Voice Biometric System for Speaker Authentication

S. Karamchandani C. Thomas V.
Sachdev P. Raut
Assistant Professor P.G. Student U.G. Student
U.G.Student
DJSCE DJSCE DJSCE DJSCE

D. Srivastava N. Shah
P.G. StudentU.G. Students
DJSCE DJSCE

ABSTRACT
This paper discusses the implementation of a text-independent, biometric, Speaker Recognition based security system. Such a bio-metric model can be used to identify a person based on his or her voice-print. Mel Frequency Cepstral Coefficients (MFCCs) are used to denote the unique features of a person's speech and Vector Quantization has been used to generate speaker specific codebooks.The traditional techniques of speech processing like end-point detection, pre-emphasis, windowing, Fourier transformation have been used. The advantages and limitations of the proposed system are also stated in this paper.

Keywords
Speaker Identification, Signal Processing, Feature Extraction, MFCCs, DCT(discrete cosine transform), Vector Quantization.

1. INTRODUCTION
From time immemorial, manipulation of identity has been a common method adopted by criminals to carry out frauds and gain access to credit card details, bank account information and tax information of civilians. With criminals becoming more tech-savvy by the day, the need for a robust security system based completely on human features unique to an individual has become crucial.Each person has a different voice-print (distinctive pattern of voice characteristics) and a security system based on Voice Authentication exploits this uniqueness.To realize such a security system, speech processing of the voice sample of an individual needs to be carried out. Speech processing involves the analysis and manipulation of certain features of the speech signal both in time and frequency domain and the end results can be used in various applications, in our case, extraction of features.There are different techniques for feature extraction, the most common being LPC i.e. linear predictive coding and MFCC i.e. Mel frequency cepstral coefficients. LPC is a time domain method and the amplitude of the speech signal suffers variations due to noise. The preferred technique for feature extraction is MFCC wherein the features are generated by transforming the signal into frequency domain. In general, cepstral features are more compact,discriminable, and most importantly, nearly de-correlated and therefore, they can provide higher baseline performance over filter bank features.The main advantage of a voice biometric system is the difficulty in forging sound, pattern and rhythm of an individual's voice. Also, an IRIS system is susceptible to poor image quality whereas a security system dependent on the voice-print of a person is much more robust and does not fail even in the event of the individual suffering from a sore throat or cold as the underlying features of the voice do not get altered due to such diseases.

2. BLOCK DIAGRAM FOR FEATURE EXTRACTION

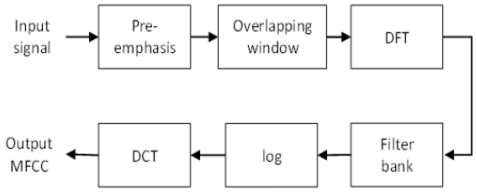

Fig 1: Feature Extraction using MFCC

3. FEATURE EXTRACTION

3.1 Endpoint Detection
Endpoint detection is a process of extracting voiced/ unvoiced content from a speech signal, i.e. discarding the silent regions. This is an important pre-processing step because it gives us the exact frames where our speech information is present and hence feature extraction is then done from these frames only. If all the frames are used instead of these selected frames, too much computation time and power will be required without any significant improvement of feature quality. Conventional methods of Endpoint detection are SHORT TIME ENERGY (STE) and ZERO CROSSING RATE (ZCR). In STE, the energy content of each frame is calculated and the mean of energy in all frames is found out. This mean acts as threshold. Now each frame's energy is compared with this threshold.If it is greater than threshold, the frame is considered for feature extraction else it is not considered. In ZCR,the number of zero crossings in each frame is calculated. Mean of these zero crossings is taken. This mean acts as threshold. Now each frame's zero crossing count is compared with the threshold. If it is greater than threshold, the frame is considered for feature extraction else it is not considered.

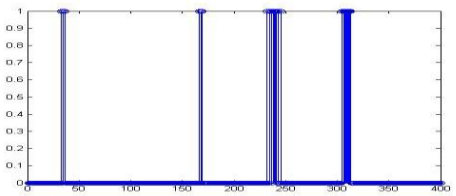

Fig 2:Endpoint Detection

These methods don't work well in noisy environments. Hence we opted for a better algorithm.In this algorithm,initial few frames when the speaker does not start speaking are used to get mean(μ) and standard deviation(σ) of the background noise. For all the frames, the Mahalanobis distance is found out using the Mahalanobis distance function (x- μ / σ).Then a threshold is set.If the Mahalanobis distance is greater than

threshold then the frame is made equal to 1 else 0.Now we divide the speech signal in frames of 30ms with 50% overlapping.Though the paper suggests non-overlapping frames,we chose to go with overlapping frames as we require overlapping frames for feature extraction.Now if a frame contains 1s more than a specified threshold, the frame is considered for feature extraction else it is not considered.

