International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 8, August - 2013

Text-Independent Speaker Identification using Vector Quantization

Samudre N. A.
Assistant Professor, Department of Instrumentation Engineering,
VPM's Maharshi Parshuram College of Engineering, Ratnagiri.

Abstract

Nowadays it is obvious that speakers can be identified from their voices. In this work the details of speaker identification from the real-time system point of view are discussed. The speaker identification systems can be subdivided into text-dependent and text-independent methods. Text-dependent systems require the speaker to utter a specific phrase (pin-code, password etc.), while a text-independent method should catch the characteristics of the speech irrespective of the text spoken. The system developed in this work is the latter, text-independent, meaning the system can identify the speaker regardless of what is being said.

This paper presents text-independent speaker identification system, which consists of mapping a speech signal from an unknown speaker to a database of known speakers, i.e. the system has been trained with a number of speakers which the system can recognize. Speaker identification has been done successfully using Vector Quantization (VQ).

1. Introduction

Modern-day security systems are wide-ranging and usually have multiple layers to get through before they can be properly cracked. Aside from the standard locks and deadbolts and alarm systems, there are very complex methods to protecting important material. Many of these are methods that can allow or disallow a specific individual to access the material – a computer system has to be able to successfully detect a fingerprint, read an individual's eye patterns, or determine the true identity of a speaker. This last point is the focus of this paper – speaker identification.

Speaker recognition has been an interesting research field for the last decades, which still yields a number of unsolved problems. Speaker recognition is basically divided into speaker identification and speaker verification. Verification is the task of automatically determining if a person really is the person he or she claims to be. The focus of this paper is speaker identification, which consists of mapping a speech signal from an unknown speaker to a database of known speakers, i.e. the system has been trained with a number of speakers which the system can recognize. The systems can be subdivided into text-dependent and text-independent methods. Text-dependent systems require the speaker to utter a specific phrase (pin-code, password etc.), while a text-independent method should catch the characteristics of the speech irrespective of the text spoken. The system developed here is the latter, text-independent, meaning the system can identify the speaker regardless of what is being said. Speaker identification has been done successfully using Vector Quantization (VQ). This technique consists of extracting a small number of representative feature vectors as an efficient means of characterizing the speaker- specific features. Using training data these features are clustered to form a speaker-specific codebook. In the recognition stage, the test data is compared to the codebook of each reference speaker and a measure of the difference is used to make the recognition decision. The VQ in this paper is done utilizing Mel Frequency Cepstral Coefficients (MFCC)[1].

This paper aims to determine the true identity of a specific speaker. The speaker will speak a word to the system, and the actual word itself can be any word. The system can accept any word because it is a text-independent system, meaning there is no specified word need. The system will determine the identity of a user by examining the vowel sounds, from the input speech signal. The vowel sounds will be analyzed in the frequency domain, specifically by looking at the peaks, or formants, of the frequency response of the signal. These formants will be compared with the formants of the entire group members previously stored in the database of the system. The group member with the highest resulting value after the comparison is the one identified as the speaker by the system. If no user reaches the set threshold value, then the system responds by saying there is no match for the given speaker.

2. Literature review

The speaker identification problem has been addressed in the literature by representing the audio signal using different features, calculated in the frequency or in the time domain. Moreover, different classification paradigms have been employed, basically neural networks and gaussian mixture models.

The paper by Reynolds and Rose [6] illustrates one of the most frequently quoted systems. It uses features evaluated in the frequency domain by using the cepstral

analysis [7] (the so-called Mel-scale cepstral coefficients). Different classification paradigms are proposed and the best results are obtained with a gaussian mixture model. The system achieved over 94% recognition rate on the KING Speech Corpus database [8].

Several authors have proposed features calculated starting from the ones described by Reynolds and Rose. In particular, in [9] a transform was applied to the Mel cepstral features in order to compensate the noise components of the audio channel. From the transformed domain, the so-called formants features were then calculated and used for classification. Results on the NIST 95 database provided a 90% recognition rate.

