# Performances Comparison between Improved DHMM And Gaussian Mixture HMM for Speech Recognition

Shing-Tai Pan
Department of Computer Science and Information Engineering
National University of Kaohsiung
Kaohsiung, Taiwan, R.O.C.
E-mail: stpan@nuk.edu.tw

Ching-Fa Chen
Department of Electronic Engineering,
Kao Yuan University
Kaohsiung, Taiwan, R.O.C.
Email:cfchen@cc.kyu.edu.tw

Wei-Der Chang
Department of Computer and Communication
Shu-Te University
Kaohsiung, Taiwan, R.O.C.
Email: wdchang@stu.edu.tw

Yi-Heng Tsai
Graduate School of Computer Science and Information Engineering
National University of Kaohsiung
Taiwan, R.O.C.
E-mail: yi_heng@seed.net.tw

*Abstract*—**This paper compares the performances, recognition rate and computation speed, between an Improved Discrete Hidden Markov Model (DHMM) and Gaussian Mixture Hidden Markov Model (GMHMM) for Mandarin speech recognition. The fuzzy vector quantization (FVQ) is used to improve the modeling of DHMM for the speech recognition. A codebook for DHMM will be first trained by K-means algorithms using Mandarin training speech feature. Then, based on the trained codebook, the speech features are quantized by the fuzzy sets and then are statistically applied to train the model of DHMM. Experimental results in this paper will show that the speech recognition rate can be improved by using FVQ algorithm to train the model of DHMM. The recognition rate by using an improved DHMM is only a little bit less than that by using GMHMM. However, the computation time for speech recognition by using improved DHMM is much less than that by using GMHMM. These results reveal that the improved DHMM is more suitable to real-time applications than GMHMM.**

***Keywords- computation time; Fuzzy Vector Quantization; Speech Recognition; Discrete Hidden Markov Model***

## I. INTRODUCTION

Recently, human life depends on electronic products more heavily due to the development in IT technology. The interface between these products and user is quite important. Since the computation ability of CPU is enormously enhanced, the speech control for an electronic product becomes more realizable. Recently, the topic on the process of audio signal attracts more attention [1-3]. There are many researches about speech recognition [4-6] because speech recognition becomes more and more important and will be a standard interface between human and electronic products in the future.

Concerning speech recognition technique, a former recognition technique is Dynamic Time Warping (DTW) [4] which used dynamic programming [7] to calculate the difference between the target speech and testing speech to recognize the testing speech. Then, Artificial Neural Network (ANN) was proposed to replace DTW for speech recognition. Because that the structure of ANN will be fixed after it is determined, the recognition rate can't be improved by online learning with more additive speech signals. Recently, Hidden Markov Model (HMM) [8] was widely applied on speech recognition [9-10]. It can solve the problem arises from variant speech speed and be constructed layer by layer to achieve automatic speech recognition (ASR). Before speech recognition, speech signal have to be pre-processed. The pre-process of speech signal includes speech sampling, point detection, pre-emphasis, Hamming window and features capture. After these processes, we can evaluate the probabilities of every HMM model corresponding to each speech and find the model which has highest probability to be the result of recognition. Consequently, in this paper, the HMM is adopted to be the speech recognition algorithm. Moreover, in order to reduce the number of data for computing, the DHMM is used here. The feature of speech signal which was used in this paper is obtained by Mel-Frequency Cepstrum Coefficient (MFCC) [8]. However, as most researchers on speech recognition well know, the GMHMM is more precise than DHMM and then some better recognition results than those by using DHMM will be obtained. To compensate this drawback of DHMM, an improved DHMM is used in this paper.

Indeed, the codebook for the speech feature quantization plays an important role on the training of DHMM model. A well-trained codebook will enhance the total performance of the speech recognition systems. In the past, the speech features are quantized through a codebook by finding the closest cluster. This winner-take-all algorithm can not perform well on vector quantization, since multiple level of an element in some vectors exists in many applications. Consequently, the fuzzy vector quantization is used improve the performance of the vector quantization. Many research show that the fuzzy vector quantization outperforms VQ [11]. Besides, the relative works of fuzzy vector quantization (FVQ), e.g. the fuzzy clustering [12-13] and fuzzy data retrieval systems [14] also demonstrate the benefit of fuzzy quantization method. This paper uses FVQ to improve the DHMM [10]. The experiments in the end of this paper will show that the computation speed for speech

978-1-4244-9306-7/11/$26.00 ©2011 IEEE

recognition by using improved DHMM is much faster than that by using GMHMM with just a little degradation in speech recognition rate. This means that the improved DHMM is more suitable to real-time applications.