3.2 Pre-Emphasis

In processing of the speech signals, pre emphasis is done in order to increase the magnitude of higher frequencies with respect to the magnitude of the lower frequencies within the frequency band. Pre emphasis also helps to improve the signal to noise ratio by minimizing the adverse effects like attenuation or distortion.Pre emphasis can be specified by,

$$Y[n] = X[n] - (a \times X[n-1]) \quad (1)$$

Where a ranges from 0.9 to 0.99

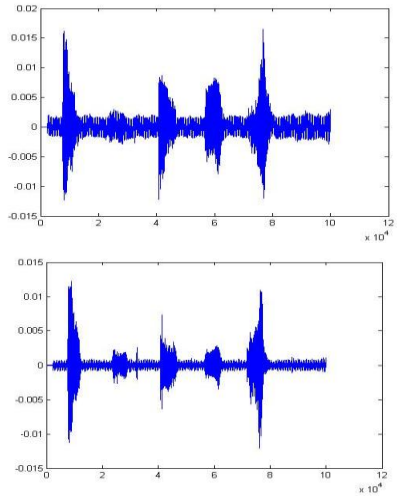

Fig 3:Original Speech and Pre-emphasized speech

3.3 Windowing

As speech is a non-stationary signal,it is difficult to work on the whole signal in the frequency domain. Thus the speech signal is broken down into frames of 25ms,in which it remains stationary. Overlapping frames have been used so that continuity in speech is retained. Each frame is then multiplied with a windowing function that overcomes the possible distortion in frequency domain due to frame blocking. We have used a hamming window and in MATLAB, a direct function, hamming() is available, which has been used to generate the window.

3.4 Discrete Fourier Transform

In order to calculate MFCC, the signal has to undergo Fourier Transform and a frequency spectrum has to be obtained. FFT (Fast Fourier Transform) is used to compute the Fourier Transform more quickly as compared to DFT. FFT of each frame is computed and stored in a two-dimensional array. The phase component of FFT is discarded and only the magnitude component of FFT was used as it contained maximum information and doing so saves memory. Finally, Power Spectrum is found out using magnitude of FFT which reduces noise in the frequency domain.

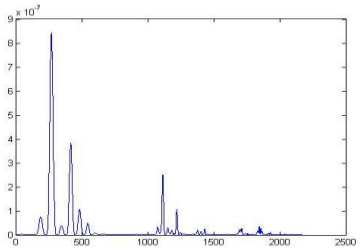

Fig 4: Power Spectrum

3.5 MFCC Filter Coefficients

The human cochlea cannot differentiate between two frequencies placed in close proximity to each other. As the number of frequencies increase, it becomes more difficult to precisely pin-point the individual frequencies. Thus, to solve this problem, the concept of Mel filter banks is introduced. The Mel filter bank tells us exactly how much energy is present near a particular frequency. It is basically a set of 20-40 triangular filters that are applied to the FFT values. We have used 25 vectors in our project where each vector has a non-zero value for a certain interval and a zero value for the rest of the frequency axis. The filter bank gives us information about how much energy of the concerned signal is present in different frequency regions. The first few filters are very narrow as we are more concerned about the variations at low frequencies. As we move from lower frequencies to the higher frequencies, the filters start getting wider. Now the energy in each band pass filter is calculated by taking sum.Logarithm of these energy coefficients is then taken.

3.6 Logarithm

In the speech production model, the 2 main components are the Vocal Tract model (VM) and the Glottal model (GM).The Glottal model shapes impulse train from the impulse train generator and vocal tract model replicates the vocal tract, essentially producing the formant frequencies. The speech production model can be represented in the time domain as:

$$S(t) = G(t) * V(t) \quad (2)$$

We know the important property of convolution which states that convolution in time domain is multiplication in frequency domain. Hence we represent the above model in frequency domain as:

$$S(f) = G(f) \times V(f) \quad \text{(eq.3)}$$

Now it is important to note here that, for Speaker Identification, separation of VM and GM is essential. This is because both models occupy different regions in the frequency domain, viz, The VM occupies Lower Frequency range and GM occupies Higher Frequency range. So separating these models will help us in de-correlating these models and hence better feature extraction can be carried out. But as we observe from equation 3, we have multiplication of VM and GM. Separating with multiplication is difficult hence we want something which is an easier operation, like addition. Taking Logarithm in this step can facilitate that.Hence we take log on both sides in equation 3.