In [10] the authors used also the wavelets. They proposed system architecture with a neural classifier for each speaker to be recognized. Identification was performed by assigning the input sample to the speaker whose associated classifier exhibited the highest output.

In order to reduce the computational complexity of the classification phase, in [11] the principal component analysis was used on the features proposed by Reynolds and Rose. In this way, the authors achieved a 90% recognition rate on a set of 50 speakers.

Other authors have proposed the use of acoustic features directly obtainable from the time domain, such as pitch, speech rate, voice quality and temporal variation of the audio signal. This is the case of [12], where a system devoted to recognize only voiced segments of the audio track is proposed and of [13], in which the authors try to identify the speakers of the KING Speech Corpus. In both cases Gaussian mixture models were employed at the classification stage. In automatic speaker identification (ASI), there is no a priori identity claim, and the system decides who the person is, what group the person is a member of, or (in the open-set case) that the person is unknown. General overviews of speaker recognition have been given by Atal, Doddington, Furui, O'Shaughnessy, Rosenberg, Soong, Sutherland, and Jack [2,9,13].

3. Applications

Practical applications for automatic speaker identification are obviously various kinds of security systems. Human voice can serve as a key for any security objects, and it is not so easy in general to lose or forget it. Another important property of speech is that it can be transmitted by telephone channel. This provides an ability to automatically identify speakers and provide access to security objects by telephone. Nowadays, this approach begins to be used for telephone credit card purchases and bank transactions. Human voice can also be used to prove identity during access to any physical facilities by storing speaker model in a small chip, which can be used as an access tag, and used instead of a pin code. Another important application for speaker identification is to monitor people by their voices. For instance, it is useful in information retrieval by speaker indexing of some recorded debates or news, and then retrieving speech only for interesting speakers. It can also be used to monitor criminals in common places by identifying them by voices. In fact, all these examples are actually examples of real time systems. For any identification system to be useful in practice, the time response, or time spent on the identification should be minimized. Growing size of speaker database is also common fact for practical systems and can also lead to system optimization.

The potential for application of speaker identification systems exists any time speakers are unknown and their identities are important. In meetings, conferences, or conversations, the technology makes machine identification of participants possible. If used in conjunction with continuous speech recognizers, automatic transcriptions could be produced containing a record of who said what. This capability can serve as the basis for information retrieval technologies from the vast quantities of audio information produced daily.

This system is not real time application at present, it can be implemented in real time product with some develpments. If it is possible to remove all the limitations of the sytstem, it can be implemented in some security mechanisms. For example, safes, quick access to doors, car protection against thefts, the protection of electronic systems like TV, video etc. The addition of a timed counter would enable these systems to be used in a attendance logging application system of workplace. The above list is by no means complete, but provides an indication of the types and variety of applications.

4. Speaker identification

Speaker identification is mapping a speech signal from an unknown speaker to a database of known speakers, i.e. the system has been trained with a number of speakers which the system can recognize. Speaker identification can be further divided into two branches. Open-set speaker identification decides to whom of the registered speakers' unknown speech sample belongs or makes a conclusion that the speech sample is unknown. In this work, we deal with the closed-set speaker identification, which is a decision making process of whom of the registered speakers is most likely the author of the unknown speech sample. Depending on the algorithm used for the identification, the task can also be divided into text-dependent and text-independent identification. The difference is that in the first case the system knows the text spoken by the person while in the second case the system must be able to recognize the speaker from any text [6].

The process of speaker identification is divided into two main phases. During the first phase, speaker enrollment, speech samples are collected from the speakers, and they are used to train their models. The collection of enrolled models is also called a speaker database. In the second phase, identification phase, a test sample from an unknown speaker is compared against the speaker database. Both phases include the same first step, feature extraction, which is used to extract speaker dependent characteristics from speech. The main purpose of this step is to reduce the amount of test data while retaining speaker discriminative information. Then in the enrollment phase, these features are modeled

and stored in the speaker database. This process is represented in Figure 1.

