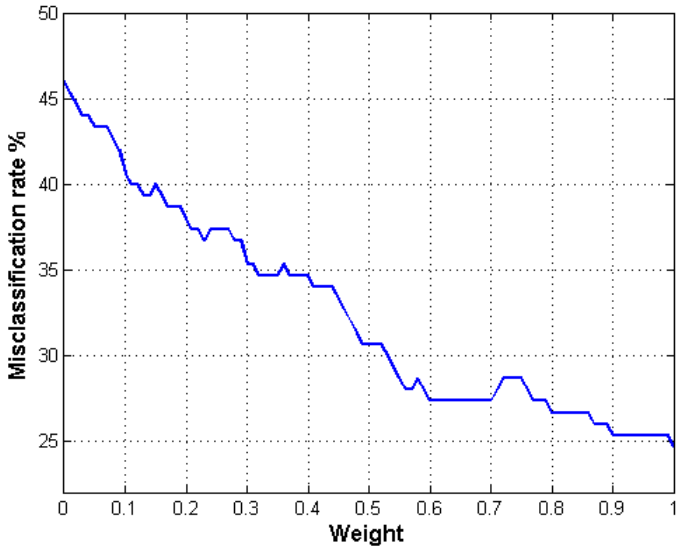

Figure 2.26: Misclassification as a function of weighting for combination of MFCC and RMFCC.

The combination of MFCC and RMFCC does not offer improvement compared to MFCC for the SiriusV310 database.

2.4 Summary

The misclassification rate for the YOHO database without combining likelihood vectors was minimum 10.13% by using MFCC feature extraction. By Combining 0.81 MFCC and 0.19 RMFCC likelihood vectors the misclassification rate was improved to 8.81%. There was a difference between the results of the YOHO database and the new SiriusV310 database. Misclassification rate of the new SiriusV310 database was 24.67%.

Chapter 3

Speaker Localization

Localization of sound sources is useful for many practical reasons, for example to avoid dangers or to help people get visual information of interesting things in their environment. In robotics, speaker localization can be used to give spatial information in the robot's environment. Increased flow of information helps the robot to react to situations around it. In situations where cameras cannot do everything the robotic hearing can be useful to analyse what is not seen. If something of interest is heard but not seen the cameras could be aimed in the direction of the sound source and visual information obtained.

Speaker localization can be divided into two parts: finding the time difference between microphones and estimating the angle of the direction of the sound source. The generalized cross correlation with phase transform [3], [7] is used to find the time difference between microphones in samples and the geometry of the microphone array is used to estimate the azimuth angle. The elevation of the speakers can also be found but is ignored because of the Kinect sensor microphone setup.

When the speaker location is estimated a beamforming is performed by delaying and summing the channels. This amplifies signal from one direction while suppressing signals from other directions. Also the noise should suppress because noise is supposed to be uncorrelated and therefore the signal to noise ratio should decrease after beamforming. The beamformed speech signal is now used for speaker identification.

3.1 Theory

3.1.1 Generalized Cross Correlation with Phase Transform (GCC-PHAT)

To estimate the sample difference between two signals a cross correlation is ideal because it measures how much a signal needs to be shifted such that it is as related to another signal. Generalized cross correlation (GCC) maximizes the traditional cross correlation. Signals received at two microphones are specified in the following way,

$$y_1(t) = h_1(t) * s(t) + n_1(t) \quad 0 \leq t \leq T, \tag{3.1}$$

$$y_2(t) = h_2(t) * s(t - \tau) + n_2(t), \tag{3.2}$$

where $s(t)$ is the speech signal, $n_1(t)$ and $n_2(t)$ are the noise signals at each channel and $h_1(t)$ and $h_2(t)$ are the impulse responses of the reverberant channels. In this work the time window T used for this processing is usually around 30 ms. The signals are assumed to be uncorrelated with the noise. The cross correlation is

$$R_{y_1,y_2}(\tau) = E[y_1(t)y_2(t - \tau)] \tag{3.3}$$

The estimation of time difference of arrival between the signals is [3],

$$\hat{\tau} = \arg\max_{\tau} R_{y_1,y_2}(\tau). \tag{3.4}$$

The generalized cross correlation for a given time lag is [3][7]:

$$R^g_{y_1,y_2}(\tau) = \int_{-\infty}^{\infty} W(f)Y_1(f)Y_2^\star(f)e^{j2\pi f\tau}df \tag{3.5}$$

where * denotes a complex conjugate, $W(f)$ is the weighting function, $Y_1(f)$ and $Y_2(f)$ are the Fourier transforms of the signals $y_1(t)$ and $y_2(t)$.
Phase transform means that the information from the phase at each frequency is only used. The phase transform weighting is given by [3][7]

$$W_{phat}(f) = |Y_1(f)Y_2^\star(f)|^{-1}. \tag{3.6}$$

Putting that weighting function into Equation (3.5) the GCC-PHAT is found to be

$$R^g_{y_1,y_2}(\tau) = \int_{-\infty}^{\infty} \frac{Y_1(f)Y_2'(f)}{|Y_1(f)Y_2^\star(f)|}e^{j2\pi f\tau}df. \tag{3.7}$$

