Journal of Information Technology and Engineering
Vol. 1 No. 2 (December, 2016)

Speaker Recognition Using Mel Frequency Cepstrum

Othman O. KHALIFA, S. KHAN, MD. Rafidul ISLAM, M. FAIZAL, and D. DOL

Electrical and Computer Engineering, International Islamic University Malaysia,Jalan Gombak
Kuala Lumpur, Malaysia *E-mail: khalifa@iiu.edu.my*

Received: 15th April 2016 Revised: 10th May 2016 Accepted: 17th December 2016

Abstract: *Automatic speaker recognition is a field of study attributed in identifying a person from his or her voice. It is widely used in biometric security system, phone banking and other relevant applications. In this context, this project involves on the development of a Matlab based text independent speaker recognition system. A literature review is carried out to identify techniques as well as theories behind speaker recognition. From the literature review, a model is developed that simulates the biological speech production system in human being. This model is used to derive a mathematical representation of the speech signal. Feature extraction method known as Mel Frequency Cepstrum Coefficient (MFCC) is used to extract speaker discriminative features from the mathematical representation of the speech signal. After that a feature matching method known as Vector quantization is implemented using the LBG Algorithm. Feature matching is carried out in order to cluster the speech features into groups of specific sound classes. This step is required in order to realize a text independent system. Finally analysis is carried out to identify parameter values that could be used to increase the accuracy of the system.*

Keywords: *MFCCÿfeature matchingÿAutomatic speaker recognition*

1. INTRODUCTION

Automatic speaker recognition is the use of a machine (i.e. software or hardware) to recognize a person from a spoken phrase. This is done by identifying characteristic acoustic features of their voice which is unique. This makes it possible to use the speaker's voice to verify their identity and can be used to control access to services such as voice dialing, banking by telephone, database access service, security control for confidential information area and remote access to computers. Basically prior to a verification or identification session, users that are authorized to use the system will need to undergo enrollment session, where they provide speech samples that will be processed and stored in the database in order to be used later during the identification process. Here only certain important features that are unique to individuals are extracted while other redundancies are discarded.

The use of personal features, unique to all human being to identify or to verify a person's identity is a field that is being actively researched. This can be seen in the usage of finger print, retina pattern and voice pattern in various applications to identify or to verify the identity of an individual. The use of fingerprint is perhaps one of the earliest applications of biometrics or personal features, to identify a person. Here in this review the focus is more on voice identification. Speaker or voice verification/ identification can be classified as shown in Fig. 1. The speaker recognition system consists of the following steps: Speaker identification; in this system, when a user inputs a test utterance, the system will identify which of the speaker made the utterance according to the speech patterns stored in the database. Therefore here the output will be either the name of the user or the system will reject if the users speech pattern is not in the database.

1.1 Speaker Verification

In this system, the user inputs a test utterance together with his or her ID number. Here the system verifies a person‘s identity claim by comparing the sample of their speech stored in the database to that of the claimed identity. The expected result is accept or reject the identity claim.

1.2 Text Independent

When a system is said to be text independent, the user can utter any text during the input of test utterance. Here, different text uttered by the same person will produce the same pattern. Therefore user will have to provide only the voice sample during the verification process.

1.3 Text Dependent

Here the restriction is that the text uttered during the test session should be identical to the one stored in the database. This is because the pattern extracted from the speech sample takes into account the word or text that is being uttered, therefore different text uttered from the same person will have different pattern. Therefore here the user needs to provide his voice sample as well as the appropriate password.

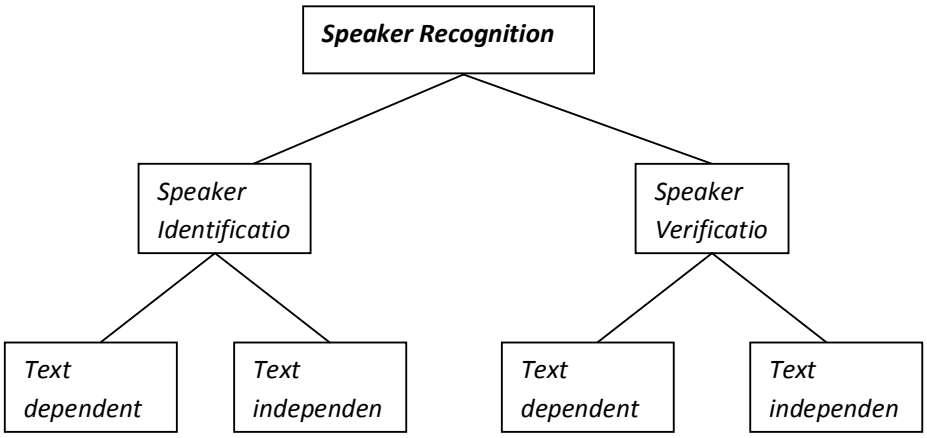

Figure 1: Classification of Speaker Recognition

2. FEATURE EXTRACTION

The main objective of feature extraction is to extract characteristics from the speech signal that are unique to each individual which will be used to differentiate speakers. In this context, mathematical representation of a speech signal developed in 2.2.1, reveals that speech signal x[n] is composed of convolution between glottal pulse u[n] and the vocal tract impulse response h[n] as shown in equation (2.1). Since the characteristic of the vocal tract is unique for each speaker, the vocal tract impulse response can be used to discriminate speakers.

Therefore in order to obtain the vocal tract impulse response from the speech signal, a deconvolution algorithm known as the Mel Frequency Cepstrum Coefficient (MFCC) is applied. This algorithm transforms the speech signal which is the convolution between glottal pulse and the vocal tract impulse response into a sum of two components known as the cepstrum that can be separated by band pass linear filters, if there is no frequency overlapping. For speech signal, this is feasible since analysis have shown both the glottal pulse and the vocal tract impulse response has different spectral variation or frequency [1, 2].

2.1 The Mel Frequency Cepstrum Coefficient

The Mel Frequency Cepstrum Coefficient (MFCC) is used to resolve the speech signal into sum of two components. This computation is carried out by taking the inverse DFT of the logarithm of the magnitude spectrum of the speech frame. This can be shown below (2.2):

$$x\hat{}[n] = \text{IDFT}\{\log(\text{DFT}\{h[n]*u[n]\})\}$$
$$= h\hat{}[n] + u\hat{}[n] \quad (2.2)$$

Where $h\hat{}[n]$, $u\hat{}[n]$ and $x\hat{}[n]$ are the complex cepstrum of h[n], u[n] and x[n] respectively.

From (2.2), the convolution of the two components is changed to multiplication when Fourier transform is performed. Then by taking logarithm, the multiplication is changed to addition. This is basically how the complex cepstrum $x\hat{}[n]$ is obtained. Fig. 3 illustrates the MFCC computation steps.