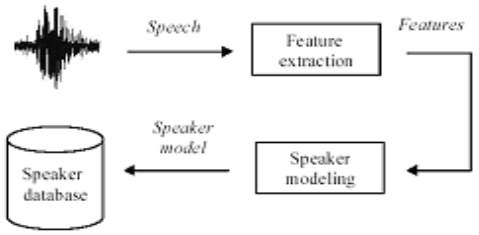

Figure 1. Enrollment Phase

In the identification step, the extracted features are compared against the models stored in the speaker database. Based on these comparisons the final decision about speaker identity is made. This process is represented in Figure 2.

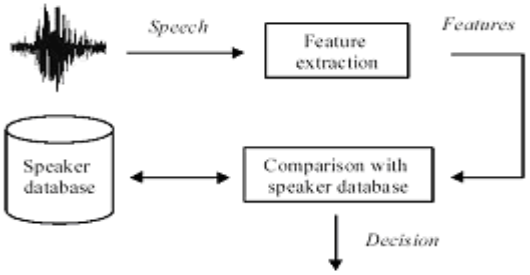

Figure 2. Identification Phase

4.1. Feature Extraction

The acoustic speech signal contains different kind of information about speaker. This includes "high-level" properties such as dialect, context, speaking style, emotional state of speaker and many others [14]. More useful approach is based on the "low-level" properties of the speech signal such as pitch (fundamental frequency of the vocal cord vibrations), intensity, formant frequencies and their bandwidths, spectral correlations, short-time spectrum and others [2].

From the automatic speaker identification point of view, it is useful to think about speech signal as a sequence of features that characterize both the speaker as well as the speech. It is an important step in identification process to extract sufficient information for good discrimination in a form and size which is amenable for effective modeling [4]. The amount of data, generated during the speech production, is quite large while the essential characteristics of the speech process change relatively slowly and therefore, they require less data. According to these matters feature extraction is a process of reducing data while retaining speaker discriminative information [3,4].

The speech wave is usually analyzed based on spectral features. There are two reasons for it. First is that the speech wave is reproducible by summing the sinusoidal waves with slowly changing amplitudes and phases. Second is that the critical features for perceiving speech by humans ear are mainly included in the magnitude information and the phase information is not usually playing a key role [15]. In our paper Mel Frequency Cepstral Coefficients as features for the classification problem are used.

4.2. Framing and Windowing

The speech signal is slowly varying over time (quasi-stationary) that is when the signal is examined over a short period of time (5-100msec), the signal is fairly stationary. Therefore speech signals are often analyzed in short a time segment, which is referred to as short-time spectral analysis[16]. It works in the following way: predefined length window (usually 20-30 milliseconds) is moved along the signal with an overlapping (usually 30-50% of the window length) between the adjacent frames. Overlapping is needed to avoid losing of information. Parts of the signal formed in such a way are called frames. In order to prevent an abrupt change at the end points of the frame, it is usually multiplied by a window function. The operation of dividing signal into short intervals is called windowing and such segments are called windowed frames (or sometime just frames). The most popular window function used in speaker identification is Hamming window function, which is described by the following equation:

$$h(n) = 0.54 - 0.46\cos(2\pi n / N-1),\ 0 \le n \le N-1 \quad (1)$$

where N is the size of the window or frame. A set of features extracted from one frame is called feature vector.

4.3. Mel-Frequency Cepstrum Coefficients

Mel-frequency cepstrum coefficients (MFCC) are well known features used to describe speech signal. They are based on the known evidence that the information carried by low-frequency components of the speech signal is phonetically more important for humans than carried by high-frequency components [16]. Technique of computing MFCC is based on the short-term analysis, and thus from each frame a MFCC vector is computed. MFCC extraction is similar to the cepstrum calculation except that one special step is inserted, namely the frequency axis is warped according to the mel-scale. Summing up, the process of extracting MFCC from continuous speech is illustrated in Figure 3.