For simplicity the GCC-PHAT would be implemented the following way:

$$\hat{G}_{y_1,y_2} = \frac{Y_1(f)Y_2(f)^\star}{|Y_1(f)Y_2(f)^\star|} \tag{3.8}$$

where the inverse Fourier transform of $\hat{G}_{y_1,y_2}$ would give $\hat{R}_{y_1,y_2}$ the estimation of $R^g_{y_1,y_2}$ and therefore the estimation of time delay would be

$$\hat{\tau} = \arg\max_{\tau} \hat{R}_{y_1,y_2}(\tau). \tag{3.9}$$

3.1.2 Sound Source Localization

When there is an array of microphones the location of sound source is simply a trigonometry. If there is knowledge about how much more time it takes for the sound to travel to one microphone than to another we can estimate the rotation angle for the Kinect sensor. The time difference of arrival (TDOA) was found in samples with GCC-PHAT as shown before and using that knowledge we know that

$$x = TDOA \cdot c \tag{3.10}$$

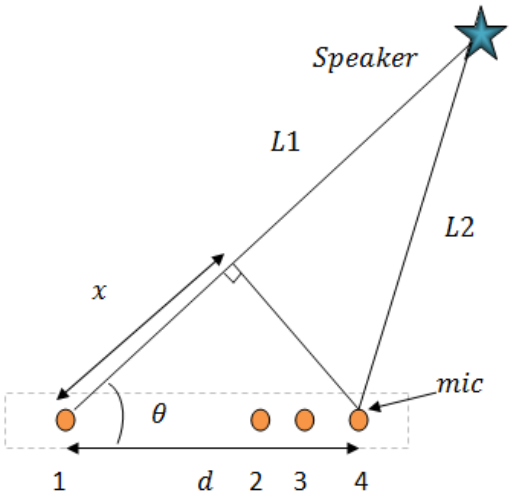

Figure 3.1: Speaker localization.

where c = 343 m/s is the speed of sound in air at 20° C, x is the distance difference for the sound wave to reach the microphones and d is the distance between the microphones. Assume that that the sound source is far away, then the lines L1 and L2 seen in Figure 3.1 can be assumed to be almost parallel and the estimation of the angle θ at microphone 1 is then

$$\theta = \cos\left(\frac{x}{d}\right). \tag{3.11}$$

The rotation of the Kinect sensor ϕ is about the middle so the rotation of the Kinect sensor is not exactly the same as the rotation at microphone 1 but because of the assumption that the speaker is far away the difference is very small. The rotation should also be given in that form, that is, if the speaker is in the front of the array the rotation should be 0°. We know that

$$sin(\theta) = cos(\pi/2 - \theta) \tag{3.12}$$

so the rotation angle should be

$$\phi = \sin\left(\frac{x}{d}\right) \tag{3.13}$$

Notice if the distance d between the microphones is less than $\lambda/2$, where λ is the wavelength of the sound wave, then there is no knowledge of which microphone comes first. This is because x cannot be larger than d and x is based on the time difference between the sound waves. If there is no knowledge of which of the microphones is before then the estimation of the TDOA becomes very difficult because there will be more than one option for TDOA.
Figure 3.2 shows for example two different waves in blue and green. The Blue wave is the reference signal and the green wave is $3\pi/2$ out of phase compared to the blue one.

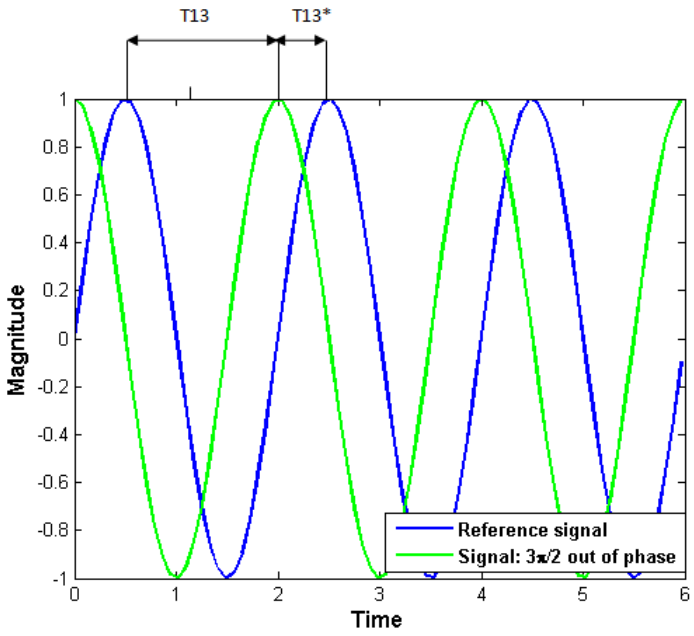

Figure 3.2: Difference between phase shifted waves.

T13 is the time difference between the blue and the green signals. T13* is the time difference between blue and green signals the other way around. If the assumption that the difference between the waves is less than half the wavelength then it should be clear that T13 should not be used because $T13 \times c$ would become larger than half the wavelength which indicates that the green wave is in front of the blue one and the time difference between them is T13*. But if the half wavelength condition is not met then there is no knowledge which signal comes first. There is no knowledge which of the time differences

should be used T13 or T13* between the green and the blue wave.

The GCC-PHAT has some restrictions. It is dependant on the sampling frequency of the device which is a restriction of most discrete time signal processing. GCC-PHAT is able to estimate how much one signal needs to be shifted such that it is identical to another signal. This means that the greater the sampling frequency the more precision is obtained. The precision shown in Figure 3.3 is obtained when using Kinect sensor which has 16 kHz sampling frequency.

