International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 5

A Robust Speaker Identification System

Zaw Win Aung
Technological University, Loikaw, Myanmar

ABSTRACT

This paper is aimed to implement a robust speaker identification system. It is a software architecture which identifies the current talker out of a set of speakers. The system is emphasized on text-dependent speaker identification system. It contains three main modules: endpoint detection, feature extraction and feature matching. The additional module, endpoint detection, removes unwanted signal and background noise from the input speech signal before subsequent processing. In the proposed system, Short-Term Energy analysis is used for endpoint detection. Mel-frequency Cepstrum Coefficients (MFCC) is applied for feature extraction to extract a small amount of data from the voice signal that can later be used to represent each speaker. For feature matching, Vector Quantization (VQ) approach using Linde, Buzo and Gray (LBG) clustering algorithm is proposed because it can reduce the amount of data and complexity. The experimental study shows that the proposed system is more robust than using the original system and it is faster in computation than the existing one. To implement this system MATLAB is used for programming.

KEYWORD: *speaker recognition; speaker identification; endpoint detection; mel-frequency cepstrum coefficients; vector quantization*

I. INTRODUCTION

Nowadays more and more attention has been paid on speaker recognition field. Speaker recognition, which involves two applications: speaker identification and speaker verification, is the process of automatically recognizing who is speaking on the basis of individual information included in speech waves. This technique makes it possible to use the speaker's voice to verify their identity and control access to services such as voice dialing, banking by telephone, telephone shopping, database access services, information services, voice mail, security control for confidential information areas, and remote access to computers [1].

The first attempts for automatic speaker recognition were made in the 1960s. Pruzansky at Bell Labs [2] was among the first to initiate research by using filter banks and correlating two digital spectrograms for a similarity measure. Pruzansky and Mathews [3] improved upon this technique; and, Li et al. [4] further developed it by using linear discriminators. Doddington at Texas Instruments (TI) [5] replaced filter banks by formant analysis. Intra-speaker variability of features, one of the most serious problems in speaker recognition, was intensively investigated by Endres et al. [6] and Furui [7].

Research on increasing robustness became a central theme in the 1990s. Matsui et al. [8] compared the VQ-based method with the discrete/continuous ergodic HMM-based method, particularly from the viewpoint of robustness against utterance variations. They found that the continuous ergodic HMM method is far superior to the discrete ergodic HMM method and that the continuous ergodic HMM method is as robust as the VQ based method when enough training data is available. They investigated speaker identification rates using the continuous HMM as a function of the number of states and mixtures. It was shown that speaker recognition rates were strongly correlated with the total number of mixtures, irrespective of the number of states. This means that using information about transitions between different states is ineffective for text-independent speaker recognition and, therefore, GMM achieves almost the same performance as the multiple-state ergodic HMM.

Matsui et al. proposed a text-prompted speaker recognition method, in which key sentences are completely changed every time the system is used [9]. The system accepts the input utterance only when it determines that the registered speaker uttered the prompted sentence. Because the vocabulary is unlimited, prospective impostors cannot know in advance the sentence they will be prompted to say. This method not only accurately recognizes speakers, but can also reject an utterance whose text differs from the prompted text, even if it is uttered by a registered speaker. Thus, a recorded and played back voice can be correctly rejected.

II. OVERVIEW OF SPEAKER IDENTIFICATION SYSTEM

Speaker identification is the process of identifying a person on the basis of speech alone. Campbell defines it more precisely as the use of a machine to recognize a person from a spoken phrase [10].

All speaker identification systems contain two main modules: feature extraction and feature matching. Feature extraction is the process that extracts a small amount of data from the voice signal that can later be used to represent each speaker. Feature matching involves the actual procedure to identify the unknown speaker by comparing extracted features from his or her voice input with the ones from a set of known speakers.

For almost all the recognition systems, there are two main phases. The first phase is called enrollment phase and the next phase is called identification or verification (testing) phase. Enrollment phase is to get the speaker models or voiceprints for speaker database. In this phase, the most useful features are extracted from speech signal for speaker identification or verification, and train models to get optimal system parameters.