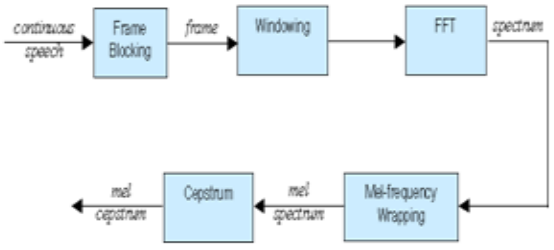

Figure 3. MFCC Block Diagram

One useful way to create mel-spectrum is to use a filter bank, one filter for each desired mel-frequency component.

Every filter in this bank has triangular bandpass frequency response. Such filters compute the average spectrum around each center frequency with increasing bandwidths.

This filter bank is applied in frequency domain and therefore, it simply amounts to taking these triangular filters on the spectrum. In practice the last step of taking inverse DFT is replaced by taking discrete cosine transform (DCT) for computational efficiency. Different approach to the computation of MFCC than described in this work is represented in [17] that is simplified by omitting filter bank analysis.

4.4. Vector Quantization

Speaker recognition systems are inherent of a database, which stores information used to compare the test speaker against a set of trained speaker voices. Ideally, storing as much data obtained from feature extraction techniques is advised to ensure a high degree of accuracy, but realistically this cannot be achieved. The number of feature vectors would be so large that storing and accessing this information using current technology would be unfeasable and impractical.[18]

Vector Quantization (VQ) is a quantization technique used to compress the information and manipulate the data in such a way to maintain the most prominent characteristics. VQ is used in many applications such as data compression (i.e. image and voice compression), voice recognition, etc. VQ in its application in speaker recognition technology assists by creating a classification system for each speaker. Given the extracted feature vectors (known as codewords) from each speaker, each codeword is used to construct a codebook. This process is applied to every single speaker to be trained into the system. VQ codebook algorithms are inherently difficult to implement. Although numerous VQ algorithms exist, Linde-Buzo-Gray or LBG VQ Algorithm is chosen, since it is the easiest to implement [19].

4.4.1 VQ-Linde Buzo Gray (LBG) Algorithm The LBG can be classified as an iterative procedure. [19] The LBG algorithm is operated on a given codebook. LBG splits the codebook into segments and performs an exhaustive analysis on each segment. The analysis compresses the training vector information creating a new codebook which is then used to compute the next segment. The Flow Diagram of VQ-LBG Algorithm is shown in figure 4.

This code book is obtained using a splitting method. In this method an initial codevector is set as the average, and then split in two vectors. Then the iterative algorithm is run with those two vectors. Resulting two vectors are then split again into 2 vectors each. This give us now four vectors, and the process is then repeated until the desired number of codevectors is obtained.

The process continues until all segments have been processed and the new codebook is created. The aim of this algorithm is to minimise any distortions in the data creating a codebook which is computationally optimised, while providing a sub-optimal solution. The performance of VQ analysis is highly dependent on the length of the voice file which is operated upon.

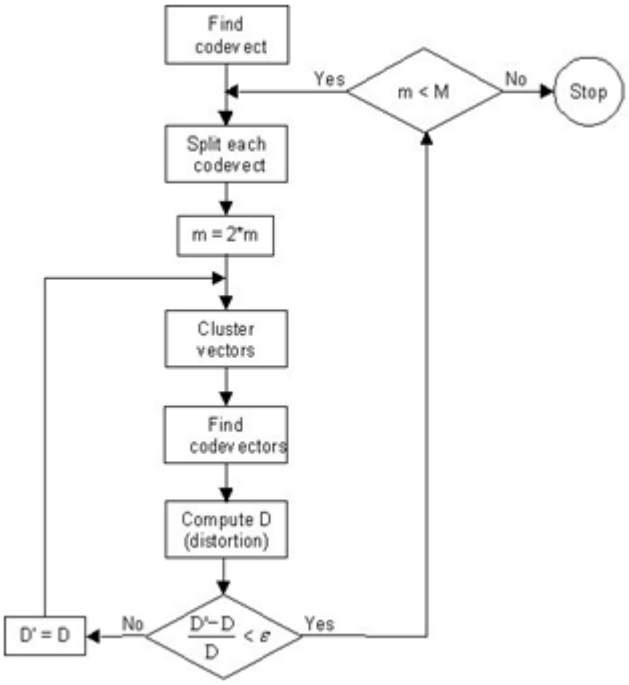

Figure 4. Flow Diagram of VQ-LBG Algorithm

4.5. Decision

The decision making logic is handled by a concept known as the threshold. The threshold determines the acceptable boundaries dictating the final answer. The system will only result in a solution if the following criteria are met.