2.2 Preprocessing

The first step of speech signal processing involves the conversion of analog speech signal into digital form. This is a crucial first step in order to enable further processing. Here the continuous time signal (speech) is sampled at discrete time points to form a sample data signal representing the continuous time signal. After sampling the signal, samples are quantized to produce a digital signal [5].

The method of obtaining a discrete time representation of a continuous time signal through periodic sampling, where a sequence of samples, x [n] is obtained from a continuous time signal x (t), according to the relationship stated in (2.3).

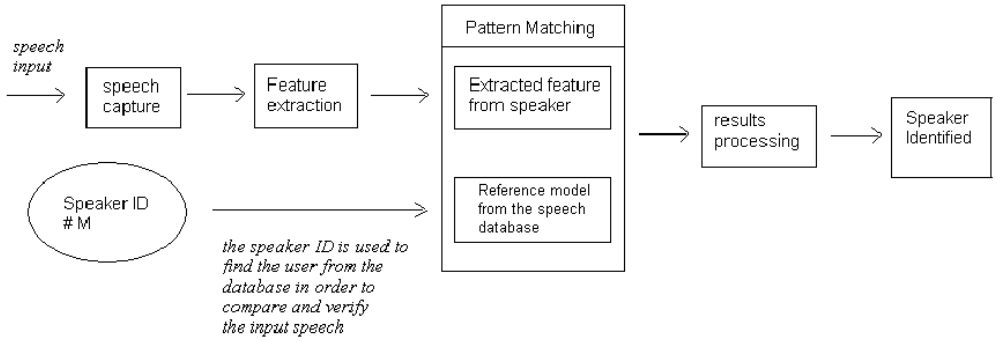

Figure 2: Functional Block Diagram of Automatic Speaker Recognition System Implementing Speaker Identification

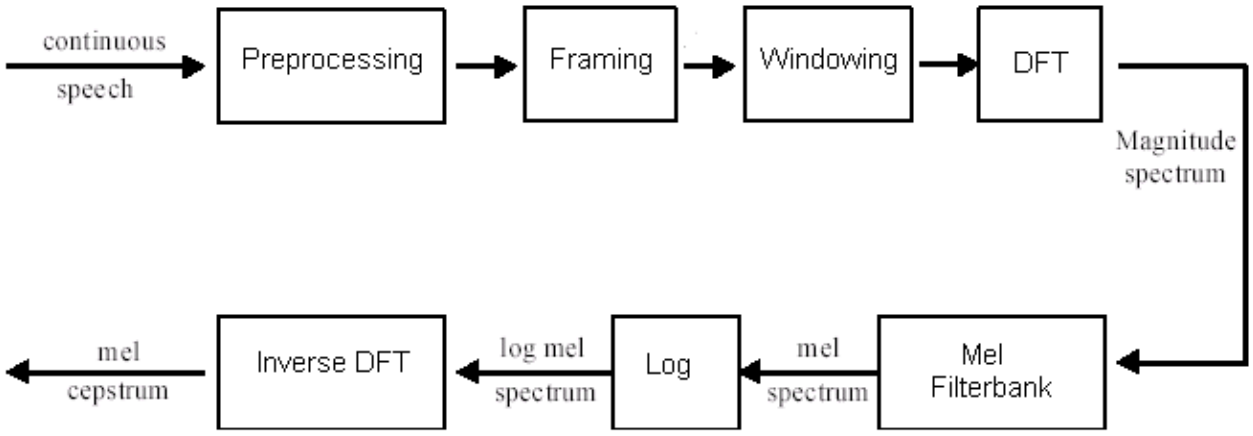

Figure 3: MFCC Computation

$$x[n] = x(nT) \quad (2.3)$$

In equation (2.3), T is the sampling period and 1/T = fs is the sampling frequency, in samples / second [3].

It is apparent that more signal data will be obtained if the samples are taken closer together (by making T smaller). But the problem here is that, what is the required sampling rate in order for a continuous signal to be well represented in digital form without any significant loss of information. This can be solved if the highest frequency of a particular signal is known and the minimum sampling rate can be obtained from the uniform sampling theorem which states that:

If the highest frequency contained in an analog signal x(t) is fmax, then x(t) is completely specified by samples taken at the uniform rate fs ≥ 2 x fmax. The minimum sampling rate, fs = 2fmax, is referred to as the nyquist rate. Signal sampled at a rate less than nyquist rate is referred to as undersampling and causes the aliasing effect. Aliasing occurs when the higher frequency signal components appear as lower frequency components which cause the spectrum of the signal to overlap [5].

The size of the sample for a digital signal is determined by the sampling frequency and the length of the speech signal in seconds. For example if a speech signal is recorded for 5 seconds using sampling frequency of 10000 Hz, the number of samples = 10000 x 5s = 50000 samples.

Basically the analog to digital conversion as well as the noise filtering is performed when the voice is input using the microphone and processed using the audio / sound card in the computer.

2.3 Framing

Framing is the process of segmenting the speech samples obtained from the A/D conversion into small frames with time length in the range of 20 to 40 ms.

From the discussion in 2.2 on the human speech production, speech signal is known to exhibits quasi-stationary behavior over a short period of time (20–40ms). Therefore framing enables the non-stationary speech signal to be segmented into quasi-stationary frames.

The purpose of framing is to enable fourier transform of the speech signal. This is because a single fourier transform of the entire speech signal cannot capture the time varying frequency content due to the non-stationary behavior of the speech signal.

Therefore fourier transform is performed on each segments separately [7].

If the frame length is not too long (20–40ms), the properties of the signal will not change appreciably from the beginning of the segment to the end. Thus, the DFT of a windowed speech segment should display the frequency–domain properties of the signal at the time corresponding to the window location.

If the frame length is long enough so that the harmonics are resolved (>80ms), the DFT of a windowed segment of voiced speech should show a series of peaks at integer multiples of the fundamental frequency of the signal in that interval. This would normally require that the window span several periods of the waveform.

If the frame is too short (<10ms), then the harmonics will not be resolved, but the general spectral shape will still be evident. This is typical of the trade – off between frequency resolution and time resolution that is required in the analysis of non stationary signals [3].

2.4 Windowing

In all practical signal processing application, it is necessary to work with short term or frames of the signal. This is to select a portion of the signal that can reasonably be assumed stationary. In the previous step a speech sample is segmented into frames with quasi stationary speech segments. The next step is to perform windowing on the speech segments. Windowing is performed to avoid unnatural discontinuities in the speech segment and distortion in the underlying

spectrum. This is to ensure that all parts of the signal are recovered and possible gaps between frames are eliminated [7, 3]. Some of the most commonly used windows are Kaiser, hamming, hanning and Blackman window. These windows are symmetric about the time (N-1) / 2, where N is the duration of the window. Figure 4 shows the time plot of these windows.