In identification phase, the same method for extracting features as in the first phase is used for the incoming speech signal, and then the speaker models getting from enrollment phase are used to calculate the similarity between the new speech signal model and all the speaker models in the database. After all comparisons are made, the new speaker will be assigned to the speaker ID which has the maximum similarity in the database. In the case of N-speaker system, N comparisons must be made for each unknown sample of speech [11].

III. SYSTEM ARCHITECTURE

Most speaker identification systems have two main modules: feature extraction and feature matching. But there are three modules in the proposed system. They are endpoint detection, feature extraction and feature matching. The additional module, endpoint detection, is used to remove unwanted signal and background noise from the input speech signal, to improve the recognition accuracy and to reduce the computing complexity. Fig. 1 and Fig. 2 show the training and testing phases of the proposed speaker identification system.

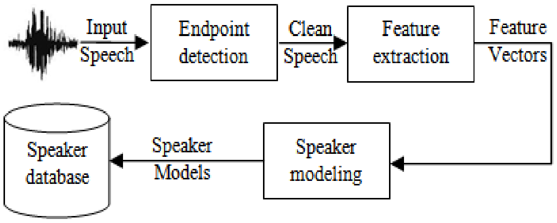


Fig.1. Training phase of speaker identification system

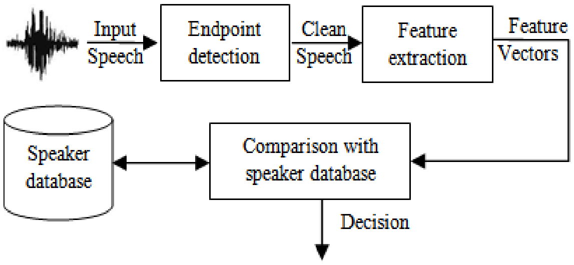


Fig.2. Testing phase of speaker identification system

A. Endpoint Detection

The process of separating the speech segments of an utterance from the background, i.e., the non speech segments obtained during the recording process, is called endpoint detection [12]. Accurate speech endpoint detection is crucial for the recognition performance in improving the recognition accuracy and reducing the computing complexity. In noisy environment, speech samples containing unwanted signals and background noise are removed by endpoint detection method. Over the years, different approaches have been proposed for the detection of speech segments in the input signal data. The early algorithms were based on extracting features such as short-term energy, zero crossing rate, linear prediction and pitch analysis. In the recent years, classification of voiced and unvoiced segments was based on cepstral coefficients, wavelet transform, periodicity measure and statistical models. The short-term energy will be used in the proposed system.

Speech is produced from a time varying vocal tract system with time varying excitation. Due to this, the speech signal is non-stationary in nature. Speech signal is stationary when it is viewed in blocks of 10-30msec [13]. Short Term Processing divides the input speech signal into short analysis segments that are isolated and processed with fixed (non-time varying) properties. These short analysis segments called as analysis frames almost always overlap one another. The energy associated with voiced speech is large when compared to unvoiced speech [14]. Silence speech will have least or negligible energy when compared to unvoiced speech [13]. Hence, Short Term Energy can be used for voiced, unvoiced and silence classification of speech. For Short Term Energy computation, speech is considered in terms of short analysis frames whose size typically ranges from 10-30 msec. Different energies used for signal analysis are as per equation (1), (2) and (3). Where, equation (1) represents Logarithmic Short-Term Energy, equation (2) represents the squared short-Term Energy and equation (3) represents Absolute Short-Term Energy[15].

$$E_{log} = \sum_{n=1}^{N} \log\,[s(n)^2] \qquad (1)$$

$$E_{sqr} = \sum_{n=1}^{N} [s(n)^2] \qquad (2)$$

$$E_{abs} = \sum_{n=1}^{N} |s(n)^2| \qquad (3)$$

Where, s(n) is the speech signal and N is length of sampled signal. The Logarithmic Short-Term Energy is most suitable, hence used in the proposed system.

B. Feature Extraction

The purpose of this module is to convert the speech waveform to some type of parametric representation (at a considerably lower information rate) for further analysis and processing. This is often referred as the signal-processing front end. The speech signal is a slowly timed varying signal (it is called quasi-stationary). When examined over a sufficiently short period of time (between 5 and 100 msec), its characteristics are fairly stationary. However, over long periods of time (on the order of 1/5 seconds or more) the signal characteristic change to reflect the different speech sounds being spoken. Therefore, short-time spectral analysis is the most common way to characterize the speech signal [16].