- The system has found the lowest Euclidean Distance between the codebook tested and the various trained codebooks.
- The distance calculated falls below a pre-defined threshold of acceptance.

Both requirements must be satisfied in order for the system to produce a result, otherwise the voice signal in test will be rendered as an "unknown speaker".

5. Implementation and results

Two Speech files of each speaker are stored one in the Train folder for Training and the other in the Test Folder for Testing. A speech sample is plotted in the time domain as shown in figure 5.

Framing is done by obtaining the 256 x 129 matrix M containing all the frames. Windowing is done by obtaining the matrix M2 using the Hamming filter. A new matrix M3 is created where the column vectors are the FFTs of the column vectors of M2. Each column in M3 is a power spectrum representation of the original signal. The result obtained is shown in the figure 6.

In this plot, the areas containing the highest level of energy are displayed in red. As it can be seen on the plot, the red area is located between 0.3 and 0.7 seconds. The plot also shows that most of the energy is concentrated in the lower frequencies (between 50 Hz and 1 kHz).

Power spectrum of a speech file is computed and plotted using different frames sizes as shown in the figure. For N = 128 there is a high resolution of time and a poor frequency resolution. For N = 256 there is a compromise between the resolution in time and the frequency resolution. For N = 512 there is an excellent frequency resolution but the resolution in time is strongly reduced. The power Spectrums with different N is shown in the figure 7.

Mel-spaced filter bank is plotted as shown in the figure 8. The spectrum of a speech file before and after the mel-frequency wrapping step is computed and plotted as shown in the figure 9. As it can be seen in the first plot, most of the information is contained in the lower frequencies. This information has been extracted and amplified in the second plot. The second plot therefore shows the main characteristics of the speech signal.

To inspect the acoustic space (MFCC vectors) we picked the two dimensions (5th and the 6th) and plotted the data points in a 2D plane as shown in the figure 10.

Mostly the two areas overlap. But certain regions seem to be used exclusively by one or the other speaker. This is what will allow us to distinguish the different speakers. The points don't form actual clusters, but there are areas where the density of points is higher. The data points of the trained VQ codeword are plotted as shown in the figure 11.

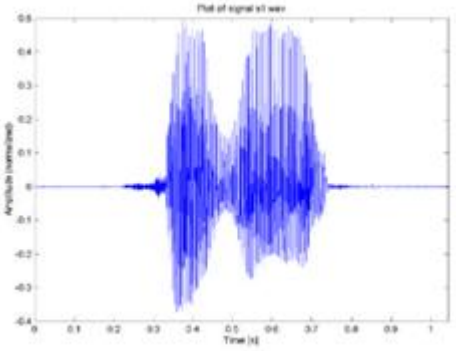

Figure 5. Plot of the Speech Signal

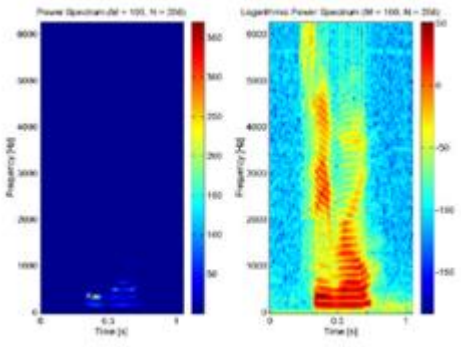

Figure 6. Power Spectrum of the speech Signal

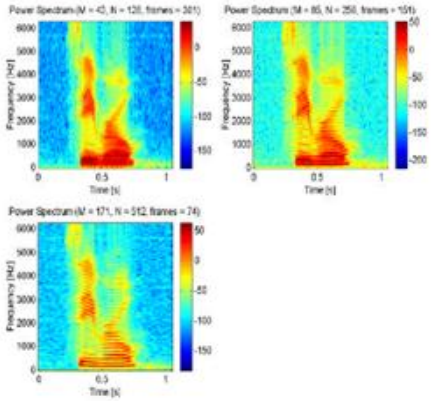

Figure 7. Power spectrum with different N.