The choice of the window is a trade off between several factors [1]:

The window shape may reduce distortion, but it may increase signal shape alteration. The length N is proportional to the frequency resolution and inversely proportional to the time resolution.

In speaker recognition, the most commonly used window shape is the hamming window, whose impulse response is a raised cosine impulse (2.4) :

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left[2\pi n / N - 1\right] & n = 0,\ldots,N-1 \\ 0 \quad \text{otherwise} \end{cases} \tag{2.4}$$

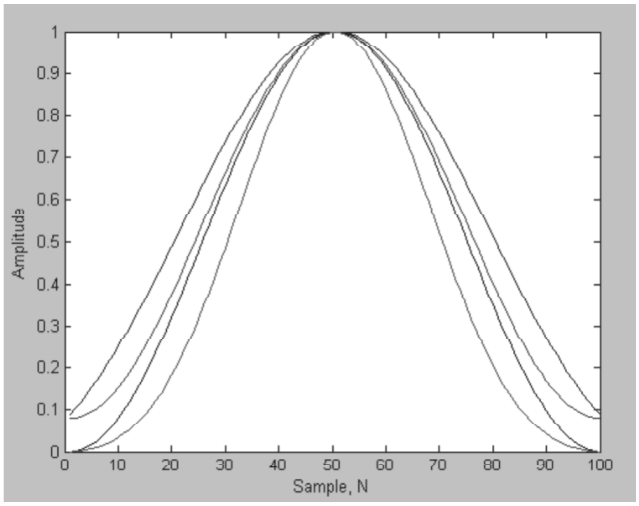

Figure 4: Comparison between Various Windows

The side lobes of this window are much lower that the rectangular window, which means leakage effect is decreased, although resolution is also appreciably reduced [1].

2.5 Discrete Fourier Transform

Fourier series enables a periodic function to be represented as a sum of sinusoids and to transform a function in the time domain into the frequency domain. The same analysis can be carried out on non periodic functions using fourier transform. Therefore this is why fourier transform is used due to the non periodic behavior of the speech signal [9].

Referring back to 2.3, The basis of performing fourier transform is to convert the convolution of the glottal pulse u[n] and the vocal tract impulse response h[n] in the time domain into multiplication in the frequency domain. This can be supported by the convolution theorem (2.5) [9].

If X(w), H(w) and Y(w) are the fourier transform of x(t), h(t) and y(t) respectively, then

$$Y(w) = FT\left[\, h(t) * x(t) \,\right] = H(w) \times X(w) \tag{2.5}$$

In analyzing speech signals, discrete fourier transform is used instead of fourier transform, because the speech signal is in the form of discrete number of sample due to preprocessing. The discrete fourier transform is represented by the equation (2.6)

Where X(k) is the fourier transform of x(n).

$$X(k+1) = \sum_{n=0}^{N-1} x(n+1) W_N^{kn}$$

Where $W_N = e^{-j\left(\frac{2\pi}{N}\right)}$ (2.6)

2.6 Mel Filterbank

The information carried by low frequency components of the speech signal is more important compared to the high frequency components. In order to place more emphasize on the low frequency components, mel scaling is performed.

A mel scale is a unit of special measure or scale of perceived pitch of a tone. It does not correspond linearly to the normal frequency, but behaves linearly below 1 kHz and logarithmically above 1 kHz. This is based on studies of the human perception of the frequency content of sound. The equation (2.7) shows the relationship between frequency in hertz and mel scaled frequency [2, 7].

Frequency (mel scaled) = 2595 log (1 + f (Hz) / 700) (2.7)

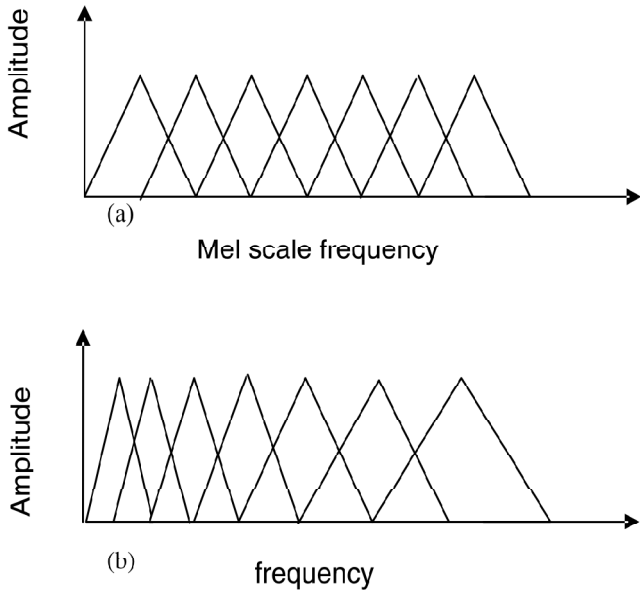

Figure 5: (a) The Filter Bank in Mel Scale Frequency (b) The Filterbank Representation in Normal Frequency

From Figure 5 (a), in the mel frequency scale, it can be seen that the center frequency of each triangular filter is spaced uniformly with respect to each other. But the spacing between the centre frequency increases logarithmically when the scale is changed to Hertz as shown in Figure 5 (b).

To implement this filterbank, the magnitude coefficient of each fourier transformed speech segment is binned by correlating them with each triangular filter in the filterbank. Here binning means that each fourier transformed magnitude coefficient is multiplied by the corresponding filter gain [10].

2.7 Logarithm

From the discussion in 2.3.1, it is known that logarithm has the effect of changing multiplication into addition. Therefore this step simply converts the multiplication of the magnitude of the fourier transform into into addition.

2.8 IDFT

Earlier discussion in 2.3 revealed that the speech signal is represented as a convolution between slowly varying vocal tract impulse response and quickly varying glottal pulse. Therefore by taking the inverse DFT of the logarithm of the magnitude spectrum, the glottal pulse and the impulse response can be separated.

3. FEATURE MATCHING

In the previous section 2.3, the feature extraction process was discussed where speaker discriminative features are extracted from the speech signal. In this section, the classification or clustering method known as vector quantization is discussed. This method is part of the decision making process of determining the identity of a speaker based on previously stored information. This step is basically divided into two parts, namely training and testing. Training is a process of enrolling or registering a speaker to the identification system database by constructing a model of the speaker based on the features extracted from the speaker‘s speech sample. Testing is a process of computing a matching score, which is the measure of similarity of the features extracted from the unknown speaker and the stored speaker models in the database. The unknown speaker is identified as the person having the minimum matching score in the database.