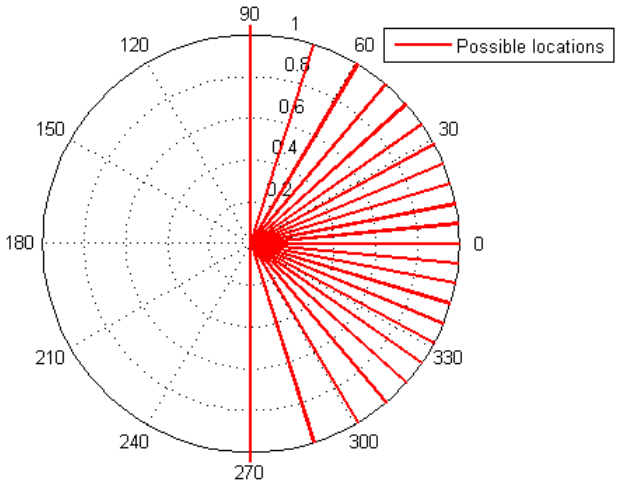

Figure 3.3: The precision of GCC-PHAT for 16 KHz sampling frequency.

As can be seen the greatest precision is at angles closer to $0°$. This happens because of how the arc sine works as seen in Figure 3.4.

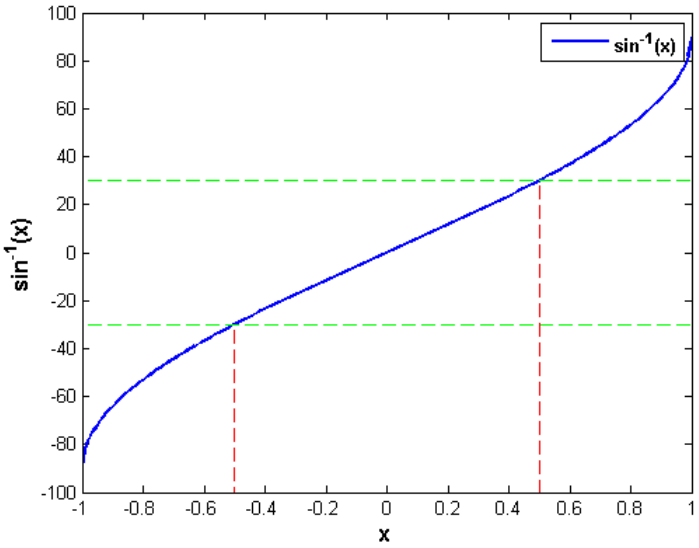

Figure 3.4: Arc sine.

This is the reason for difference between possible locations. The difference between possibilities are almost linear for $-30°$ to $30°$ but becomes greater for angles which are further away from zero. The linear part corresponds to half of the x axis but only one third of the y axis. Figure 3.4 demonstrates how 50% of the angles are in the range of $-30°$ to $30°$. This problem can be solved by having the microphone array in a cross structure. The issue of the great jumps happening on the part from $80°$ to $120°$ would become linear like the jumps between possibilities from $-30°$ to $30°$. Having the microphone array in a cross structure, would give more information and precision all around the device. That is, it would give valuable information whether or not the speaker is in front or behind the device and more precision to locate speakers on the side, see Figure 3.5 for details.

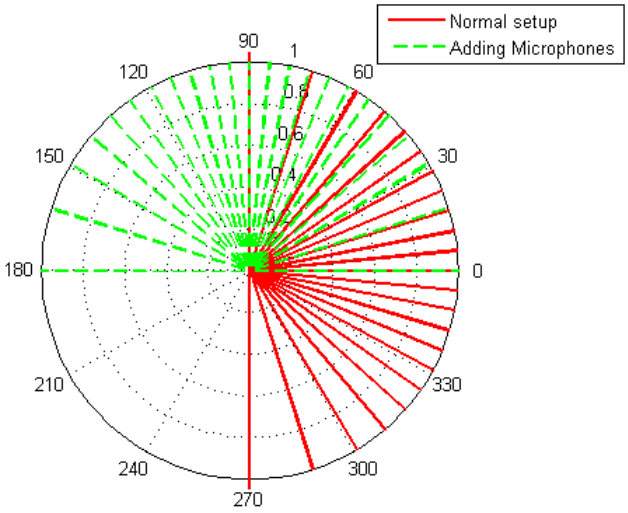

Figure 3.5: The precision with cross structure.

For real-time processing the sound source localization should be performed on a running windows of the signal. Determining the size of the window needs to be done because there is a trade off between speed and precision. More data leads to more precision for the correlation but more data also means more computational time.

3.1.3 Delay And Sum Beamforming

When different microphones record data some time difference is between the microphones because the microphones are separated. This difference can be used to do beamforming. The delay-and-sum beamforming is given by:

$$y_{beamed}(t) = \frac{1}{K}\sum_{i}^{K} y_i(t + \Delta_i t) \tag{3.14}$$

where $y_{beamed}(t)$ is the beamformed signal, K is the number of microphones, y_i is the signal for microphone i and $\Delta_i t$ is the time difference between the reference microphone and microphone i. The final signal becomes the sum of the channels all moved at the same time stamp. This is an ability to aim at some specific place and listen more closely at that place and because the signals are correlated but the noise is not, the signal becomes stronger and the noise decreases.

3.2 Implementation

For speaker localization the following steps are used for analysis.

1. **Collect data.** Speech signals are recorded at known angles.
2. **Find the phase difference.** The signals are windowed and GCC-PHAT is used to estimate the sample difference between any two signals.
3. **Estimate azimuth.** The azimuth angle is found using the time difference obtained from the sample difference found before.
4. **Estimate errors.** The estimated angle is compared with the desired angles giving the estimated errors.