A wide range of possibilities exist for parametrically representing the speech signal for the speaker identification task, such as Linear Predictive Coding (LPC), Mel-frequency Cepstrum Coefficients (MFCC), and others. MFCC is perhaps the best known and most popular, and these will be used in this system.

MFCCs are based on the known variation of the human ear's critical bandwidths with frequency; filters spaced linearly at low frequencies and logarithmically at high frequencies have been used to capture the phonetically important characteristics of speech. This is expressed in the mel-frequency scale, which is linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. The block diagram of MFCC processor is shown in Fig. 3.

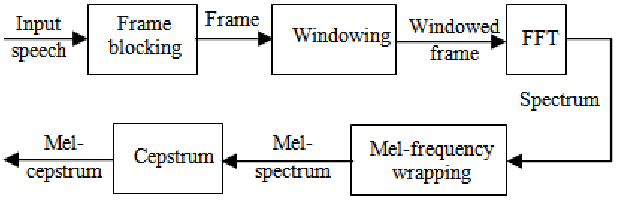


Fig.3. Block diagram of MFCC processor

Firstly, the input speech signal is blocked into frames of N samples overlapping by N-M samples. The values for N and M are 256 and 100. Then, the blocked frames are windowed with hamming window which has the form:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), 0 \le n \le N-1. \qquad (4)$$

The result of windowing is the signal,

$$y1\ (n) = x1\ (n)\ w\ (n),\ 0 \le n \le N\ \text{-}1. \qquad (5)$$

And each windowed frame of N samples is converted from the time domain into frequency domain by FFT which is defined as follow:

$$X_n = \sum_{k=0}^{N-1} x_k e^{-2\pi jkn/N}, n = 0,1,2,\ldots, N-1 \qquad (6)$$

And then the mel-frequency is computed for a given frequency f in Hz by the following formula:

$$mel(f) = 2595 \times \log 10(1 + f/700). \qquad (7)$$

The number of mel spectrum coefficients, K, is typically chosen as 20. Finally, the mel power spectrum coefficients are converted back to time domain and mel-frequency cepstrum coefficients are calculated as follow:

$$\tilde{C}_n = \sum(\log \tilde{S}_k)\cos\left[n\left(k-\frac{1}{2}\right)\frac{\pi}{k}\right], \; n=1,2,\ldots,K. \qquad (8)$$

Where, $\tilde{S}_k$, k = 1, 2, …, K is mel power spectrum coefficients.

By applying the procedure described above, for each speech frame of around 30 msec with overlap, a set of mel-frequency cepstrum coefficients is computed. These are result of a cosine transform of the logarithm of the short-term power spectrum expressed on a mel-frequency scale. This set of coefficients is called an acoustic vector. Therefore each input utterance is transformed into a sequence of acoustic vectors.

C. Feature Matching

The state-of-the-art in feature matching techniques used in speaker identification includes Dynamic Time Warping (DTW), Hidden Markov Model (HMM), and Vector Quantization (VQ). In this paper, the VQ approach will be used, due to ease of implementation and high accuracy. VQ is a process of mapping vectors from a large vector space to a finite number of regions in that space. Each region is called a cluster and can be represented by its center called a codeword. The collection of all code words is called a codebook. The distance from a vector to the closest codeword of a codebook is called a VQ-distortion. In the recognition phase, an input utterance of an unknown voice is "vector-quantized" using each trained codebook and the total VQ distortion is computed. The speaker corresponding to the VQ codebook with smallest total distortion is identified. Fig. 4 shows block diagram of the basic VQ training and classification structure.