This paper is organized as follows. The speech pre-processes used in this paper are first introduced in Section II. In section III, the improved DHMM by using FVQ is investigated. The experiments of the speech recognition system are then presented in Section VI where the speech recognition rates and computation time by using improved DHMM and GMHMM are compared. Some conclusions are made in the final section, Section V.

## II. Speech Pre-Processing

### A. Speech Sampling

The continuous speech signal which was recorded by a microphone must be transformed into discrete data because computer only can process discrete data. All the values which were recorded at any specific time can describe the wave of speech. Unsuitable sampling frequency is an important reason for the loss of speech. Higher sampling frequency will loss less data but has to deal with more data while lower sampling frequency will loss more data but has less data to be processed. According to sampling theorem [15]: sampling frequency can not smaller than 2 times of the signal bandwidth, we adopt 8kHz to be sampling frequency because the bandwidth of speech signal is smaller than 4kHz.

### B. Point Detection

The recorded speech sound signals will include speech segments, silence segments and background noise. The process to separate speech segments and silence segments is called End-Point Detection (EPD), see please Fig. 1 for example. If the unnecessary parts, i.e. the silence segment, are removed, the number of frames for recognition will decrease, and then the recognition speed will be enhanced.

There are many algorithms for speech sound signal EPD, which can be roughly divided into three types according to the different domain for representing the signal: (1) time-domain EPD; (2) frequency-domain EPD (3) mixed-parameter EPD. Among them, time-domain EPD is one of the simplest and the most popular ways. But it has the disadvantage of weak anti-noise capacity. As for frequency-domain EPD and mixed-parameter EPD, both have stronger anti-noise capacity and hence are more precise in recognition. But the disadvantage is that more complex calculation is needed for frequency-domain analysis.

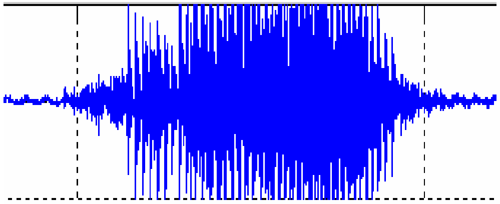

Fig. 1 End-point detection

### C. Pre-emphasis

A speech signal will attenuate in magnitude when it spreads via air. The signal with higher frequencies will attenuate more seriously. In order to compensate the attenuated magnitude of high-frequency speech signals, the speech signal will be fed into high-pass. The high-pass filter used in this paper is governed by the equation as follows.

$$S(n) = X(n) - 0.95X(n-1), \quad 1 \le n \le L;$$

In the equation (1), $S(n)$ represents the signal that has been processed with pre-emphasis, while $X(n)$ represents the original signal, and $L$ is the length (number of sampling) of each audio frame.

### D. Hamming Window

The purpose to apply Hamming window to each frame of speech signals is to avoid the discontinuity exists between every two frames and in both ends of every frames. By multiplying by Hamming window, the influence of non-continuity will decrease (to make each audio frame more centered on the frequency spectrum). Hamming window can be expressed by the following equation:

$$W(n) = \begin{cases} 0.54 - 0.46\cos(\dfrac{2n\pi}{N-1}), 0 \le n \le N-1; \\ 0, otherwise; \end{cases}$$

$$F(n) = W(n) \times S(n);$$

in which $N$ is the length of audio frame; $S(n)$ is a frame of speech signal; $W(n)$ is the Hamming window and $F(n)$ is the result of speech signal multiplied by Hamming window.

### E. Feature Capture

In speech recognition, the methods commonly used for extracting the feature of speech signals can be divided into two main categories: one is time-domain analysis, and the other is frequency-domain analysis. The way of the time-domain analysis is more direct and time-saving, with fewer operations. On the other hand, the frequency-domain analysis has to take Fourier transform on the signal, so it needs more operations and is more complicated and hence leads to the requirement of much more computation time compared to time-domain analysis. The most popular methods for features extraction are Linear Predict Coding (in time domain) and Cepstrum Coefficient and MFCC (in frequency domain) [8]. Because MFCC is more close to the distinction made by human ears toward speech sound, we use it to extract the feature for speech sound in this paper. The processes of MFCC are described as follows. First, each audio frame is transformed to frequency domain, says $|X(k)|$. Due to masking effect in sound, we make the energy in each frequency domain $|X(k)|$ be multiplied by a triangle filter as follows.