3.3 Speaker Localization Results

GCC-PHAT was used to estimate the time difference between signals and using that information the azimuth angles were estimated. Figures 3.6 and 3.7 show estimated locations of each speaker for each utterance. The whole utterances are used in this part but next chapter shows the results for simultaneous localization by windowing the signals. The ENROLL folder was used for localization and the five measured angles were $-56°, -29°, 0°, 32°$ and $54°$.

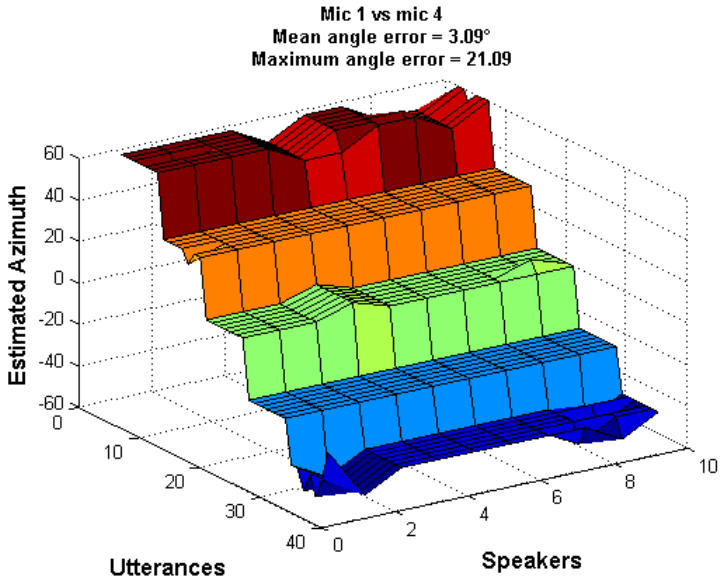

Figure 3.6: Localization of speakers of the SiriusV310 database using mic 1 and 4.

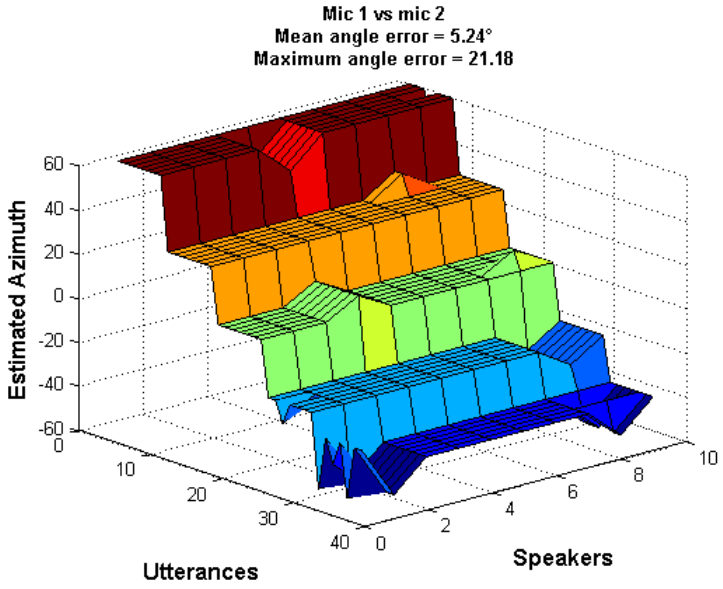

Figure 3.7: Localization of speakers of the SiriusV310 database using mic 1 and 2.

As can be seen on Figures 3.6 and 3.7 the behaviour is almost staircase with small errors. The greatest errors happen when the angles are supposed to be close to $\pm 60°$. The mean errors are not large but the maximum errors are quite high. The reason for that could be that the difference between the measurement and the estimated values becomes larger for higher degrees because of the issue of the arc sine. The number and magnitude of errors for the SiriusV310 database can be seen in the histograms in Figures 3.8 and 3.9.

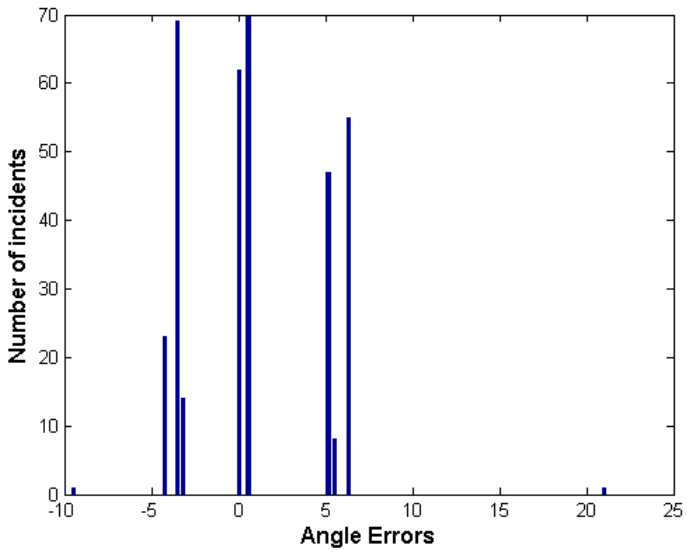

Figure 3.8: Angular errors between measured and estimated for microphones 1 and 4.