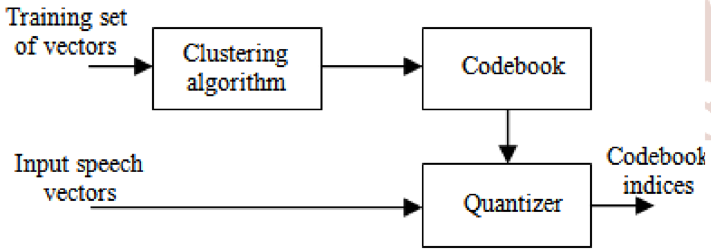


Fig.4. Block diagram of the basic VQ training and classification structure

Initially, the training set of vectors is used to create the optimal set of codebook vectors for representing the spectral variability observed in the training set. Then, similarity or distance is measured between a pair of spectral analysis vectors so as to be able to cluster the training set vectors as well as to associate or classify arbitrary spectral vectors into unique codebook entries. The next step is a centroid computation procedure. Finally, a classification procedure chooses the codebook vectors that closest to the input vector and uses the codebook index as the resulting spectral representation. This is often referred to as the nearest-neighbor labeling or optimal encoding procedure. The classification procedure is essentially a quantizer that accepts, as input, a speech spectral vector and provides, as output, the codebook index of the codebook vectors that best matches the input [17].

After the enrollment session, the acoustic vectors extracted from input speech of a speaker provide a set of training vectors. Then, the next important step is to build a specific VQ codebook for this speech signal using those training vectors. There is a well-know algorithm, namely LBG algorithm for clustering a set of L training vectors into a set of M codebook vectors. The algorithm is formally implemented by the following recursive procedure:

1. Design a 1-vector codebook: this is the centroid of the entire set of training vectors (hence, no iteration is required here).
2. Double the size of the codebook by splitting each current codebook y_n according to the rule:

$$y_n^+ = y_n(1+\varepsilon)$$
$$y_n^- = y_n(1-\varepsilon) \qquad (9)$$

Where n varies from 1 to the current size of the codebook, and ε is a splitting parameter (choose ε = 0.01).

3. Nearest-Neighbor Search: for each training vector, find the codeword in the current codebook that is closest (in terms of similarity measurement), and assign that vector to the corresponding cell (associated with the closest codeword).
4. Centroid Update: update the codeword in each cell using the centroid of the training vectors assigned to that cell.
5. Iteration 1: repeat steps 3 and 4 until the average distance falls below a preset threshold.
6. Iteration 2: repeat steps 2, 3 and 4 until a codebook size of M is designed.

Intuitively, the LBG algorithm designs an M-vector codebook in stages. It starts first by designing a 1-vector codebook, then uses a splitting technique on the codeword to initialize the search for a 2-vector codebook, and continues the splitting process until the desired M-vector codebook is obtained. Fig. 5 shows block diagram of the LBG algorithm.

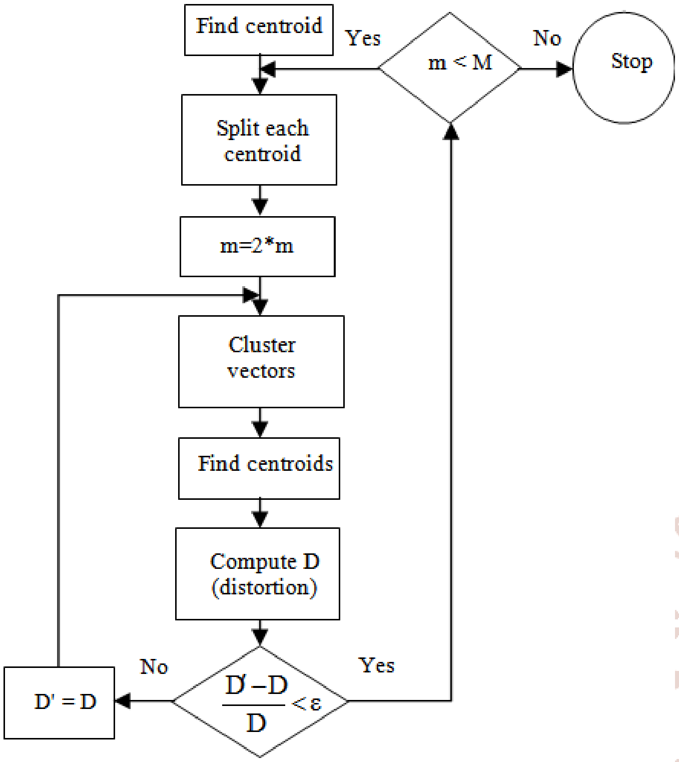
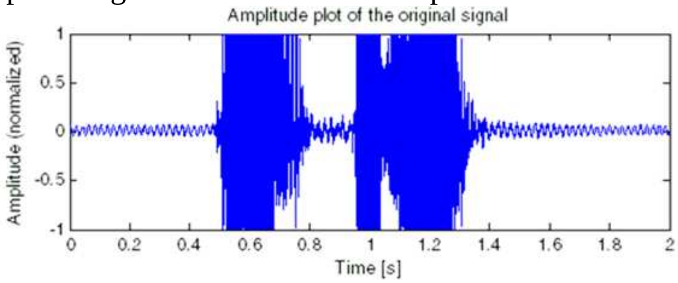