$$B_m(k) = \begin{cases} 0, k < f_{m-1} \\ \dfrac{k - f_{m-1}}{f_m - f_{m-1}}, f_{m-1} \le k \le f_m \\ \dfrac{f_{m+1} - k}{f_{m+1} - f_m}, f_m \le k \le f_{m+1} \\ 0, f_{m+1} < k \end{cases} \quad (1)$$

where $1 \le m \le M$ and $M$ is the total number of the filters. After accumulating and applying the $log(.)$ function, we can get a energy function

$$Y(m) = \log\left\{ \sum_{k=f_{m-1}}^{f_{m+1}} |X(k)| B_m(k) \right\}. \qquad (2)$$

Applying the Discrete Cosine Transform on $Y(m)$, we then obtain

$$c_x(n) = \frac{1}{M}\sum_{m=1}^{M} Y(m)\cos\left(\frac{\pi n(m-\frac{1}{2})}{M}\right) \qquad (3$$

)

in which $c_x(n)$ is the obtained MFCC.

## III. Speech Recognition Platform

After speech pre-processing, features of speech are available. Then these features will be fed into recognition platform for recognition. The recognition platform in the paper is DHMM. Before the modeling of DHMM, the FVQ for the speech features is introduced.

### *A. Fuzzy Vector Quantization (FVQ)*

In order to train the DHMM model, a codebook should be set up first. In the codebook, feature vector must be classed into some cluster by vector quantization [8]. The *K*-means algorithm is adopted to train the codebook at head. Then, combing the fuzzy set, the codebook is arranged as a fuzzy codebook and all the speech features includes speech in training phase and in testing phase will be fuzzy vector quantized through the fuzzy codebook. Suppose a set of real numbers are expressed as $\{v_1, v_2, \cdots, v_n\}$ in which $v_i \in R$ and the corresponding fuzzy membership degree are expressed as $\{u_i(x) \mid i = 1, 2, \cdots n\}$, where $\sum_{1}^{n} u_i(x) = 1, \forall x \in R$. The triangular fuzzy set is depicted in Fig. 2, in which $A_i, i = 1, 2, \cdots, n$ are the non-uniform distribution membership functions with respect to the vectors of the codebook $v_i, i = 1, 2, \cdots, n$. It is noted that since the elements of the codebook are not uniformly distributed in the domain of the speech signal after the training by *K*-means algorithm, the triangular fuzzy set is not symmetric in this application. The membership function in Fig. 2 is described as [10]

$$u_i(x) = \begin{cases} \dfrac{x - v_{i-1}}{v_i - v_{i-1}}, & \forall x \in [v_{i-1}, v_i], \\ \dfrac{x - v_{i+1}}{v_i - v_{i+1}}, & \forall x \in [v_i, v_{i+1}]. \end{cases}$$

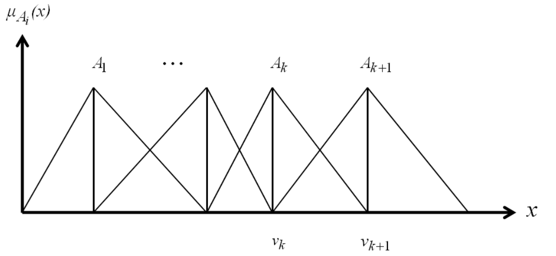

Fig. 2 Fuzzy set corresponding to each element of codebook

### *B. Discrete Hidden Markov Model (DHMM)*

The DHMM is a double layers random process. The transfer of hidden states will correspond to the transfer of observations. Each model of DHMM can describe a specific speech. The features of a speech are the observations used to estimate the hidden states. The target speech can be recognized by calculating the probability of the DHMM model. The model with highest probability in all DHMM models represents the most possibility of the recognized speech corresponding to the model [8].

$\lambda$ : DHMM model, $\lambda = (A, B, \pi)$

*A*: $A = [a_{ij}]$, $a_{ij}$ is the probability of state $x_i$ transferring to state $x_j$, $a_{ij} = P(q_t = x_j \mid q_{t-1} = x_i)$

*B*: $B = [b_j(k)]$, $b_j(k)$ is the probability of *kth* observation which is observed from the state $x_j$, i.e., $b_j(k) = P(o_t = v_k \mid q_t = x_j)$

*π*: $\pi = [\pi_i]$, $\pi_i$ is the probability of the case where the initial state is $x_i$, $\pi_i = P(q_1 = x_i)$

