International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:9, 2009

Voice Command Recognition System Based on MFCC and VQ Algorithms

Mahdi Shaneh, and Azizollah Taheri

***Abstract*—**The goal of this project is to design a system to recognition voice commands. Most of voice recognition systems contain two main modules as follow "feature extraction" and "feature matching". In this project, MFCC algorithm is used to simulate feature extraction module. Using this algorithm, the cepstral coefficients are calculated on mel frequency scale. VQ (vector quantization) method will be used for reduction of amount of data to decrease computation time. In the feature matching stage Euclidean distance is applied as similarity criterion. Because of high accuracy of used algorithms, the accuracy of this voice command system is high. Using these algorithms, by at least 5 times repetition for each command, in a single training session, and then twice in each testing session zero error rate in recognition of commands is achieved.

***Keywords*—**MFCC, Vector quantization, Vocal tract, Voice command.

I. Introduction

SPEECH processing is one of most important branches in digital signal processing. Speech signals can be used for speech recognition, speaker recognition or voice command recognition systems. For example in a motorized wheelchair, voice command recognition systems can be utilized instead of usual mechanical command systems. Proposed voice command recognition system includes two main stages. First stage contains feature extraction and storage of extracted features as training data. Second stage is test. In this stage, features of a new entered command are extracted. These features are used in order to make comparison with stored features to recognize command.

MFCC algorithm is used for feature extraction and vector quantization algorithm is used to reduce amount of achieved data in form of codebooks. These data are saved as acoustic vectors.

In the matching stage, features of input command are compared with each codebook using Euclidean distance criterion.

This paper is organized as follows. In section II proposed method is detailed, section III contains experimental result and section IV is conclusion.

Authors are with Islamic Azad University, Najafabad branch, Iran (e-mails: mahdishaneh@yahoo.com, taheri_az@yahoo.com).

II. Voice Command Recognition System

In this section, first speech production mechanism, voiced and unvoiced sounds and formants are described.

After familiarizing with these concepts, main parts of proposed recognition system, feature extraction and feature matching will be described.

A. Speech Production

The speech signal is an acoustic sound pressure wave that originates by exiting of air from vocal tract and voluntary movement of anatomical structure. Fig. 1 shows schematic diagram of the human speech production mechanism. The components of this system are the lungs, trachea larynx (organ of voice production), pharyngeal cavity, oral cavity and nasal cavity [1].

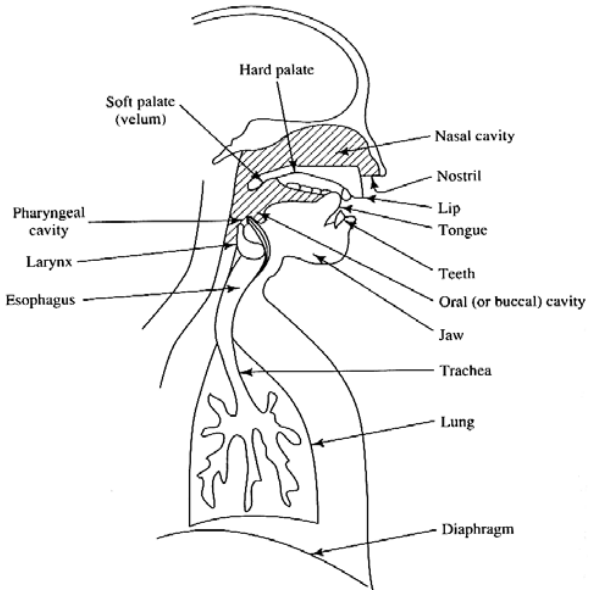

Fig. 1 Schematic diagram of the human speech production mechanism

In technical discussion, the pharyngeal and oral cavities are usually called the "vocal tract". Therefore the vocal tract begins at the output of the larynx and terminates at the input of lips. Finer anatomical components critical to speech production include the vocal cords, soft palate or velum, tongue, teeth, and lips. These components can move to different position to change the size and shape of vocal tract and produce various speech sound. For engineering purposes,

we can consider the speech production mechanism in term of an acoustic filtering operation. Thus, instead of anatomical model (Fig. 1), a technical model for speech production can be considered (Fig. 2). This filter is excited by the organs below it.