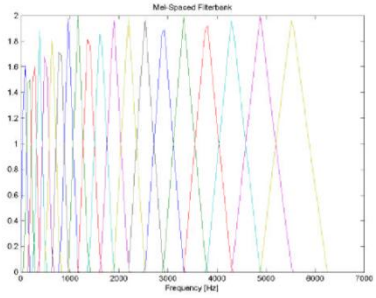

Figure 8. Mel Spaced Filter Bank

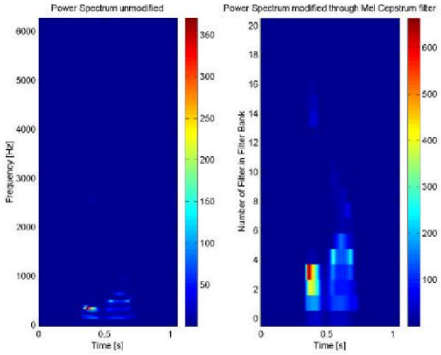

Figure 9. The spectrum of a speech file before and after the mel-frequency wrapping

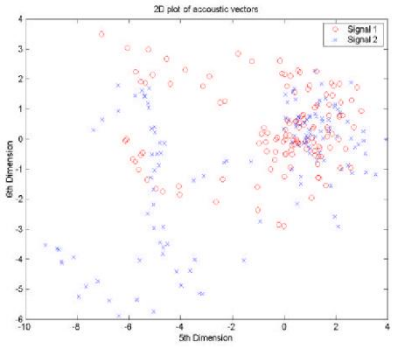

Figure 10. 2D plot of Acoustic Vectors

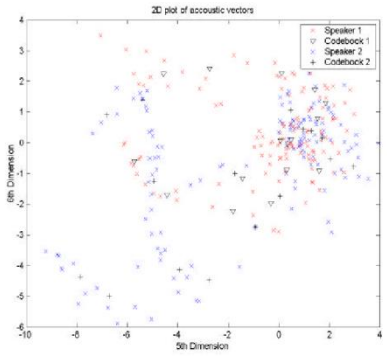

Figure 11. Data points of the trained VQ codeword

This system is able to recognize 7 out of 8 speakers. This is an error rate of 12.5%. The recognition rate of our system is much better than the one of a human's recognition rate. However you must be aware that this test is not really representative of the computer's efficiency to recognize voices because only 8 persons are tested, with only one training session and with only one word.

The result is:

Speaker 1 matches with speaker 1
Speaker 2 matches with speaker 2
Speaker 3 matches with speaker 7
Speaker 4 matches with speaker 4
Speaker 5 matches with speaker 5
Speaker 6 matches with speaker 6
Speaker 7 matches with speaker 7
Speaker 8 matches with speaker 8

6. Conclusions

The goal of this paper was to implement a text-independent speaker identification system. The feature extraction is done using Mel Frequency Cepstral Coefficients (MFCC). The speakers are modeled using Vector Quantization (VQ). Using the extracted features a codebook from each speaker was build by clustering the feature vectors. The clustering was done using the LBG algorithm. Codebooks from all the speakers were collected in a speaker database.

The experiments conducted showed that it was possible to obtain 100% identification rates for MFCC based features. From the results it can be said that VQ using cepstral features is an simple and efficient way to do speaker identification.

7. References

[1]Lasse L. Molgard, Kasper W. Jorgensen, Speaker Recognition: special course, IMM, DTU, December, 14, 2005.

[2] B. S. Atal, "Automatic Recognition of Speakers from their Voices",*Proceedings of the IEEE*, vol 64, 1976, pp 460 – 475.