*X*: the state vectors of DHMM, $X = (x_1, x_2, \cdots, x_N)$

*V*: the observation event vector of DHMM, $V = (v_1, v_2, \cdots, v_M)$

*O*: the observation results of DHMM, $O = o_1, o_2, \cdots, o_T$

*Q*: the resulting states of DHMM, $Q = q_1, q_2, \cdots, q_T$

To train the DHMM model parameters $\lambda = (A, B, \pi)$ based on existing data, some notations are defined for convenience as follows:

$E_{ij}$ : the event of the transition from state $x_i$ to state $x_j$

$E_{i\bullet}$ : the event of the transition from state $x_i$ to other states

$E_{\bullet j}$ : the event of the transition from other states to state $x_j$

$E_{hi}$ : the event of state $x_i$ appears at initial state

$n(E_{ij})$ : the number of the transition from state $x_i$ to state $x_j$

$n(E_{i\bullet})$ : the number of the transition from state $x_i$ to other states

$n(E_{\bullet j})$ : the number of the transition from other states to state $x_j$

$n(E_{\bullet j}, o = v_k)$ : the number of enter to state $x_j$ and observation code is $v_k$

$n(E_{hi})$ : the number of the event of state $x_i$ appears at initial state

For training the $A, B,$ and $\pi$ of DHMM, the hidden states for each observation are estimated first through the initial

$A, B,$ and $\pi$ . Then these values $n(E_{ij})$ , $n(E_{i\bullet})$ , $n(E_{\bullet j})$ , $n(E_{\bullet j}, o = v_k)$ , $n(E_{hi})$ are found for the whole training data. Subsequently, the elements in matrices $A, B,$ and $\pi$ are updated as follows.

$$\overline{a_{ij}} = \frac{n(E_{ij})}{n(E_{i\bullet})}, \tag{4}$$

$$\overline{b_j}(k) = \frac{n(E_{\bullet j}, o = v_k)}{n(E_{\bullet j})}, \tag{5}$$

$$\overline{\pi_i} = \frac{n(E_{hi})}{n_{TD}}, \tag{6}$$

where $n_{TD}$ is the number of training data. The transition array $A$ stores the probability of state $x_j$ following state $x_i$ . Then the above method is repeated until the $A, B,$ and $\pi$ converge.

The number $n(E_{\bullet j}, o = v_k)$ is obtained through the fuzzy codebook as follows. Suppose that the speech features are classified into *n* classes by *K*-means algorithm, i.e., the trained codebook is expressed as $CB_{n \times m} = \{FK_1, FK_n, \cdots, FK_n\}^T$ where $FK_i \in R^m$ is the center vector of *i*th class. The fuzzy degree contribute to the number $n(E_{\bullet j}, o = v_k)$ for each speech feature $f \in R^m$ is computed as $u_i(v_i + \|FK_i - f\|)$ for $n(E_{\bullet j}, o = v_i)$ and $u_i(v_{i+1} - \|FK_{i+1} - f\|)$ for $n(E_{\bullet j}, o = v_{i+1})$ . This tactic is not only used for the training phase of the DHMM model but also for the testing phase of the speech recognition. In this paper, the strategy of using DHMM to recognize the speech signal is that we train first DHMM model λ for each corresponding speech by fuzzy codebook. Then, in testing phase, the tested speech features which is viewed as a sequence of observation *O* are fed into each DHMM model. The highest probabilities are found from computing the probability for all models by using the following equation:

$$P(O \mid \lambda) = \sum_{allQ} P(O, Q \mid \lambda)$$
$$= \sum_{q1, q2, \ldots, qT} \pi_{q1} \cdot b_{q1}(o_1) \cdot a_{q1q2} \cdot b_{q2}(o_2) \cdot a_{q2q3} \cdots a_{qT-1qT} \cdot b_{qT}(o_T) \tag{7}$$

Figure 3 reveals the processes for the training of the fuzzy codebook and DHMM model. Besides, the speech recognition process also can be found in the figure.