Fig.5. Block diagram of LBG algorithm

"Cluster vectors" is the nearest-neighbor search procedure which assigns each training vector to a cluster associated with the closest codeword. "Find centroids" is the centroid update procedure. "Compute D (distortion)" sums the distances of all training vectors in the nearest-neighbor search so as to determine whether the procedure has converged.

IV. IMPLEMENTATION OF THE SYSTEM

The proposed speaker identification system is simulated in MATLAB with speech signals as input and produces the identity of speaker as output. The speaker utters his/her name once in a training session and again in a testing session later on. The sounds to be trained and tested were recorded as wave format. In the proposed system, there are three main steps. In the first step, the input speech signal is processed for endpoint detection using the Logarithmic Short-Term Energy to remove unwanted signal and background noise. Fig. 6 shows the amplitude plot of the input speech signal before and after endpoint detection.



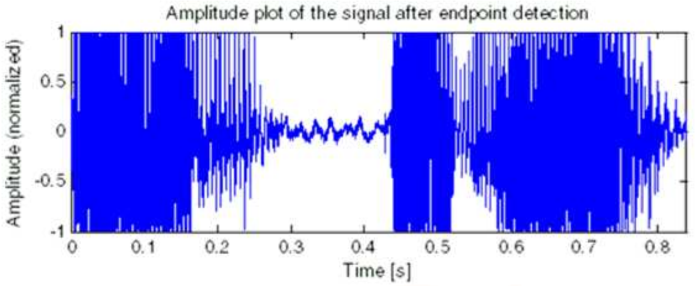


Fig.6. Amplitude plot of the speech signal before and after endpoint detection

In the second step, the speech signal is processed for feature extraction using MFCC feature extraction algorithm. Firstly, the speech signal is blocked into frames of N samples. Then, each frame of N samples is converted from the time domain into frequency domain. Fig. 7 shows the linear power spectrum plot and logarithmic power spectrum plot in frequency domain.

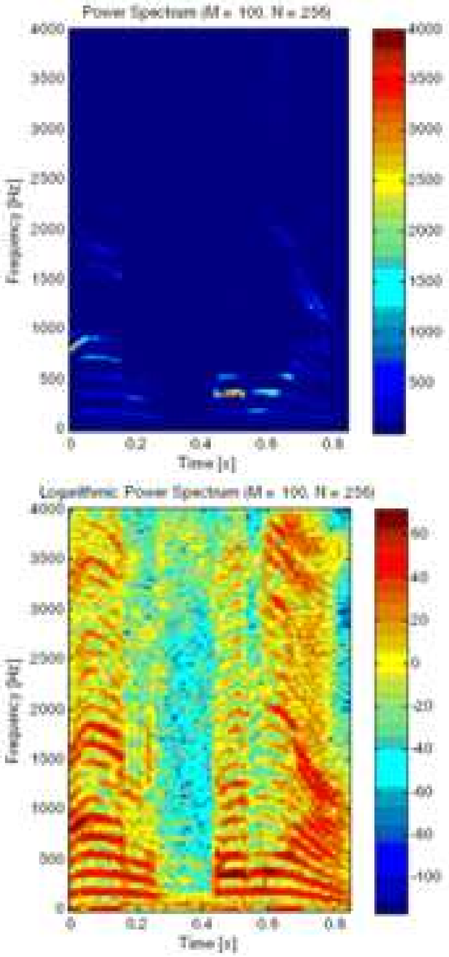


Fig.7. Power spectrum and logarithmic power spectrum of the speech signal

And then the power spectrum is converted into filter bank outputs. Fig. 8 shows the mel-spaced filter bank output.