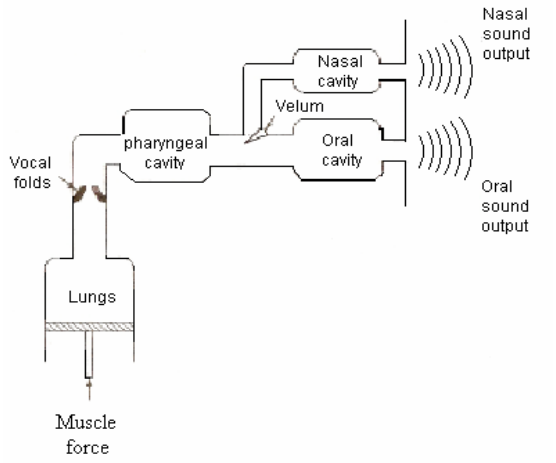

Fig. 2 Technical model for speech production

In speech processing, there are two fundamental excitation types "voiced" and "unvoiced". Voiced sounds are produced by forcing air through the glottis. Therefore the vocal cords vibrate. This vibration in vocal cords produces quasi-periodic airflow through vocal tract. By this means, larynx produces a periodic excitation to the system. The sound produced in this way is called "voice"[1]. Unvoiced sounds are produced when larynx is open and there is no vibration in vocal cords, so flowing air through the vocal tract is not periodic. Thus unvoiced sound has low amplitude and noisy form. Anyway, during the voiced sound production, we have a periodic signal and the vocal tract with varying shape as a function of time.

The vocal tract is a non-uniform acoustic tube. For a uniform tube, the resonance frequencies are obtained as follows:

$$F_i = \frac{C}{4L}(2i-1) \qquad \text{for i = 1,2,3,……} \qquad (1)$$

Where length of tube, L=17.5 cm (almost equal to an adult human vocal tract length) and C= speed of sound. Therefore we obtain different resonance frequency for this tube (in this case 500Hz, 1500Hz, 2500Hz…).

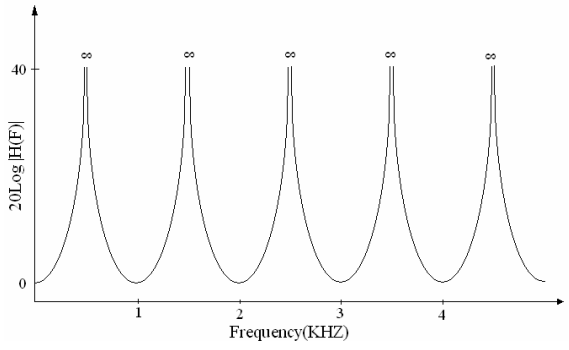

Fig. 3 Resonance frequencies for a uniform acoustic tube

However about vocal tract, during speaking there is varying in shape of this tube. So the resonance frequencies are changing. These resonance frequencies are called formants. We can characterize shape of vocal tract by these formants. For each voiced sound, there are infinite number of formants, but usually a few first of them are used. But for unvoiced sound, there is not any resonance frequency, because there is no periodic (or quasi-periodic) excitation in vocal tract. Fig. 4 shows formants of "i" and "o" as example of voiced sounds formants.