3.1 Vector Quantization

Vector quantization (VQ) is a lossy data compression method based on the principle of block coding. It is a fixed-to-fixed length algorithm. In the earlier days, the design of a vector quantizer is considered to be a challenging problem due to the need for multi-dimensional integration. In 1980, Linde, Buzo, and Gray (LBG) proposed a VQ design algorithm based on a training sequence. The use of a training sequence by passes the need for multi-dimensional integration. A VQ that is designed using this algorithm are referred to in the literature as an LBG -VQ. A VQ is nothing more than an approximator. The idea is similar to that of rounding off (say to the nearest integer). An example of a 1 dimensional VQ is shown in Figure 6.

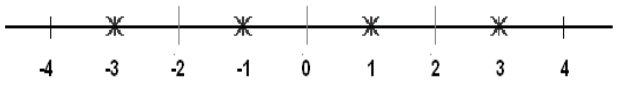

Figure 6: 1 Dimensional VQ

In Figure 6 every number less than -2 is approximated by -3. Every number between -2 and 0 are approximated by -1. Every number between 0 and 2 are approximated by +1. Every number greater than 2 are approximated by +3.

An example of a 2-dimensional VQ is shown in Figure 7:

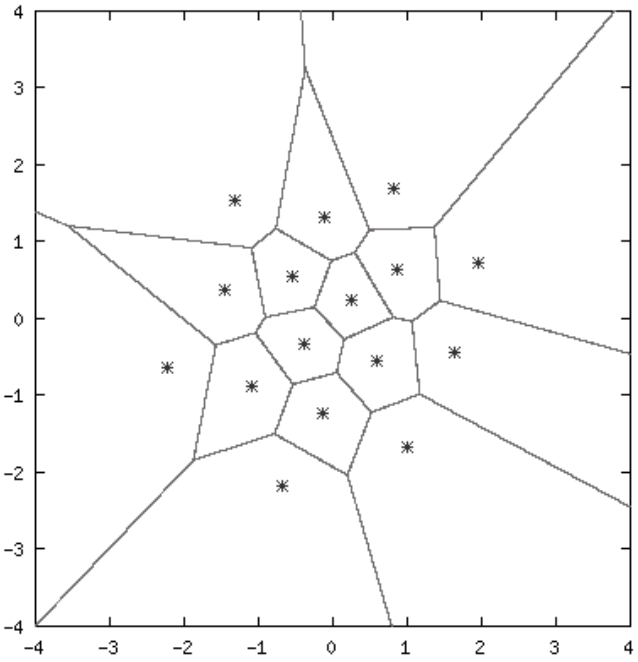

Figure 7: 2-dimensional VQ

Here, every pair of numbers falling in a particular region are approximated by a star associated with that region. Note that there are 16 regions and 16 stars. Therefore the 2 dimensional vector in this case are clustered into 16 different cluster or codevectors.

In automatic speaker recognition, vector quantization is used to cluster or group together feature vectors extracted from the speech sample according to their sound classes as discussed in section 2.2 , i.e. quasi periodic, noise like and impulse like sound. This can also be seen in figure 2.2. Hence each cluster or centroid represents a different class of speech signal. This enables a text independent speaker recognition system to be realized because the speech vectors are not clustered according to the spoken words but clustering is based on sound classes [7]. In the testing or identification

session, the euclidean distance between the feature vector and codebook for each speaker is calculatedand the speaker with the smallest average minimum distance is picked.

During the training session, a speaker is modeled as a set of feature vectors generated from his / her voice sample. The speaker models are constructed by clustering the feature vectors in K separate clusters. Each clusters is then represented by a code vector, which is the centroid (average vector) of the cluster. The resulting set of code vectors is called a codebook, and it is stored in the speaker database. Therefore each speaker have theirs own set of codebook stored in the database. In the codebook, each vector represents a single acoustic unit typical for the particular speaker.

The matching of an unknown speaker is then performed by measuring the euclidean distance between the feature vector of the unknown speaker to the model (codebook) of the known speakers in the database. The goal is to find the codebook that has the minimum distance measurement in order to identify the unknown speaker.

4. SOFTWARE SIMULATION AND ANALYSIS

In this chapter, the Automatic Speaker Recognition (ASR) software that has been developed using Matlab 6.5 is demonstrated. Figure 8 shows the main command window for the software.

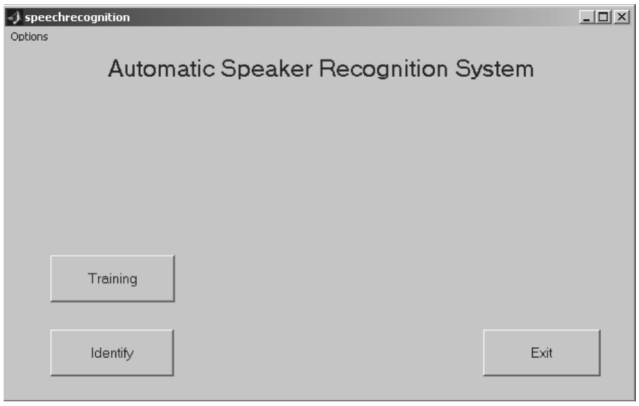

Figure 8: ASR Main Command Window

4.1 Registering a New User

The new user registration dialog box is displayed when the training button in the main command window (Figure 8) is clicked. Figure 9 shows this dialog box.

The dialog box prompts the user to enter a 6 character ID. This ID will be used as a tag for the speech feature that will be stored in the database. If the user enters an ID that already exists in the database an error message is displayed as shown in Figure 10.

When the start recording button is clicked, the system will start recording the user speech sample. If there is an error in the input speech, an error message prompting the user to provide another speech sample is displayed as shown in Figure 11

Figure 9: New User Registration Dialog Box

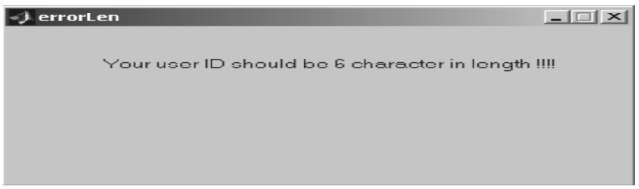

Figure 10: ID Error Message

Figure 11: Input signal error message

When the registration process is successful, a message appears as shown in Figure 12.