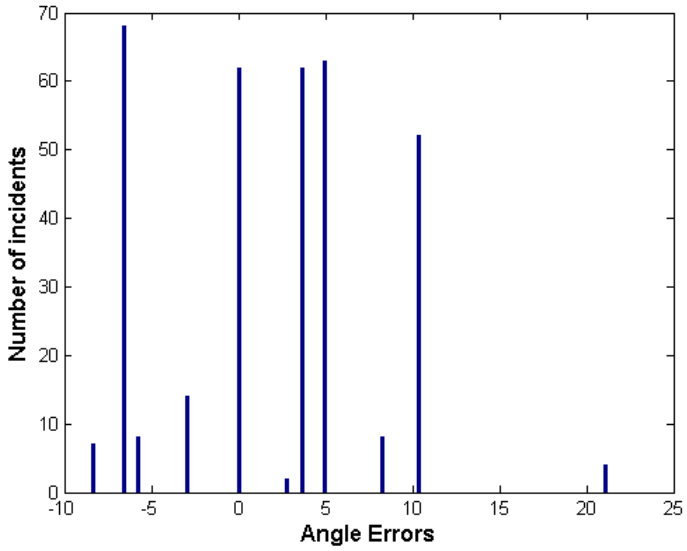

Figure 3.9: Angular errors between measured and estimated for microphones 1 and 2.

The corresponding average, maximum and standard deviation for the azimuth errors can be seen in Table 3.1

Table 3.1: Summary of average errors, maximum errors and error standard deviation for localization without windowing.

Microphones	Average error	Max error	Standard deviation
1 and 4	$3.09°$	$21.09°$	$3.92°$
1 and 2	$5.24°$	$21.18°$	$6.13°$

From Table 3.1 the precision of the estimated locations is better when the distance between the microphones is larger. The average precision for the human ear has been researched to be $5 - 6°$ by Carlile et al [10]. This estimation for the human hearing is not far away from what SiriusV310 database gave using GCC-PHAT. In practice the azimuth and elevation should be estimated to give realistic behaviour to a robot. To be able to estimate the elevation the microphones could be arranged in a pyramid structure giving information about elevation azimuth and precision for all sides.
The sample difference between the microphones was found using GCC-PHAT. This difference was then used to delay-and-sum the signals for beamforming. Figure 3.10 shows the confusion matrix for the speaker identification of the beamformed database.

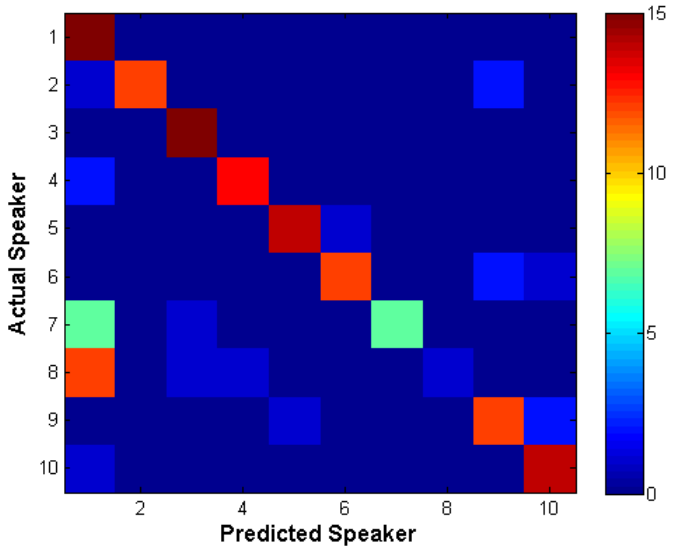

Figure 3.10: Confusion matrix using mfcc features on beamformed data.

The test set misclassification rate for the beamformed SiriusV310 database using GMM and MFCC was found to be 23.33%, which improves the non beamformed MFCC version by 5.4% in misclassification rate. This improvement is because after beamforming the signal to noise ratio becomes lower therefore less noise is in the signal leading to better speaker identification.

3.4 Summary

Minimum average error was 3.09° when the microphones were further away from each other, which is on the same level as the human localization precision. This showed the robustness of the GCC-PHAT method.
Delay and sum beamforming the signals improved the misclassification rate for the SiriusV310 database to 23.33%.

Chapter 4

Real-Time Simultaneous Speaker Identification and Localization

Applications that could benefit from speaker identification and localization need real-time results. For real-time processing the acquired data is of some specific size. Some applications lack memory so the amount of data can be crucial for performance. The aim of this chapter is to find out how the identification and localization is affected by the amount of data used for computation.

4.1 Implementation

4.1.1 Preparation

First there is one major issue which was solved. The signals can not begin with noise. For a typical record the speaker starts talking few moments after the record button is hit. So to be able to estimate the misclassification rate as a function of window size the signal must start where the speech starts. This was done by separating the records into bins. Each bin containing 1000 samples (1/16 of second) of recorded data. For each bin the power is found (sum of each sample squared). The average value of these power values is calculated and the bin which is the first to have higher power value than the average power value indicates the start of the speech. Figure 4.1 shows first the signal and the bins then how the signal has been shifted to the start of the speech.