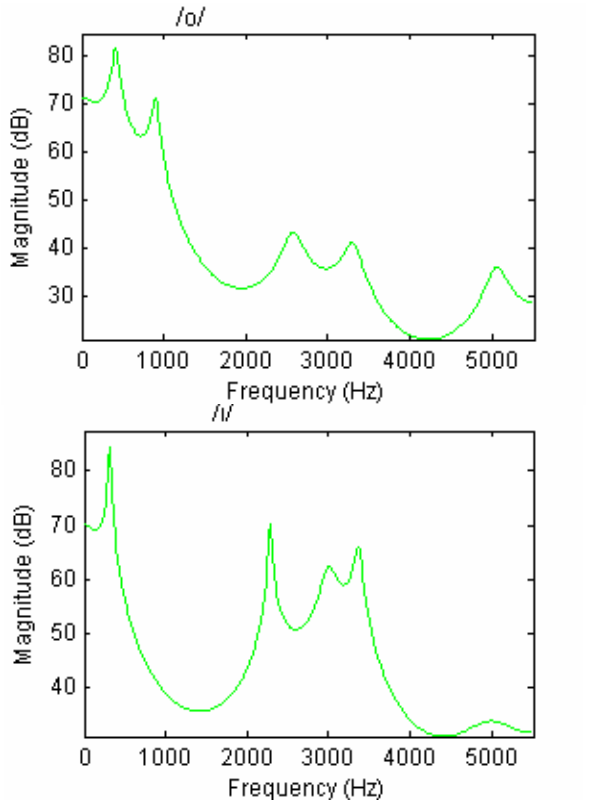

Fig. 4 Formants of "i" and "o" vowels

The formants are useful for speech dependent recognition systems. Using these formants, vocal tract and also utterance vowels can be characterized. Because in our system the commands have different vowels, an input command can be recognized via comparison of its characteristics with stored characteristics in database.

B. Feature Extraction

Before identifying or training a command that should be identified by the system, the voice signal must be processed to extract important characteristics of speech. Pitch frequency and formants are most important features of voice signal. Pitch is fundamental frequency of speech signal. The pitch frequency corresponds to the fundamental frequency of vocal cord vibrations. Pitch is a characteristic of excitation source.

Formants are resonance frequencies of vocal tract and so they are characteristics of vocal tract. Fig. 5 shows general linear discrete-time model for speech production [1].

According to this model, speech signal, S(n), is composed of a convolved combination of excitation signal, with the vocal tract impulse response.

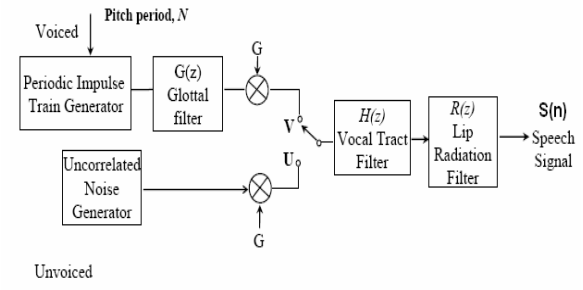

Fig. 5 A general discrete-time model for speech production

We have access only to the output signal, S(n), but we need separated e(n) and θ(n) for recognizing the command. Because individual parts are not combined linearity, the cepstral analysis is used to separate e(n) and θ(n).

In order to feature extraction, calculation of cepstral coefficients in mel frequency scale is required.

C. Cepstral Analysis

Cepstral is a time domain analysis that its main idea is separation of two convolved signals [1].

The output signal of speech production system S(n), is as follows:

$$s(n) = e(n) * \theta(n) \quad (2)$$

Using Fourier transform we have:

$$s(w) = E(w)\theta(w) \quad (3)$$

With taking logarithm, following equation is obtained:

$$\log|s(w)| = \log|E(w)| + \log|\theta(w)| \quad (4)$$

This equation is shown as follows:

$$cs(w) = ce(w) + c\theta(w) \quad (5)$$

Using IDFT, the cepstral coefficients are obtained.

$$cs(n) = ce(n) + c\theta(n) \quad (6)$$

In other word, cepstral coefficients are computed in the form of:

$$cs(n) = f^{-1}(\log[f(s(n))] \quad (7)$$

D. Mel-frequency Scaling

Physiological studies have shown that human auditory system does not follow a linear scale. Thus for each tone with an actual frequency, f, measured in Hz, a subjective pitch is mapped on a scale called the mel scale. The mel-frequency scale is a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. The main advantage of using mel frequency scaling is that mel frequency scaling is very approximate to the frequency response of human auditory systems and can be used to capture the phonetically important characteristics of speech.

One approach for simulating the subjective spectrum is to use a filter bank, spaced uniformly on the mel scale (see Fig. 6). That filter bank has a triangular band pass frequency response, and the spacing as well as the bandwidth is determined by a constant mel frequency interval.