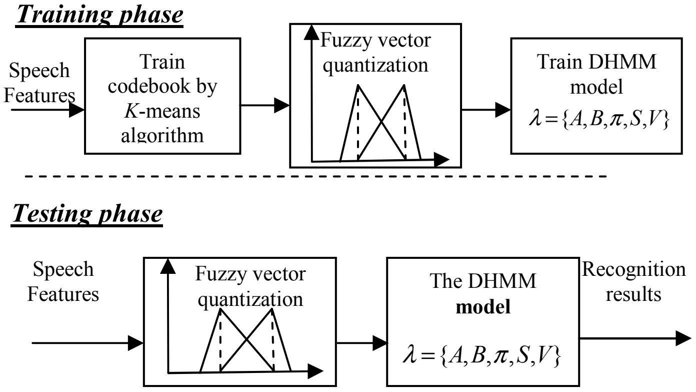

Fig. 3 Strategy of the improved DHMM [10]

## IV. Experiment of Improved DHMM Speech Recognition Systems

In speech pre-processing stage, the recording format is 8kHz, single channel, and 16bits length. The length of frame is 256 sampling points. The overlapping rate of frame is 50%. We adopted for time domain end point detection and calculated the threshold by the following formula:

$$Threshold = 7.5\% \times \max[E(n)] + \frac{1}{K}\sum_{i=1}^{k} E(i);\ 1 \le n \le N \tag{8}$$

in which $E(i)$ represents the energy of *i*th frame and *N* is the number of frames.

As for the implementation of DHMM on speech recognition platform, the number of hidden states can be arbitrarily set and the number of observation is set to be the number of the cluster in the codebook which is used to quantize the speech signal to be identified. In this application, we use 7 hidden states, 64 observations (the number of cluster in codebook) for DHMM. Consequently, the dimension of the codebook in this experiment is 64×13. Moreover, the dimension of the matrices A, B, and $\pi$ , in the DHMM model are 7×7, 7×64, and 7×1, respectively. Every model starts at state 0 and all the states only can jump to next or next 2 states, see Fig. 4 for detail. Fig. 5 presents the frames corresponds to the state-observation diagram for DHMM.

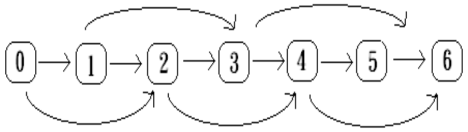

Fig. 4 DHMM states transfer structure

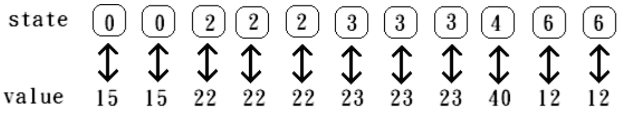

Fig. 5 DHMM corresponding state-observation diagram

A codebook is trained by *K*-means algorithm with training speech signals. The DHMM then be modeled for each speech based on the trained codebook. The speeches which will be recognized are 0 ~ 9. And hence there are 10 DHMM models after the training phase. Each number was recorded and recognized by DHMM 100 times.

The improved DHMM strategy in Fig. 3 is then used to implement the speech recognition systems. The testing speech signal includes 0 ~ 9 clean signals. The experiment is performed by three different ways: one is that only *K*-means algorithm is used for training the codebook and DHMM model (without FVQ), and the other one is that the FVQ is used to train the codebook and DHMM model, the last one is to use GMHMM model. The results for clean speech signal are listed in Table 1 and Table 2. From Table 1, it can be seen that if the FVQ is used to improve the DHMM modeling the speech recognition rates are improved compared to DHMM modeling without FVQ. It is obvious that the speech recognition rate by using FVQ is better than that without FVQ. Moreover, compared to GMHMM, only 1% degradation occurs in average speech recognition rate of FVQ+DHMM. As for the computation time for speech recognition listed in Table 2, we can see that the computation time by using DHMM+FVQ is about only one half of that by using GMHMM. Consequently, we can conclude that the strategy DHMM+FVQ is more suitable to real applications than GMHMM.

Table 1 Comparison of recognition rates

| Speech | ***DHMM*** | ***DHMM+FVQ*** | ***GMHMM*** |
|---|---|---|---|
| 0 | 1 | 1 | 1 |
| 1 | 1 | 0.94 | 0.96 |
| 2 | 1 | 1 | 1 |
| 3 | 0.96 | 0.96 | 0.98 |
| 4 | 0.92 | 0.98 | 0.96 |
| 5 | 0.94 | 1 | 0.96 |
| 6 | 0.92 | 0.96 | 0.98 |
| 7 | 0.94 | 0.94 | 0.98 |
| 8 | 0.92 | 0.96 | 0.98 |
| 9 | 0.96 | 0.96 | 1 |
| **Average** | **0.956** | **0.97** | **0.98** |

Table 2 Comparison of computation speed (sec.)