[3] J. R. Deller, J. H. L. Hansen, J. G. Proakis, *Discrete-Time Processing of Speech Signals*, Piscataway (N.J.), IEEE Press, 2000.

[4] H. Gish and M. Schmidt, "Text Independent Speaker Identification", *IEEE Signal Processing Magazine*, Vol. 11, No. 4, 1994, pp. 18-32.

[5] L. Besacier, J.F. Bonastre, "Frame Pruning for Speaker Recognition", *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference*, Vol. 2, pp. 765-768.

[6] D.A. Reynolds, R.C. Rose, "Robust text-independent speaker identification using Gaussian Mixture speaker models", *IEEE Trans. on Speech and Audio Processing*, vol. 3, no. 1, 1995, pp. 72 -83.

[7] J. Picone, "Signal Modeling Techniques in Speech Recognition", *IEEE Proceedings*, vol. 81, no. 9, 1993, pp. 215-1247.

[8] J. Godfrey, D. Graff, A. Martin, "Public databases for speaker recognition and verification", *Proc. of the ESCA Workshop Automatic Speaker Recognition, Identification, Verification*, 1994, pp. 39-42.

[9] H.A. Murthy, F. Beaufays, L.P. Heck, M. Weintraub, "Robust text-independent speaker identification over telephone channels", *IEEE Trans. on Speech and Audio Processing*, vol. 7, no. 5 , 1999, pp. 554 -568.

[10] H.M. Torres, H. Rufiner, "Automatic speaker identification by means of Mel cepstrum, wavelets and wavelet packets", *Proc. of the 22nd Annual IEEE Intern. Conf. on Engineering in Medicine and Biology*, vol. 2, 2000, pp. 978–981.

[11] C. Miyajima, Y. Hattori, K. Tokuda, T. Masuko, T. Kobayashi, T. Kitamura, "Speaker identification using Gaussian mixture models based on multi-space probability distribution", *Proc. of ICASSP*, vol. 1, 2001, pp. 433 -436.

[12] J. M.Naik, "Speaker Verification: A Tutorial", *IEEE Communications Magazine*, January 1990, pp.42-48.

[13] S. Furui, *Digital Speech Processing, Synthesis and Recognition*, New York, Marcel Dekker, 2001.

[14] S. Molau, M. Pitz, R. Schluter, H. Ney, "Computing Mel-Frequency Cepstral Coefficients on the Power Spectrum", *Acoustics, Speech, and Signal Processing*, 2001 IEEE International Conference, Volume: 1, 2001, pp. 73-76

[15] C.-H. Lee, F.K. Soong, K.K. Paliwal: "Automatic Speech and Speaker Recognition" - advanced topics. Kluwer Academic Publishers, pp. 42-44, Norwell, Massachusetts, USA, 1996

[16] S. Sookpotharom, S. Manas "Codebook Design Algorithm for Classified Vector Quantization" Bangkok University, Pathumtani, Thailand pp. 751-753 2002

[17] M. Brooks, _Voicebox: Speech Processing Toolbox for MATLAB,_ http: //www.ee.ic.ac.uk/hp/staff/dmb/voicebox/.

[18] J. R. Deller, J. G. Proakis and J. H. L. Hansen, Discrete-time Pro-cessing of Speech Signals, Prentice Hall, New Jersey, 1993.

[19] M. N. Do, _Digital Signal Processing Mini-Paper: An Automatic Speaker Recognition System,_ http://lcavwww.epfl.ch/~minhdo/asr_paper/.

[20] L. Feng, Speaker Recognition, Master's thesis, Technical University of Denmark, Informatics and Mathematical Modelling, 2004, ISSN: 1601- 233X.

[21] J. P. C. Jr, _Speaker Recognition: A Tutorial,_ in Proceedings of the IEEE, vol. 85 no. 9, 1997.

[22] E. Karpov, Real-Time Speaker Identi_cation, Master's thesis, University of Joensuu Department of Computer Science, 2003.