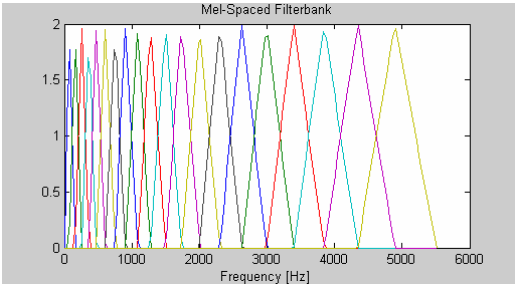

Fig. 6 Mel spaced filter bank

The relation between linear frequency and mel frequency is as follows:

$$Mel(f)=2595* \log 10 (1+ f / 700) \quad (8)$$

E. MFCC Computation

A block diagram of the structure of an MFCC processor is as shown (Fig. 7). The main purpose of the MFCC processor is to mimic the behavior of the human ears.

In first step, the continuous speech signal is blocked into frames of N samples, with adjacent frames being separated by M (M < N). Typical values for N and M are N = 256 and M = 100.

The next step in the processing is to window each individual frame so as to minimize the signal discontinuities at the beginning and end of each frame.

Typically the Hamming window is used.

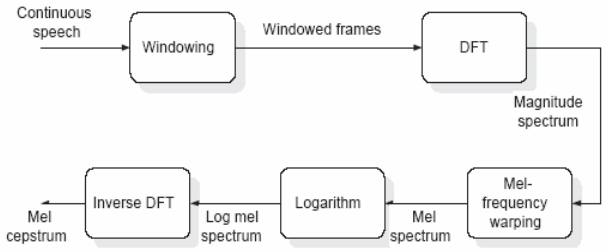

Fig. 7 MFCC calculation

The next processing step is the Fast Fourier Transform, which converts each frame of N samples from the time domain into the frequency domain. After that the scale of frequency is converted from linear to mel scale. Then logarithm is taken from the results. In final step, the log mel spectrum is converted back to time domain. The result is called the mel frequency cepstrum coefficients (MFCC). The cepstral representation of the speech cepstrum provides a good representation of the local spectral properties of the signal. Using triangular filter bank, we obtain significant decrease in amount of data. But for more simplicity in next computations, more decreasing in amount of data is needed. For this purpose vector quantization algorithm is used [5].

F. Vector Quantization

Vector quantization (VQ) is used for command identification in our system. VQ is a process of mapping vectors of a large vector space to a finite number of regions in that space. Each region is called a cluster and is represented by its center (called a centroid). A collection of all the centroids make up a codebook. The amount of data is significantly less, since the number of centroids is at least ten times smaller than the number of vectors in the original sample. This will reduce the amount of computations needed when comparing in later stages [2],[4].

Even though the codebook is smaller than the original sample, it still accurately represents command characteristics. The only difference is that there will be some spectral distortion.

G. Codebook Generation

There are many different algorithms to create a codebook. Since command recognition depends on the generated codebooks, it is important to select an algorithm that will best represent the original sample. For our system, the LBG algorithm (also known as the binary split algorithm) is used. The algorithm is implemented by the following recursive procedure [2], [5],[6] :

1. Design a 1-vector codebook; this is the centroid of the entire set of training vectors (hence, no iteration is required here).

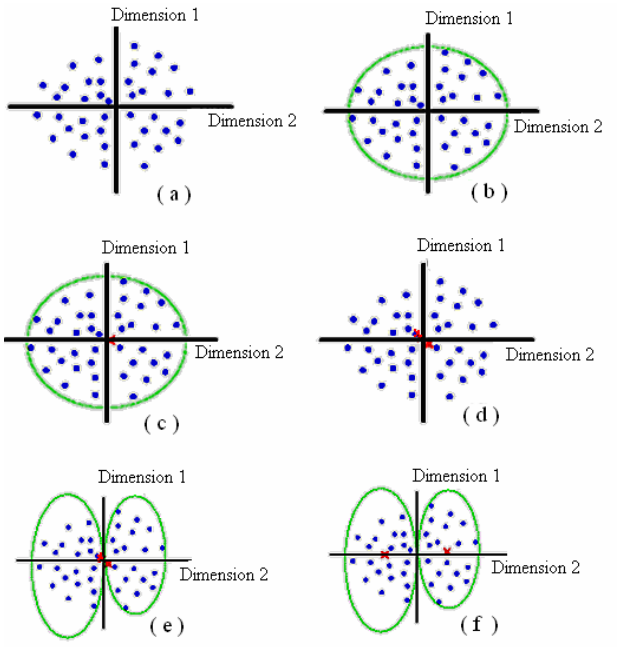

Fig. 8 The process of VQ codebook generation; the features are shown by blue dots, the group boundary in green and the centroids are in red

2. Double the size of the codebook by splitting each current codebook y_n according to the rule: where n varies from 1 to the current size of the codebook, and e is the splitting parameter. For our system, e = 0.001.