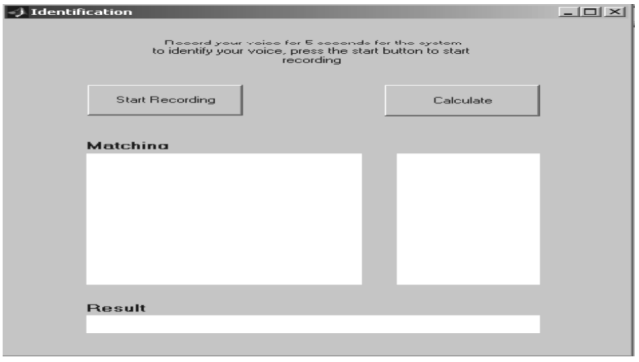

Figure 12: Registration Complete Message

4.2 Speaker Identification

The speaker identification dialog box as shown in Figure 13 appears when the identify button on the main command window is clicked. Here the user will be prompted to provide a speech sample in order for the system to identify the user.

When the speech sample has been provided, clicking the calculate button in Figure 13 will display the matching score as well as the identified user. The user is identified by selecting the minimum matching score. This is shown in Figure 14.

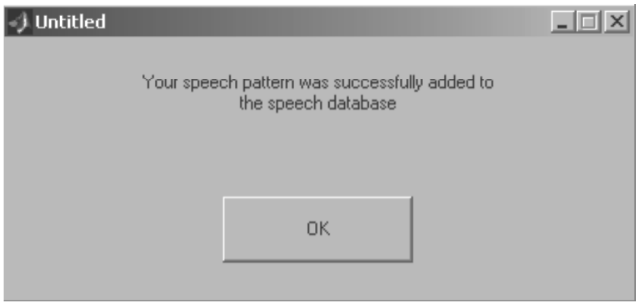

Figure 13: Speaker Identification Dialog Box

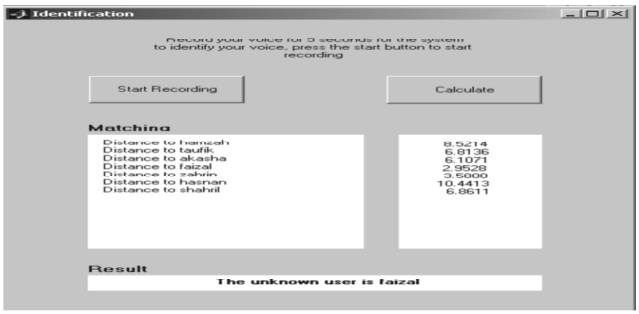

Figure 14: Matching Score Calculation and Identified User

4.3 Analysis

The ASR software is designed in such a way that the user can adjust certain parameters that is associated to the feature extraction and matching process. This enables analysis to be carried to study the effect of changing these parameters in the accuracy of the system. Figure 15 shows the options that could be selected when the option tab on the main command window is clicked. The following sections 4.3.1 and 4.3.2 shows the effect of changing the parameters associated to MFCC and vector quantization on the efficiency of the system. The methodology used to calculate the efficiency of the system is by measuring the difference in distance measurement between two users when one of the users is being identified. Higher matching score difference shows higher efficiency because this enables the discriminative ability of the system to increase.

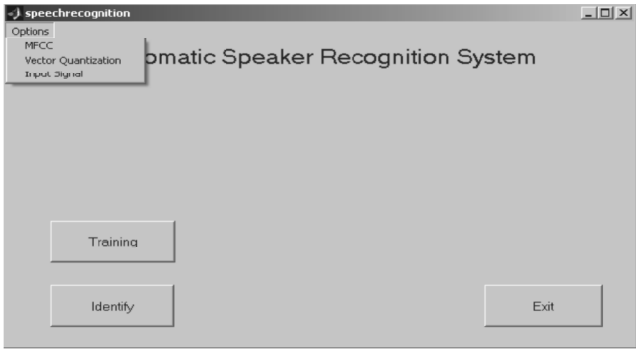

Figure 15: Speaker Recognition Software Options

4.3.1 Mel Frequency Cepstrum Coefficient

The MFCC option dialog box enable the user to adjust parameters such as number of frames, frame size (number of samples per frame) and number of filters in mel filterbank. This is shown in Figure 16.

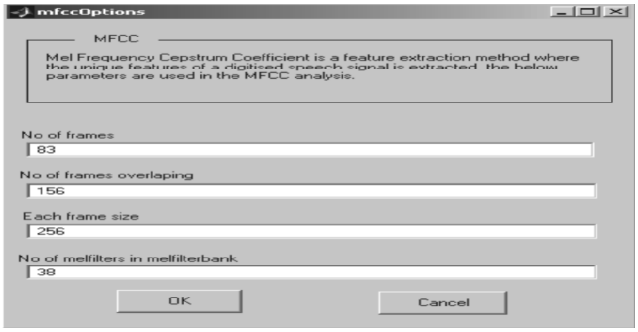

Figure 16: MFCC Options Dialog Box

4.3.1.1 Frame Size

The frame size is changed from 200 to 500 samples per frame. Other parameters are left unchanged. Distance measurement for each case is shown in Table 1. Figure 17 shows the matching score difference as a function of frame size.

Table 1
Distance Measurement for Different Frame Size

Number of samples per frame	*Speech pattern in database for speaker 1*	*Distance measurement (matching score)*
200	Speaker 1	3.29750
	Speaker 2	4.58755
	Difference	**1.29005**
300	Speaker 1	3.53254
	Speaker 2	4.91759
	Difference	**1.38505**
400	Speaker 1	3.49640
	Speaker 2	5.10682
	Difference	**1.61042**
500	Speaker 1	3.26857
	Speaker 2	4.40645
	Difference	**1.13788**

Fixed parameters	
Number of frames	120 frames
Number of samples overlapping	70 samples
Number of mel filters	20
Number of code vector	12
Splitting parameter	0.02
Sampling frequency	10000 Hz

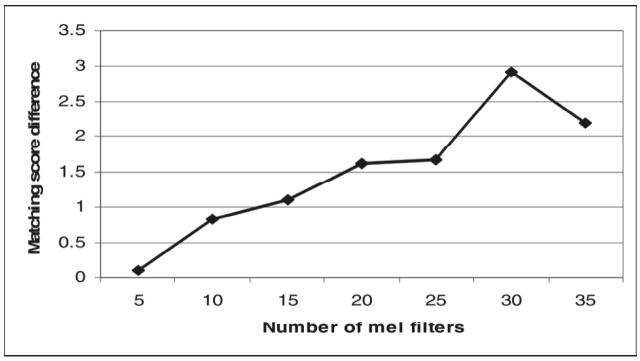

Figure 17: Matching Score Difference as a Function of Frame Size

From the Figure 17, it can be seen that the matching score difference which is associated to the identification efficiency increases when the frame size which represents the number of samples per frame is increased. The efficiency reaches its highest point when the frame size is increased up until 400 samples per frame. Further increase in the frame size causes the efficiency to decrease. The frame size from 200 to 400 represents a time length of 20 to 40 ms. From section 2.2, it is known that the speech signal shows quasi – stationary behavior for the time interval of 20 to 40 ms. Therefore for the frame size higher than 400, the speech segment shows non stationary behavior and this causes the efficiency to decrease.