$$\text{Log}(S(f)) = \text{Log}(G(f) \times V(f)) \quad (4)$$

This can be written as

$$\text{Log}(S(f)) = \text{Log}(G(f)) + \text{Log}(V(f)) \quad (5)$$

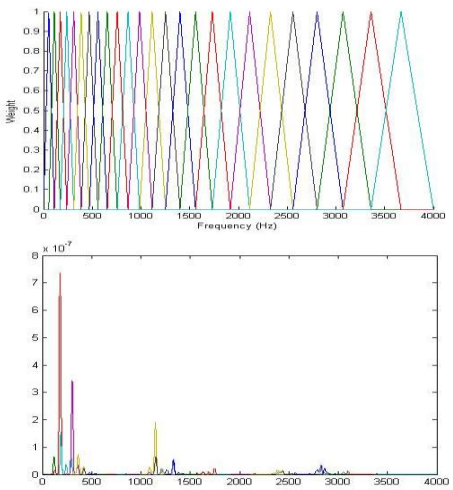

Fig 5: Mel filter bank and output after multiplication with power spectrum

3.7 Discrete Cosine Transform

In conventional cepstral analysis, Inverse Discrete Fourier transform is taken after the previous step to get the cepstral coefficients. But when Mel banks are used, DCT is used instead of IDFT. The main motive behind this is, due to overlapping band pass filters, there is a lot of correlation between adjacent filters.DCT de-correlates these filters. This step can also be thought of a compression step. As we want to come up with least number of features per speaker, compression is essential. We have used 25 filter banks in the Mel scale. So every frame gives us 25 coefficients. If we have 100 frames of voiced speech in the speech signal, we get 25x100 coefficients. The first coefficient is discarded because it contains the DC value and it contributes very little in speaker discrimination. So we finally get 24x100 coefficients for a speaker. These coefficients need to be passed through Vector quantization to get the final model for a Speaker.Thus for a frame if there are K number of coefficients and Sk denotes the coefficients, MFCC are obtained by this DCT equation,

$$C_n = \sum_{k=1}^{K} (logS_k) \cos[n(k - \frac{1}{2})\frac{\pi}{K}], \quad n = 0,1,\dots K-1 \qquad (6)$$

3.8 VECTOR QUANTIZATION

Vector Quantization is a clustering technique which is used to make code book for a speaker. Code book basically is a set of code vectors. These code vectors are obtained from the MFCC vectors. With VQ, we are trying to achieve lossy compression of data sets. This is essential because MFCC gives us large amount of data and there is a lot of variance in the data. Recognizing a speaker directly on the basis of MFCC vectors will lead to low efficiency. So we need to break this data down into a smaller set of vectors, collectively known as a code book .This is done by breaking a data set into N number of clusters.
For Speaker Recognition we have considered 24 mel band pass filters as a filter bank applied to each voiced speech frame. If we have 100 voiced speech frames, we end up having 24 x 100 MFCC vectors.

Now for code book generation, we consider one band pass filter at a time. Each band pass filter is referred to as a 'dimension'. For example 1th dimension has 100 MFCC vectors .We pass it through VQ training algorithm known as LBG algorithm which converts these 100 vectors into N code vectors. In LBG algorithm, value of N is some power of 2 and it is predefined. Let us assume N to be 8.So LBG algorithm will give us 8 code vectors per dimension. In all we get a code book of size 24 x 8.

3.9 The LBG Algorithm

VQ can be implemented using 2 algorithms, LBG algorithm and classical K-means clustering algorithm. Out of these the K-means algorithm suffers from some limitations like it requires a large amount of data for effective clustering and the number of clusters have to be pre-defined.LBG works better so we go with LBG. In LBG algorithm, we start with 1-vector code book and gradually increase the size of the code book in powers of 2 till we reach the desired size. The following algorithm is applied for one dimension at a time. So when we refer to complete data set, we refer to the complete data set in that dimension.The Algorithm is as follows:

N is the desired Codebook size and m is the current codebook size.

Initialize the size of codebook as m=1.The code vector will be the centroid of the complete data set.

1. Double the size of the code book by splitting each centroid into two
   $C_{2i-1} = C_i$ x (1-e)
   $C_{2i} = C_i$ x (1+e)
   Where e is a small number, we assume it to be 0.01
2. Create clusters equal to current code book size by finding out the closest centroid for each vector in the data set. The vector is assigned to that cluster whose centroid has the least Euclidean distance from the vector. This gives total m clusters.
3. Upgrade the Centroid of each cluster.
4. Calculate average distortion between the new centroids and the old ones. Repeat steps 2 and 3 until average distortion falls below pre-defined threshold.
5. If m=N, take C is the final codebook. Else go to step 2.