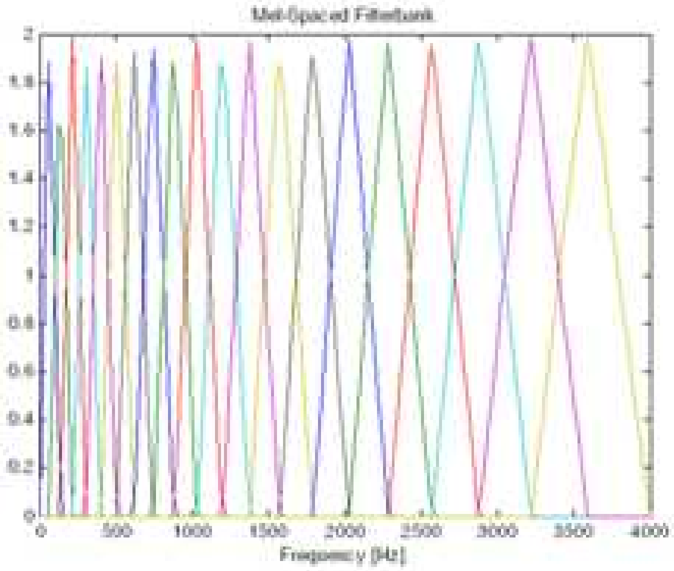


Fig.8. Mel-spaced filter bank output

Fig. 9 shows unmodified power spectrum and modified power spectrum after passing through mel-frequency filter bank.

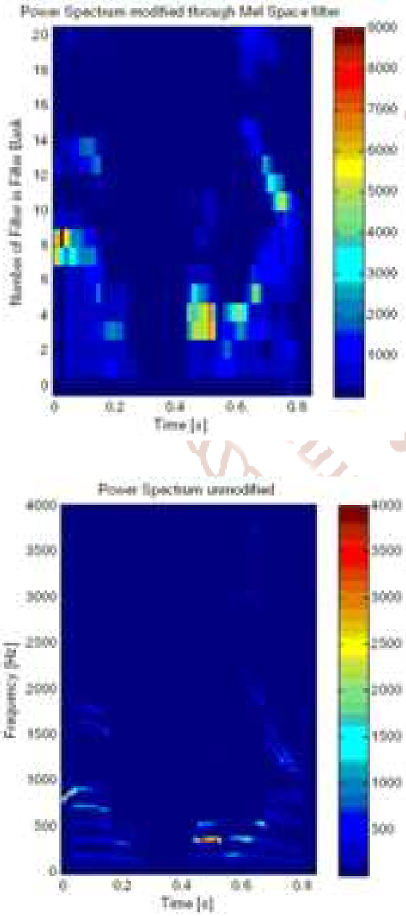


Fig.9. Unmodified power spectrum and modified power spectrum by MFCC filter

After passing through the filter bank, the mel-spectrum is converted into time domain and mel-frequency cepstrum coefficients are obtained.

In the third step, the speech signal is processed for feature matching and decision making using Vector Quantization (VQ) approach. The input for this step is acoustic vector of feature extraction stage. Fig. 10 shows 2D plot of acoustic vector for input speech signal.

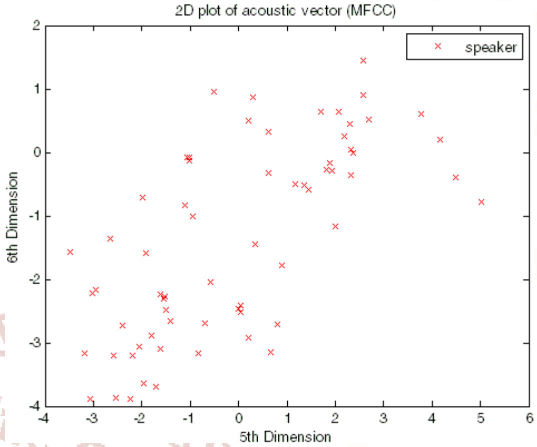


Fig.10. 2D plot of acoustic vector

In the training phase, the codebook or reference model for each speech signal is constructed from the MFCC feature vectors using LBG clustering algorithm and store it in the database. Fig. 11 shows the plot for 2D trained VQ codebook.