4.3.1.2 Mel Filterbank Size

The size of mel filterbank is changed from 5 to 35 and the difference in matching score is measured. Table 2 shows the effect of changing the number of mel filters on the matching score difference. Figure 18 shows the matching score difference as a function of mel filterbank size.

Table 2
Distance Measurement for Different Filterbank Size

Number of melfilters	*Speech pattern in database*	*Distance measurement (matching score) for speaker 1*
5	Speaker 1	0.97172
	Speaker 2	1.0793
	Difference	**0.10758**
10	Speaker 1	1.87023
	Speaker 2	2.70960
	Difference	**0.83937**
15	Speaker 1	2.58963
	Speaker 2	3.69096
	Difference	**1.10133**
20	Speaker 1	3.49640
	Speaker 2	5.10682
	Difference	**1.61042**
25	Speaker 1	3.59349
	Speaker 2	5.26192
	Difference	**1.66843**
30	Speaker 1	3.98954
	Speaker 2	6.90351
	Difference	**2.91397**
35	Speaker 1	4.82648
	Speaker 2	7.02025
	Difference	**2.19377**

Fixed parameters	
Number of frames	120 frames
Frame size (samples / frame)	400 samples
Number of samples overlapping	70 samples
Number of code vector	12
Splitting parameter	0.02
Sampling frequency	10000 Hz

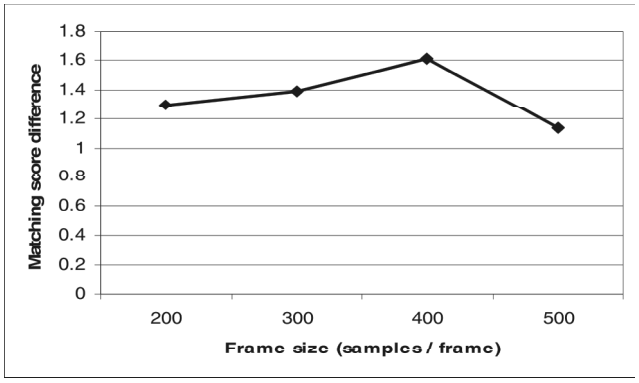

Figure 18: Matching Score Difference as a Function of Mel Filterbank Size

4.3.2 Vector Quantization

The vector quantization dialog box enables the user to adjust parameters such as number of code vectors and splitting parameter. This is shown in Figure 19.

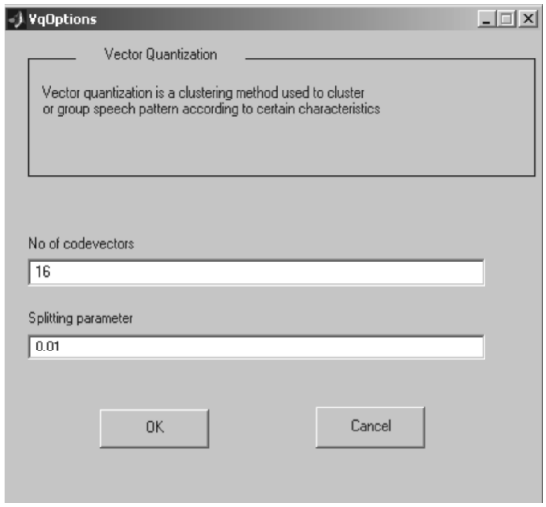

Figure 19: VQ Options Dialog Box

4. CODE VECTOR SIZE

The code vector size is changed from 1 to 18 to identify the effect on the efficiency. The methodology used is the same as in the previous section. Table 3 shows the distance measurement. Figure 20 shows the matching score difference as a function of code vector size.

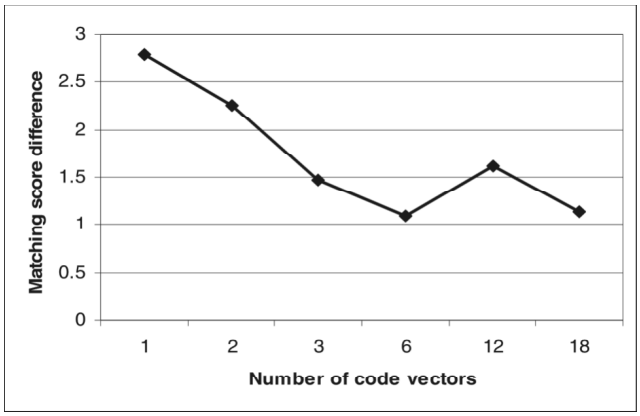

Figure 20: Matching Score Difference as a Function of Codevector Size.

Table 3
Distance Measurement for Different Codevector Size

Number of codevector	*Speech pattern in database*	*Distance measurement For speaker 1*
1	Speaker 1	1.62882
	Speaker 2	4.40978
	Difference	**2.78096**
2	Speaker 1	1.95466
	Speaker 2	4.20694
	Difference	**2.25228**
3	Speaker 1	2.39033
	Speaker 2	3.85462
	Difference	**1.46429**
6	Speaker 1	3.34617
	Speaker 2	4.44338
	Difference	**1.09721**
12	Speaker 1	3.49640
	Speaker 2	5.10682
	Difference	**1.61042**
18	Speaker 1	3.56294
	Speaker 2	4.70508
	Difference	**1.14214**

Fixed parameters	
Number of frames	120 frames
Frame size (samples / frame)	400 samples
Number of samples overlapping	70 samples
Number of melfilters	20
Splitting parameter	0.02
Sampling frequency	10000 Hz

4.3.2.2 Splitting Parameter

The splitting parameter is changed from 0.01 to 0.04 to investigate the effect on the efficiency. This is shown in Table 4. Figure 21 shows the matching score difference as a function of splitting parameter.