4. SIMULATION RESULTS

Complete system was developed on MATLAB and Experiments were carried out to test its efficiency. We considered 2 sets of data of 8 different speakers with each saying the word 'zero'. One was recorded for training phase and the other for testing phase. The system was trained with training data of these speakers. An important step we performed in training after generating the complete code book for a speaker was to sort code vectors in each dimension. This improved the efficiency of the system.After training, speaker identification was carried out. Code book was generated for a speaker using the same method mentioned above. Then Mean Square Error (MSE) was calculated between the testing code book and all other training code books individually. If code book A has vectors $x_{11}, x_{12}, x_{13} \dots x_{ij}$ and code book B has vectors $y_{11}, y_{12}, y_{13} \dots y_{ij}$ then MSE is calculated as:

$$MSE_{AB} = \sum_{i,j=1}^{i=24,j=N} (xij - yij)^2 \qquad (7)$$

As we have 8 speakers, we get 8 values of MSE. The one with the least value of MSE, is our actual speaker. Here we understand the importance of sorting because without sorting, each corresponding vector won't be close enough to the other and would result into reduced efficiency.The following table shows the results of Mean square error calculation for all 8 speakers. The rowsrepresent speech samples recorded during the training phase whereas the columns represent speech samples recorded during the testing phase. Each value in the table represents a MSE value between the corresponding speakers.

5. CONCLUSION

The proposed algorithm has been verified for eight speakers. The correlation between speakers two and one is minimum. The speaker two gives the best results according to the mse parameter verified in Table 1. The confusion matrix of the table suggest that speaker three gives the worst performance among the eight speakers. The efficiency can be increased using adaptive filters banks.

6. REFERENCES

[1] L.R Rabiner and R.W. Schafer, *Digital* Processing of Speech Signals, Prentice-Hall, Englewood Cliffs, N.J., 1978.

[2] L.R. Rabiner and B.H. Juang, Fundamentals of SpeechRecognition, Prentice-Hall, Englewood Cliffs, New Jersey, 1993.

[3] S. Furui, "An overview of speaker recognition technology", *ESCA Workshop on* Automatic Speaker Recognition, Identification and Verification, pp. 1-9, 1994.

[4] F.K. Song, A.E. Rosenberg and B.H. Juang, "A vector quantisation approach to speaker recognition", *AT&T Technical Journal*, Vol. 66-2, pp. 14-26, March 1987.

[5] A. Martin, D. Charlet, and L. Mauuary, "Robust speech / non- speech detection using

[6] LDA applied to MFCC", in Proc. ICASSP2001}, vol. 1, 2001, pp. 237–240.

[7] Ashish Kumar Panda, Amit Kumar Sahoo "Study of speaker recognition systems"

[8] Department Of Electronics And Communication National Institute Of Technology, Rourkela 2007.

[9] Prof. Ch.Srinivasa Kumar, Dr. P. Mallikarjuna Rao "Design of an automatic speaker recognition system using MFCC, vector quantization and LBG algorithm" Ch.Srinivasa Kumar et al./ International Journal on Computer Science and Engineering (IJCSE) vol. 3 No. 8 August 2011.

[10] Alan V. Oppenheim, Ronald W. Schafer "Discrete-Time Signal Processing, Prentice Hall, Englewood Cliffs, N.J., 1994.

[11] G. Saha, Sandipan Chakroborty, Suman Senapati,A New Silence Removal and Endpoint Detection Algorithm for Speech and Speaker Recognition Applications,Department of Electronics and Electrical Communication Engineering,Indian Institute of Technology, Khragpur, Kharagpur-721 302, India

[12] Practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/

7. APPENDIX

Table I: MSE values corresponding to the eight speakers

Speaker	1	2	3	4	5	6	7	8
1	**2.55**	0.94	5.72	0.51	5.82	1.24	0.98	0.69
2	8.78	**0.19**	8.52	1.35	5.38	2.22	0.68	1.06
3	8.06	1.02	**3.95**	0.65	5.50	0.98	0.73	0.60
4	4.08	0.89	4.40	**0.24**	5.25	1.48	0.77	0.32
5	4.08	0.49	4.34	0.78	**1.07**	1.34	0.38	0.57
6	7.01	1.81	7.24	0.61	9.73	**0.10**	1.48	0.73
7	3.97	0.55	4.60	0.69	2.01	1.17	**0.26**	0.45
8	3.42	0.83	4.74	0.48	4.71	1.05	0.53	**0.22**