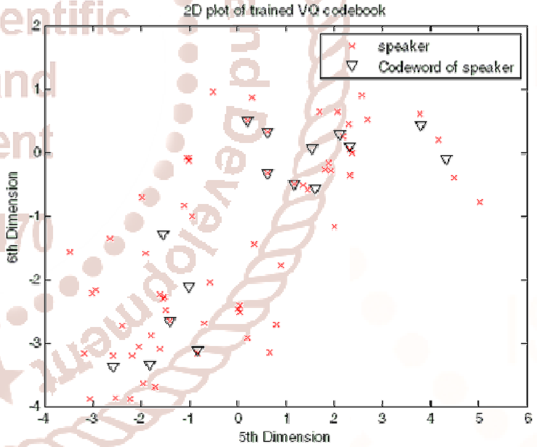


Fig.11. 2D plot of trained VQ codebook

In the identification phase, the input speech signal is compared with the stored reference models in the database and the distance between them is calculated using Euclidean distance. And then, the system outputs the speech ID which has minimum distance as identification result.

V. EXPERIMENTAL RESULT

The Purpose Of This Section Is To Illustrate The Performance Of The Proposed System Comparing With The Original System. The Speech Samples Were Collected From 20 Adults, Ten Male Speakers And Ten Female Speakers. Speakers Were Asked To Utter

Their Name In Normal Speed, Under Normal Laboratory Conditions. The Same Microphone Is Used For All Recordings. Speech Signals Are Sampled At 8000 Hz. The Speaker Utters His/Her Name Once In A Training Session And Again In A Testing Session Later On. Training Samples Are Recorded By Uttering The Name Of The Speaker (E.G. "I Am Zaw Win Aung"), Which Is About 2 Seconds Long For Each Sample. Testing Samples Are Also Recorded In The Same Way.

The Experiments Were Carried Out Using Different Database Sizes (20 Samples, 40 Samples And 60 Samples In The Database). In Order To Show The Effectiveness Of The Proposed System, The Computation Time And Accuracy Of The Proposed System And The Original System Were Computed And Compared.

In the training phase, three experiments were carried out. In the first experiment, 1 speech sample was collected from each speaker for training. So, there are 20 speech samples in the database. In this experiment, total length of training time with the proposed system is 1.23 seconds but the original system takes 1.41 seconds for training 20 speech samples. In the second experiment, 2 speech samples were collected from each speaker for training. So, the size of the database was increased to 40. While the proposed system takes 2.59 seconds, the original system takes 2.78 seconds for training 40 speech samples. In the third experiment, 3 speech samples were collected from each speaker for training. So, the size of the database became 60. In this experiment, it is found that the computation times for the proposed system and the original system are 3.59 seconds and 4.06 seconds respectively. The training times taken by the two systems in different database sizes are shown in Table 1.

Table1. Computation time taken by the two systems in training phase

No	No: of Trained Samples	Time taken by the proposed system (seconds)	Time taken by the original system (seconds)
1	20	1.23	1.41
2	40	2.59	2.78
3	60	3.59	4.06

In the testing phase, twenty speech samples which were collected from 10 male speakers and 10 female speakers were used as test speech samples. Firstly the experiment was carried out using 20 speech samples in the database. While the computation time taken by the proposed system is 0.72 seconds, the original system takes 0.8 seconds for testing 20 speech samples. The accuracy is 75 percent and 65 percent respectively for the proposed system and the original system. The system is also tested with 40 speech samples in the database. The proposed system takes 0.95 seconds while original system takes 1.09 seconds for testing 20 speech samples. In the case of accuracy, the proposed system achieves 90 percent accuracy while the original system has only 80 percent accuracy. In the experiment of testing with 60 speech samples in the database, it is found that the computation times for the proposed system and the original system are 1.20 seconds and 1.48 seconds respectively for testing 20 speech samples. When the accuracy is taken into account, the proposed system and the original system achieves the accuracy of 95 percent and 90 percent respectively. The testing times taken by the two systems and the accuracy of the two systems in different database sizes are shown in Table 2 and Table 3.

Table2. Computation time taken by the two systems in testing phase

No	No: of Test Samples	No: of Samples in the Database	Time taken by The proposed system (seconds)	Time taken by the original system (seconds)
1	20	20	0.72	0.8
2	20	40	0.95	1.09
3	20	60	1.20	1.48

Table3. Accuracy of the two systems

No	No: of Test Samples	No: of Samples in the Database	Accuracy of the proposed system(percent)	Accuracy of the original system(percent)
1	20	20	75%	65%
2	20	40	90%	80%
3	20	60	95%	90%

According to the experiments, it can be seen that the training time and testing time of the proposed system is less than that of the original system. In the case of accuracy, the proposed system achieves higher accuracy than the existing one.