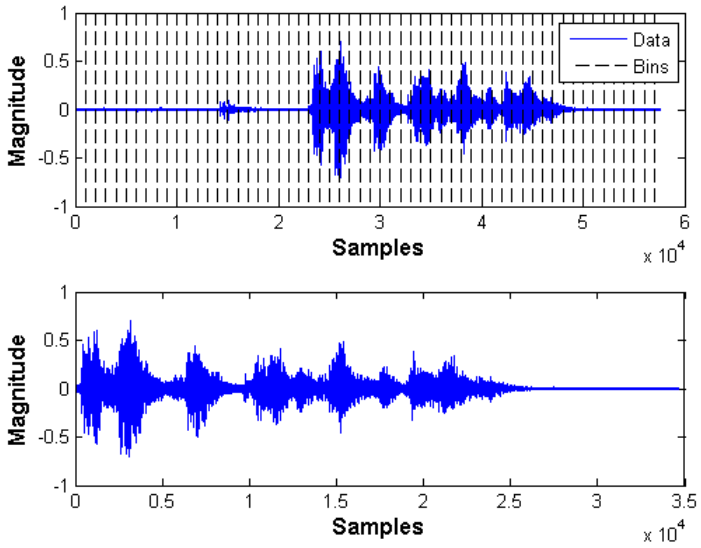

Figure 4.1: Typical signal.

The localization of the speakers also needs the speech to be in the beginning of the signal.

4.1.2 Identification And Localization

When the data has been prepared the identification and localization can begin. The identification period starts at 7520 samples (470 ms) and increases by 880 samples (55 ms) in each iteration for speaker identification. The reason that the period is not smaller is because the MFCC and GMM programs had issues when the amount of data was too small. These issues have to be solved as future work. The identification is set up as before, with MFCC feature extraction and GMM for modelling. The models are made from whole sentences as before (ENROLL data). The only difference is that the VERIFY data is identified according to the identification period size. The identification period for localization starts at 200 samples (12.5 ms) and increases by 1000 samples (62.5 ms) in each iteration.

4.2 Results

4.2.1 Speaker Identification

Figure 4.2 shows the misclassification rate as a function of identification period size for the SiriusV310 database.

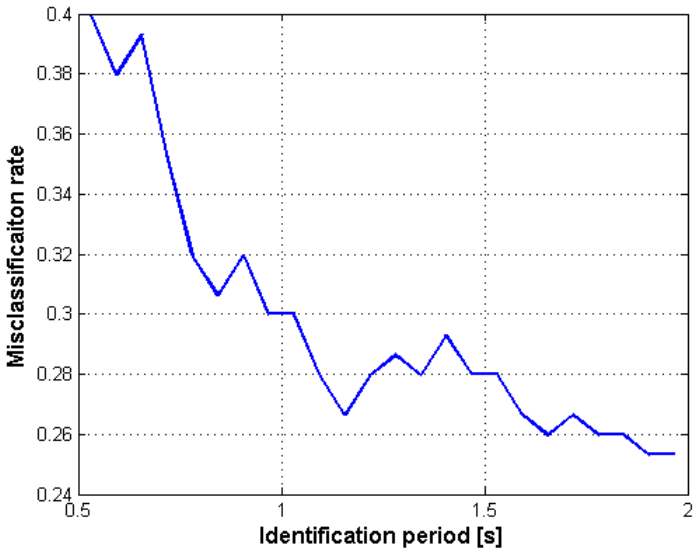

Figure 4.2: Misclassification as a function of identification period.

As has been found before the misclassification rate was around 25% for identification period of maximum size (the whole data). Figure 4.2 shows that the misclassification increases a lot by decreasing the identification period. The largest sound file is around 2 seconds long after the preparation procedure so the assumption could be made that larger sound files, with more speech, should lower the misclassification rate. That is by listening more to a person talking increases the accuracy of the system. Also notice that these real-time simultaneous speaker identification results forget the past. Future work could be, using the previous likelihoods and the present knowledge of the speaker for classification.

4.2.2 Real-Time Speaker Localization

Figure 4.3 shows the mean error and standard deviation of errors as a function of the identification period.

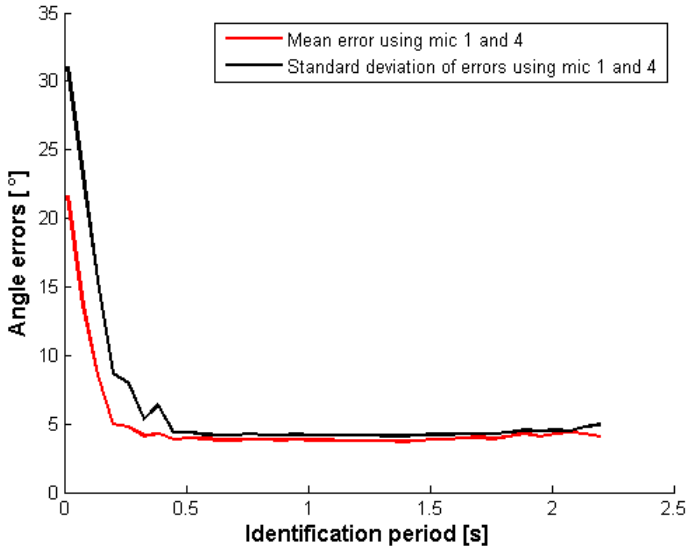

Figure 4.3: Mean errors and standard deviation.

As seen in Figure 4.3 the accuracy of localization is not improved further with identification periods larger than approximately 0.5 seconds. Real-time application for speaker localization should therefore consider 0.5 seconds as minimum identification period.

Chapter 5

Summary Of Results

5.1 Speaker Identification

Table 5.1: Speaker identification summary for the YOHO and SiriusV310 database.