Table 4
Distance Measurement for Different Splitting parameter

Splitting parameter	*Speech pattern in database*	*Distance measurement For speaker 1*
0.01	Speaker 1	3.10937
	Speaker 2	4.37776
	Difference	**1.26839**
0.02	Speaker 1	3.49640
	Speaker 2	5.10682
	Difference	**1.61042**
0.03	Speaker 1	3.05611
	Speaker 2	4.33347
	Difference	**1.27736**
0.04	Speaker 1	3.04985
	Speaker 2	4.22286
	Difference	**1.17301**

Fixed parameters	
Number of frames	120 frames
Frame size (samples / frame)	400 samples
Number of samples overlapping	70 samples
Number of melfilters	20
Number of codevector	12
Sampling frequency	10000 Hz

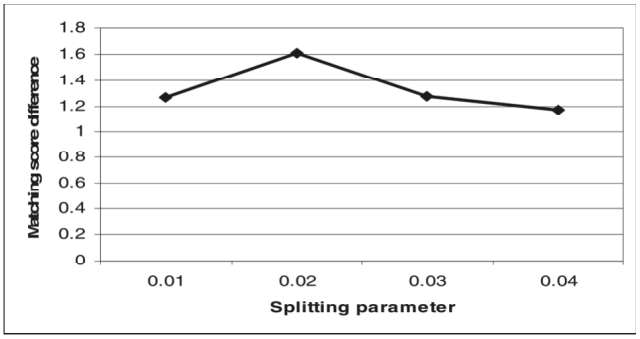

Figure 21: Matching Score Difference as a Function of Splitting Parameter

4.3.3 Input Signal

The input signal dialog box enables the user to adjust parameters such as sampling frequency and recording time for training and identification. This is shown in Figure 22.

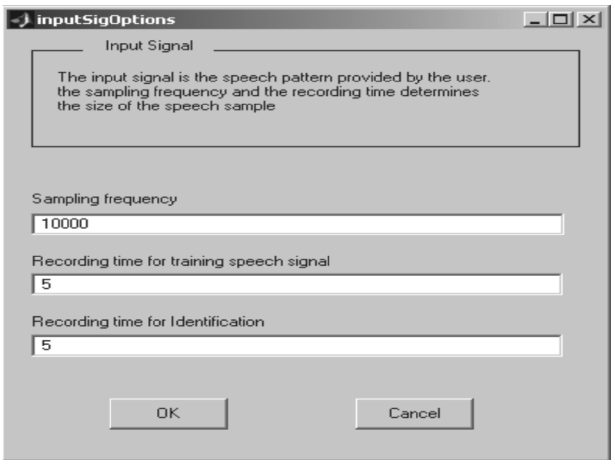

Figure 22: Adjust Parameters Such as Sampling Frequency and Recording Time for Training and Identification

4.4 Result

Identification is carried out by using 10 different speakers. First each of these speaker's speech pattern is stored in the speech database. Then Identification is carried out for each of these speakers. The phrase uttered during the registration session is "International Islamic University Malaysia". And during the identification session, the speakers are free to utter any phrase. The results are as shown in Table 5.

4.5 Database Setup and Configuration

In ASR software, two different databases are used to store the user ID and the codebook for each registered user respectively. For the user ID, Microsoft access database is used. While the codebook of each user is stored in the form of Lotus 1-2-3 file format. The filename used to save the codebook is the same as the user ID. Therefore prior to using the ASR software both Microsoft Access and Lotus 1-2-3 should be installed in the PC.

4.5.1 Codebook Database

The codebook for each registered user is saved in a folder named 'c:\f_codebook\' in the *c* drive. The user must make sure that a folder is created in the c drive using the above mentioned name to avoid any error when running the ASR software. Lotus 1-2-3 file format is used to save the codebook.

4.5.2 User ID Database

The user ID database uses Microsoft Access to save the ID of a registered user. Before the database could be accessed from the ASR software, a pointer or an interface is created to enable Matlab to have access to the database. The following step shows how to establish this interface.

To set up the data source:

(1) From the Windows Start menu, select Control panel -> Administrative Tools -> Data Sources (ODBC). The ODBC Data Source Administrator dialog box appears, listing any existing data sources. This is shown in Figure 24.

(2) Select the User DSN tab. A list of existing user data sources appears.Click Add and a list of installed ODBC drivers appears in the Create New Data Source dialog box. This is shown in Figure 25.

(3) Select Microsoft Access Driver (*.mdb) and click finish. The ODBC Setup dialog box appears as shown in Figure 26 for the selected driver. Use the name 'cbook' for the data source name. Then in the ODBC Setup dialog box, click Select. The Select Database dialog box appears as shown in Figure 26.

(4) In the select database dialog box as shown in Figure 27, select the 'userIDdbase' database in the ASR Software folder. Click OK to close the Select Database dialog box. Then Click OK to close the ODBC Data Source Administrator dialog box. The user ID database is now ready to be used with the ASR software.

Table 5
Matching Score and Result

Database speech pattern	*Test speaker's matching score* *Hamzah*	*Faizal*	*Zahrin*	*Aqasha*	*Taufik*	*Hasnan*	*Shahril*	*Daman*
Hamzah	**5.8120**	7.5655	6.1131	10.2318	8.2844	5.3211	6.1510	7.7343
Faizal	6.3259	**6.1165**	5.5116	5.0563	6.5766	4.5215	6.8495	8.5216
Zahrin	4.2219	7.4561	4.6528	5.6158	5.1421	7.3212	9.1347	7.9900
Aqasha	10.8213	6.8946	**3.9461**	4.6517	5.9576	6.8465	**5.4116**	7.5151
Taufik	6.0086	8.9411	8.4653	4.7329	**4.9920**	4.1646	7.0439	8.4916
Hasnan	6.2046	6.9633	8.1556	8.1412	7.9865	**3.6791**	9.8494	6.4813
Shahril	6.3371	6.3510	5.4114	**4.6514**	5.1546	7.8465	5.8017	6.1647
Daman	7.9618	6.9402	4.8165	9.5321	7.1911	5.4989	10.5649	**6.0047**
Result	pass	pass	fail	fail	pass	pass	fail	pass