Vi. CONCLUSION

From this work it can be concluded that the proposed system is more robust and faster than the original system. So, the proposed system can be used in real-world speaker identification applications where a moderate number of speakers are available such as teleconferences and speaker tracking. And the proposed system is reasonably fast for working in real-time. A range of future improvements is also possible such as text independent speaker identification system and identification of a male, female, child and adult.

ACKNOWLEDGMENTS

The author would like to thank all of his colleagues who have given him support and encouragement during the period of the research work. The author would like to extend his thanks to his students who provided the speech samples for this research work. Finally the author would like to express his indebtedness and deep gratitude to his beloved parents, wife and son for their kindness, support and understanding during the research work.

REFERENCES

1. R. A. Cole and colleagues, Survey of the State of the Art in Human Language Technology, National Science Foundation European Commission, 1996. http://cslu.cse.ogi.edu/HLTsurvey/ch1node47.html
2. S. Pruzansky, “Pattern-matching procedure for automatic talker recognition,” J.A.S.A., 35, pp. 354-358, 1963.
3. S. Pruzansky and M. V. Mathews, ”Talker recognition procedure based on analysis of variance,” J. A. S. A., 36, pp. 2041-2047, 1964.
4. K. P. Li, et. al., “Experimental studies in speaker verification using a adaptive system,” J.A. S. A., 40, pp. 966-978, 1966.
5. G. R. Doddington, “A method of speaker verification,” J. A. S. A., 49, 139 (A), 1971.
6. W. Endress, et. al., “Voice spectrograms as a function of age,” Voice Disguise and Voice Imitation, J. A. S. A., 49, 6(2), pp. 1842-1848, 1971.
7. S. Furui, “An analysis of long-term variation of feature parameters of speech and its application to talker recognition,” Electronics and Communications in Japan, 57-A, pp. 34-41, 1974.
8. T. Matsui and S. Furui, “Comparison of text independent speaker recognition methods using VQ-distortion and discrete/continuous HMMs,” Proc. ICSLP, pp. II-157-160, 1992.
9. T. Matsui and S. Furui, “Concatenated phoneme models for text-variable speaker recognition,” Proc. ICASSP, pp. II-391-394, 1993.
10. R. Ramachandran, K. Farrell, R. Ramachandran, and R. Mammone, Speaker Recognition - General Classifier Approaches and Data Fusion Methods, Pattern Recognition 35, 2002, pp. 2801–2821.
11. C. A. Norton, Text Independent Speaker Verification using Binary-pair Partitioned Neural Networks, Master of Science, Electrical Engineering, Old Dominion University, December, 1995.
12. Miael Nilsson and Marcus EJnarsson, “Speech Recognition using Hidden Markov Model”, Department of Telecommunications and speech Processing, Biekinge Institute of Technology, 2002.
13. Ronald W.Schafer and Lawrence R. Rabiner, “Digital Representations of Speech Signals”, Proceedings of the IEEE, Vol.63, No.4, April 1975.
14. R. G. Bachu, S.Kopparthi, B.Adapa and B. D. Barkana, “Voiced/Unvoiced Decision for Speech Signals Based on Zero-Crossing Rate and Energy”, Advanced Techniques in Computing Sciences and Software Engineering, pp.279-282, 2010.
15. Nitin N Lokhande, Navnath S Nehe and Pratap S Vikhe, " Voice Activity Detection Algorithm for Speech Recognition Applications ", Proceeding of the International Journal of Computer Applications® (IJCA), 2011.
16. M. N. Do, An Automatic Speaker Recognition System, Digital Signal Processing Mini-Project, Audio Visual Communications Laboratory, Swiss Federal Institute of Technology, Lausanne, Switzerland, 2003-2004. http://lcavwww.epfl.ch/~minhdo/asr_project.pdf
17. L. Rabiner, B. H. Juang, Fundamentals of Speech Recognition, Prentice Hall, 1993.