Database	Feature extraction type	Misclassification rate
YOHO Full	MFCC	$10.13\% \pm 0.41\%$
YOHO Full	RMFCC	$30.96\% \pm 0.62\%$
YOHO Full	IAIF	$62.04\% \pm 0.65\%$
YOHO Full	Random classification	99.96%
SiriusV310	MFCC	$24.67\% \pm 3.52\%$
SiriusV310	RMFCC	$46.00\% \pm 4.10\%$
SiriusV310	Beamformed	$23.33\% \pm 3.45\%$

Table 5.2 shows the combinations for the YOHO database. No combination improved the misclassification rate for the SiriusV310 database.

Table 5.2: Lowest misclassification rate of combined methods for the YOHO database.

Combinations	Misclassification rate	Best combination
MFCC and RMFCC	8.81%	81% MFCC 19% RMFCC
MFCC and IAIF	10.13%	100% MFCC 0% IAIF
RMFCC and IAIF	23.86%	62% RMFCC 38% IAIF
MFCC, RMFCC and IAIF	8.81%	80% MFCC 20% RMFCC and 0% IAIF

For real-time identification the identification period is crucial for precision of the system. Figure 4.2 showed how the misclassification rate increased rapidly for smaller identification period.

5.2 Speaker Localization

Table 5.3 shows a summary of localization errors.

Table 5.3: Accuracy of the speaker localization methods. Summary of errors.

Type	Microphones	Average error	Max Error	Standard deviation
Not windowed	1 and 4	3.09°	21.09°	3.92°
Not windowed	1 and 2	5.24°	21.18°	6.13°

The distance between the microphones seems to matter for the localization accuracy, the further away the better. Also the mean and standard deviation of the azimuth errors were shown to increase for identification periods smaller than 0.5 seconds. The accuracy was almost constant for identification periods larger than 0.5 seconds.

Chapter 6

Conclusions, Discussion And Future Work

6.1 Conclusions

The reasons for the differences between the SiriusV310 database results and the YOHO database results are many. One reason is that the sampling frequency between the databases are different. The speaker identification results from the YOHO database showed improvement of 13.06% when MFCC feature extraction was combined with RMFCC. For the SiriusV310 database the speaker identification for a real-time processing by windowing, showed that the misclassification rate is dependent on window size (identification period). More data leads to less misclassification rate. When designing an application for speaker identification the identification period needs to be considered. If the application needs high accuracy the identification period should be as large as possible. Also accumulating the likelihoods over a period of time is of interest.
The localization accuracy of the SiriusV310 database using GCC-PHAT showed that the majority of errors are within $\pm 10^{\circ}$. As was stated before the accuracy is dependent on sampling frequency which indicates that with higher sampling frequency the error could become lower than the human sound source localization error.
For both identification and localization the identification period is crucial. It depends highly on what the user wants for accuracy. If an application needs high accuracy for speaker identification the identification period should be as large as possible but for an application which only needs to locate speakers an identification period of 0.5 seconds should be enough. Notice that 0.5 seconds are 8000 samples of recorded data according to the 16 kHz sampling rate of the Kinect sensor. Increasing the accuracy could decrease

the size of ideal identification period in seconds but the amount of data would probably be the same.

6.2 Discussion

Improving the speaker identification method using MFCC feature extraction by combining RMFCC and MFCC shows that new feature extraction methods have to be invented. Also it could improve the performance to combine different methods of modelling data along with combination of different feature extraction methods. The difference between misclassification rates between the YOHO database and the SiriusV310 databases is not fully known and deserves further investigation in future work.
Localization accuracy of the speakers is on par with the human accuracy, which shows the robustness of the GCC-Phat method. Increasing the sampling frequency would therefore be of interest. The possibility of being more accurate than humans is useful and the ability to locate speakers in three dimensions using multiple microphones is very practical.
Applications that need to identify and locate speakers in real-time should consider the identification period size. Does the application need high accuracy in identification or localization? The misclassification rate for small identification periods has to be improved because sounds can easily be less than one second in length.
In reality there are usually more than one speaker to be heard at any given time. Humanoid robots should be able to recognize and locate multiple speakers at the same time. The signals need to be separated to be able to do so.
If a humanoid robot needs to react to what happens around him then the robot needs to know the difference between different sources of sound. Most speaker recognition systems are based on human speaker recognition, but most sounds do not come from human speakers. A Good research question could be: How to recognize sounds in general? If the robots are ever going to take over the world they have to be able to understand what happens around them.
If a humanoid robot would be made in the near future the microphone setup should be a pyramid form or even in the same form as the Eigenmike (MH acoustics) which is a sphere covered with microphones with abilities to aim to desired directions. That kind of setup could be wise for a humanoid robots, that is, the head would be a sphere covered with microphones and cameras giving information all around the robot.

Figure 6.1: Eigenmike by MH acoustics (www.mhacoustics.com).

An application which could also be considered is a localization and recognition system for deaf people. By adding some microphones to a shirt or a belt an artificial intelligent system could be made to help deaf people recognizing their environment. The microphones would all be connected to different mini computers which would communicate to a phone. Vibration at different locations in the clothing could be used to indicate what sounds are in the environment and where they come from, and the intensity of the vibration could tell the user what kind of sound it is. All the information would be collected to the phone which would give visual information of what is happening.

6.3 Future Work

Future work could be one of the following:

1. Estimate the misclassification rate of the YOHO database as a function of identification period.
2. Implement a speech separation for speaker recognition purposes.
3. Implement a speech separation on windowed signals for speaker recognition.
4. Enlarge the SiriusV310 database.
5. Test speaker identification of sounds in general, humans and non-human speakers.
6. Locate speakers in 3D and estimate the distance to them with microphones.
7. Have two speakers and use delay and sum beamforming to listen more closely to one of them. Find the corresponding misclassification rate.
8. Investigate different feature extraction methods for sounds in general.
9. Research speech separation methods and implement them.