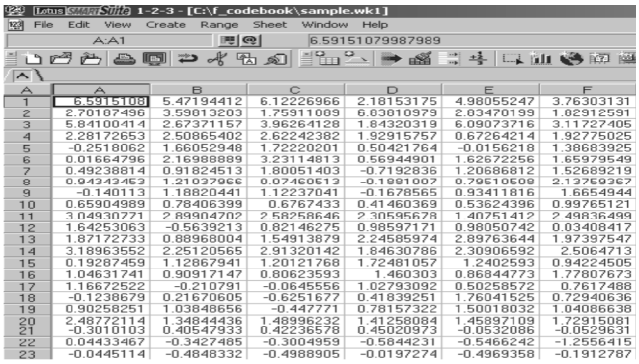

Figure 23: Sample codebook in Lotus 1-2-3 format

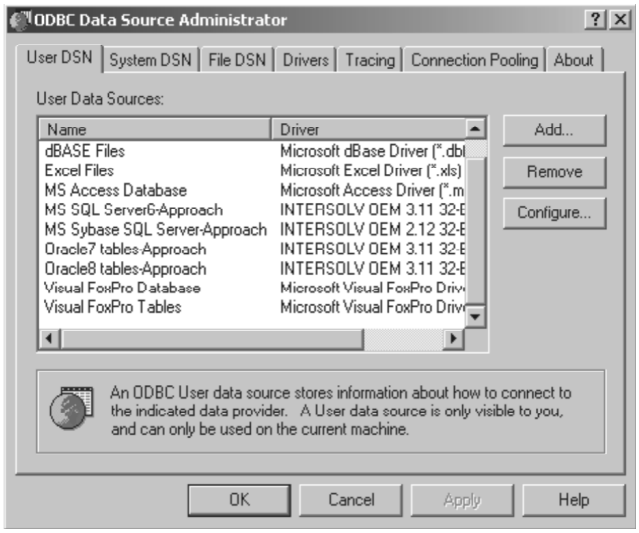

Figure 24: ODBC Data Source Administrator

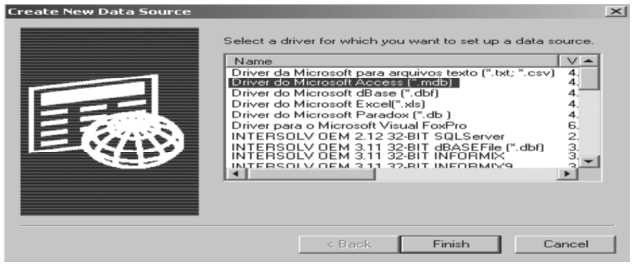

Figure 25: Create New Data Source

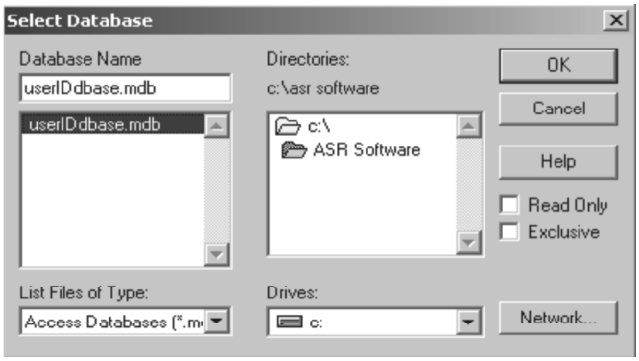

Figure 26: ODBC Microsoft Access Setup

5. CONCLUSION AND RECOMMENDATION

The most important steps in implementing Automatic speaker recognition is the feature extraction and matching. Feature extraction is used to extract speaker discriminative feature from the input speech signal, this is discussed in 2.3. Then feature matching is used to cluster the extracted features into its sound classes as discussed in 2.2 and shown in Figure 3. The outcome of feature extraction is a speaker model that can be used to identify a user.

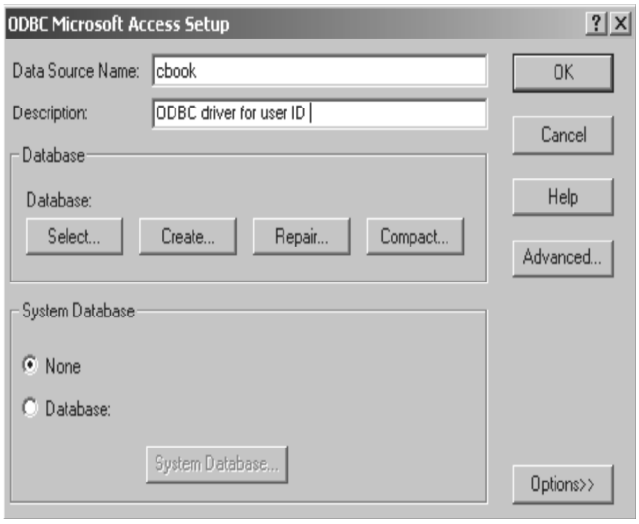

Figure 27: Select Database Dialog Box

The accuracy of the identification process can be influenced by certain factors such as different level of surrounding noise during the registration and identification session, the quality of the microphone used to input the speech signal, the health condition of the speaker i.e. sore throat and many other factors. Even though it is difficult to avoid some of these factors, steps should be taken to minimize the effect.

From the analysis that was carried out in section 4.3, parameter values that gives the highest identification efficiency was found.

Further research could be carried to develop a system that could identify the gender of a person from the speech pattern. The ASR software developed in this project could be used to accomplish this task by making some adjustments in the parameters used for feature extraction and matching. This is why the ASR software is designed with options as discussed in 4.3. Furthermore, various other analyses could be carried out such as age and emotion identification of a speaker from the voice pattern.

REFERENCES

[1] Claudio Becchetti and Lucio Prina Ricotti, *Speech Recognition*, John Wiley and Sons, England, 1999.

[2] Evgeny Karpov, "Real Time Speaker Identification," Master`s thesis, Department of Computer Science, University of Joensuu, 2003

[3] Alan V. Oppenheim and Ronald W. Schafer, *Discrete Time Signal Processing*, 2nd Edition, Prentice Hall, New Jersey, 1999.

[4] Ben Gold and Nelson Morgan, *Speech and Audio Signal Processing*, John Wiley and Sons, New York, NY, 2000.

[5] Gordon E. Carlson, *Signal and Linear System Analysis*, John Wiley and Sons, New York, NY, 1998

[6] Wai C. Chu, *Speech Coding Algorithm*, Wiley Interscience, New York, NY, 2003

[7] Thomas F. Quatieri, *Discrete Time Speech Signal Processing*, Prentice Hall, New Jersey, 2002.

[8] John R Deller, John G Proakis and John H.L. Hansen, *Discrete Time Processing of Speech Signals*, Macmillan, New York, NY, 1993.

[9] Charles K. Alexander and Matthew N.O. *Sadiku*, *Fundamentals of Electric Circuit*, McGraw Hill, New York, NY, 2000.

[10] *www.labrosa.ee.columbia.edu*

[11] J. P. Campbell, "Speaker Recognition : A Tutorial," *Proc. IEEE*, Vol 85, No. 9, pp. 1437–1462, Sept. 1997.

[12] B. S. Atal, "Automatic Recognition of Speakers from Their Voices," *IEEE Proc.*, Vol. 64, No. 4, pp. 460–475, 1976.

[13] S. Furui, "Cepstral Analysis Technique for Automatic Speaker Verification," *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. 29, No. 2, pp. 254–272, April 1981.

[14] F. Soong, A. Rosenberg, L. Rabiner and B. Juang, "A Vector Quantization Approach to Speaker Recognition," *Proc. Int. Conf. Acoustic, Speech, and Signal Processing*, Vol. 1, pp. 387-390, Tampa, FL, 1985.