| Speech | ***DHMM*** | ***DHMM+FVQ*** | ***GMHMM*** |
|---|---|---|---|
| 0 | 0.0369 | 0.0375 | 0.074 |
| 1 | 0.03654 | 0.0378 | 0.076 |
| 2 | 0.03686 | 0.03688 | 0.07 |
| 3 | 0.03566 | 0.0325 | 0.068 |
| 4 | 0.0334 | 0.03378 | 0.068 |
| 5 | 0.03532 | 0.0372 | 0.071 |
| 6 | 0.03628 | 0.03684 | 0.074 |
| 7 | 0.03622 | 0.03656 | 0.072 |
| 8 | 0.0377 | 0.03778 | 0.076 |
| 9 | 0.03512 | 0.03582 | 0.069 |
| **Average** | **0.036** | **0.036266** | **0.0718** |

## V. Conclusions

The performance, speech recognition rate and between DHMM, DHMM+FVQ and GMHMM are compared in this paper. The fuzzy vector quantization (FVQ) had been used on the modeling of Discrete Hidden Markov Model (DHMM) to improve the speech recognition rate for the Mandarin speech. All the speech features should go through the FVQ based on the fuzzy code book before being fed into the DHMM model for recognition. Experimental results reveal that the speech recognition rate can be improved by using FVQ algorithm to train the model of DHMM. Moreover, using DHMM+FVQ strategy, we can improve the computation time from GMHMM for speech recognition and hence can realize the real-time applications.

## Acknowledgment

This research work was supported by the National Science Council of the Republic of China under contract NSC 99-2221-E-390-027.

## References

[1] X. Huang, Y. Abe, and I. Echizen, "Capacity Adaptive Synchronized Acoustic Steganography Scheme," *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 1, No. 2, pp. 72-90, Apr. 2010

[2] J. McAuley, J. Ming, D. Stewart, and P. Hanna, "Subband Correlation and Robust Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, Vol. 13, pp.956-964, 2005.

[3] K. Yamamoto and M. Iwakiri, "Real-Time Audio Watermarking Based on Characteristics of PCM in Digital Instrument," *Journal of Information Hiding and Multimedia Signal Processing,* Vol. 1, No. 2, pp. 59-71, Apr. 2010.

[4] C. Wan and L. Liu, "Research and Improvement on Embedded System Application of DTW-based Speech Recognition," *International Conference on Anti-counterfeiting, Security and Identification*, pp. 401 – 404, 2008.

[5] T. Kinjo and K. Funaki, "On HMM Speech Recognition Based on Complex Speech Analysis," *Conference on IEEE Industrial Electronics*, pp. 3477 – 3480, 2006.

[6] H. Sayoud and S. Ouamour, "Proposal of a New Confidence Parameter Estimating the Number of Speakers -An experimental investigation," *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 1, No. 2, pp. 101-109, Apr. 2010.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms 2nd Editon.* McGraw-Hill, 2002.

[8] X. Huang, A. Acero, and H. Wuenon, *Spoken Language Processing A Guide to Theory, Algorithm and System Developmen.* Pearson, 2005.

[9] J. Tao, L. Xin and P. Yin, "Realistic visual speech synthesis based on hybrid concatenation method," *IEEE Trans. on Audio, Speech, and Language Processing*, Vol. 17, No. 3, pp. 469-477, 2009.

[10] Shing-Tai Pan, Fuzzy Vector Quantization on The Modeling of Discrete Hidden Markov Model for Speech Recognition, *International Journal of Fuzzy Systems*, Vol. 13, No. 2, pp. 130-139, June 2011.

[11] W. Pedrycz and K. Hirota, " Fuzzy Vector Quantization with The Particle Swarm Optimization: A Study in Fuzzy Granulation-Degranulation Information Processing," *Signal Processing*, Vol. 87, pp. 2061-2074, 2007.

[12] V. Kapoor, S. S. Tak, and V. Sharma, " Location Selection – A Fuzzy Clustering Approach," *International Journal of Fuzzy Systems*, Vol. 10, No. 2, pp. 123-128, June 2008

[13] C. H. Li, W. C. Huang, B. C. Kuo, and C. C. Hung*, "* A Novel Fuzzy Weighted C-Means Method for Image Classification," *International Journal of Fuzzy Systems*, Vol. 10, No. 3, pp. 168-173, June 2008

[14] A. Lakdashti, M. S. Moin, and K. Badie, "Reducing the Semantic Gap of the MRI Image Retrieval Systems Using a Fuzzy Rule Based Technique," *International Journal of Fuzzy Systems,* Vol. 11, No. 4, pp. 232-249, December 2009

[15] S. Haykin and B. V. Veen, *Signals and System*s *2nd Edition.* Wiley, 2003.