10. Investigate the difference in misclassification rates between the YOHO and SiriusV310 databases.
11. Use the previous likelihoods and the present knowledge of the speaker for classification.

Bibliography

[1] Paavo Alku. Glottal wave analysis with pitch synchronous iterative adaptive inverse filtering. *Speech communication*, 11:109–118, 1992.

[2] Christopher M. Bishop. *Pattern Recognition and Machine learning*. Springer Science, 2006.

[3] Michael S Brandstein and Harvey F. Sivlerman. A robust method for speech signal time-delay estimation in reverberant rooms. *ICASSP*, 1:375–378, 1997.

[4] Joseph P. Campbell. Speaker recognition: A tutorial. *Proceedings of the IEEE*, 85, Issue: 9:1437–1462, 1997.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal Royal Statistical Society, Series B*, 39:1–38, 1977.

[6] Hoang Do, Ivan Tashev, and Alex Acero. A new speaker identification algorithm for gaming scenarios. *ICASSP*, pages 5436–5339, 2011.

[7] Charles H. Knapp and C. Clifford Carter. The generalized correlation method for estimation of time delay. *IEEE Transaction on signal prcessing*, 24:320–327, 1976.

[8] Douglas A. Reynolds. Speaker identification and verificartion using Gaussian mixture speaker models. *Speech communication*, 17:91–108, 1995.

[9] Douglas A Reynolds, Thomas F. Quatieri, and Robert B. Dunn. Speaker verificartion using adapted Gaussian mixture models. *Digital signal processing*, 10:19–41, 2000.

[10] Carlile S, Leong P, and Hyams S. The nature and distribution of errors in sound localization by human listeners. *Hear Research*, 114:179–196, 1997.

Appendix A

Appendix

A.1 Speaker identification configurations

For speaker identification for both of the databases the following configurations were used

A.1.1 MFCC and RMFCC configurations

For MFCC and RMFCC a Hamming window was used for the time domain and triangular shaped filters in the mel domain. The filters act in the absolute magnitude domain and the zeroth order cepstral coefficient was also included. Deltas dc/dt were also included and delta delta coefficients d^2c/dt^2.
Other configurations can be seen in Table A.1 for the YOHO database and table A.2 for the SiriusV310 database

Table A.1: MFCC and RMFCC configurations for the YOHO database

number of cepstral coefficients excluding 0'th coefficient	12
number of filters in filterbank	26
length of frame in samples	256
frame increment	80
Low end of the lowest filter as fraction of fs	0
High end of highest filter as fraction of fs	0.5

Table A.2: MFCC and RMFCC configurations for the SiriusV310 database

number of cepstral coefficients excluding 0'th coefficient	12
number of filters in filterbank	26
length of frame in samples	512
frame increment	160
Low end of the lowest filter as fraction of fs	0
High end of highest filter as fraction of fs	0.5

A.1.2 IAIF configurations

The configurations for the IAIF feature extraction can be seen in table A.3. Note that the IAIF is a manipulation of the signals which uses MFCC to make the features.

Table A.3: IAIF configurations

First vocal tract LPC order	20
Glottal source LPC order	4
Second vocal tract LPC order	20
number of cepstral coefficients excluding 0'th coefficient	12
number of filters in filterbank	26
length of frame in samples	240
frame increment	80
Low end of the lowest filter as fraction of sampling frequency	0
High end of highest filter as fraction of sampling frequency	0.5

A.1.3 GMM configurations

The configuration for GMM can be seen in table A.4

Table A.4: GMM configurations

Number of mixtures	64
Train iterations	10
Maximum loop count	20
Stopping threshold	0.001
Initialization mode	K-means, K randomly selected points.

A.2 Database descriptions

The specifications for the databases:

TIMIT Speakers 630, near Ideal conditions [8]. 10 utterances per speaker, 8 dialect regions, Recorded with 16 kHz sampling rate .

NTIMIT Speakers 630, "used to gauge the identification performance loss incurred by transmitting speech over the telephone network for the same large population experiment as TIMIT"[8]

Switchboard Reynolds [8] uses 113 speakers. Switchboard is a large multi speaker database of telephone bandwidth speech. Contains 2430 conversations with about 3 million words spoken by >500 speakers.

YOHO Speakers 138, "Used to determine performance on a vocabulary-dependent, office environment verification task."[8] Recorded at 8 kHz sampling frequency.

SiriusV310 Speakers 10, 5 males, 5 females, Recorded in an office environment with Kinect sensor. 4 Channels of data sampled at 16 kHz. Two sentences are the same for all speakers and eight sentences are unique for each speaker.

A.3 Computer specifications

Two computers were used in the whole process. Computer 1 is a desktop which was only used in the end. Computer 2 is a laptop and was mostly used for smaller runs.

Table A.5: MFCC and RMFCC configurations for the YOHO database

Computer nr	Operating system	Memory	Processor
1 Desktop	Ubuntu 11.10 32 bit	7.8 GB	Intel Xeon (R) CPU E31245 @3.30GHz x8
2 Laptop	Windows 7 professional, 64 bit	4 GB	Intel(R) Core(TM) 2 Duo T9300 @2.5 GHz

School of Science and Engineering
Reykjavík University
Menntavegur 1
101 Reykjavík, Iceland
Tel. +354 599 6200
Fax +354 599 6201
www.reykjavikuniversity.is
ISSN 1670-8539