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

3. Nearest-Neighbor Search: for each training vector, find the centroid in the current codebook that is closest (in terms of similarity measurement), and assign that vector to the corresponding cell (associated with the closest centroid). This is done using the K-means iterative algorithm.

4. Centroid Update: update the centroid 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 of size M is reached.

H. Command Matching

In the recognition phase the features of unknown command are extracted and represented by a sequence of feature vectors $\{x_1 \ldots x_n\}$.

Each feature vector in the sequence X is compared with all the stored codewords in codebook, and the codeword with the minimum distance from the feature vectors is selected as proposed command For each codebook a distance measure is computed, and the command with the lowest distance is chosen.

One way to define the distance measure is to use the Euclidean distances:

$$D = \left(\sum (x_i - y_j)^2\right)^{1/2} \quad (10)$$

Fig. 9 describes the schematic of the Nearest Neighbor search [4].

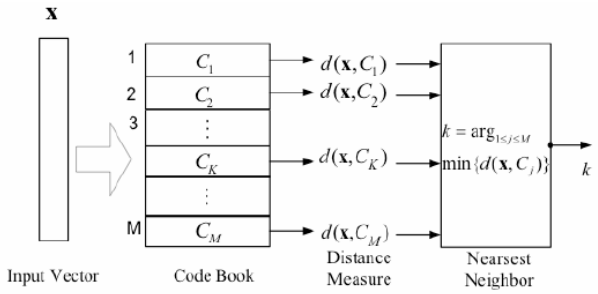

Fig. 9 A schematic of the Nearest neighbor search on the VQ decoding process

As we see, the search of the nearest vector is done exhaustively, by finding the distance between the input vector X and each of the codewords C1-CM from the codebook C. The one with the smallest distance is coded as the output command.

III. Experimental Results

To implement proposed voice command recognition system, a system with 20 voice commands was considered. Some Commands are as follow: start, stop, up, down, forward, backward, increase, decrease, left, right, fast and slow.

Training phase was done in two forms. First system was trained with one repetition for each command and once in

each testing sessions. With this type of training error rate is about 15%.

In second form, speaker repeated the words 5 times in a single training session, and then twice in each testing session. By doing this zero error rate in recognition of commands was achieved.

IV. Conclusion

As a result of changes in shape of human vocal tract during generation of different words, resonance frequencies of vocal tract, formants, also changes. Using this phenomenon, we can extract voice features of each command and we can implement a voice command recognition system.

In training phase, if stated voice commands contain more vowel differences between them, we will have more accurate recognition system. Accuracy of system also increases if we increase number of repetitions for each command in training stage.

References

[1] Deller J.R. Hansen, J.H.L &.Proakis J.G., (1993), Discrete-Time Processing of Speech Signal, New York, Macmillan Publishing Company .

[2] Rabiner, L. R. and Juang, B.-H. (1993), Fundamentals of Speech Recognition, Prentice-Hall, Englewood Cli_s, NJ.

[3] Winther Jørgensen and Lasse Lohilahti Mølgaard, IMM-THESIS-2006, Tools for Automatic Audio Indexing

[4] Christian Spanner 2005, Speech codec identification for Error Correction of Across-Channel effects in speech coded environments

[5] B. Richard, january, 2001, "Text-independent speaker recognition using source based features", Master of philosophy, Wildermoth Griffith University Australia

[6] Tejaswini Hebalkar, Spring 2000 Voice Recognition and Identification System Final Report 18-551 Digital Communications and Signal Processing Systems Design

[7] Nilsson Magnus, October 2001, Speaker Verification in JAVA, A thesis submitted in partial fulfillment of the requirements for the degree of Master of Computer and Information Engineering, School of Microelectronic Engineering, Griffith University.