A Speech Analysis System Based on Vector Quantization Using the LBG Algorithm and Self-Organizing Maps

Norbert Ádám
Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics
Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic
Email: norbert.adam {at} tuke.sk

Abstract— **Speech is the basic means of communication of people. In addition to having certain content, it also contains important acoustic information, which allows us to identify the speaker. The identification of persons has long been interesting for researchers. There is a whole set of methods aimed at this purpose, commonly called biometrics. However, if we want to identify a person by his/her voice, we have to analyze his/her speech and decide about the identity by means of specific properties of the voice. This article provides an overview of the possibilities of processing voice recordings and describes a speaker identification system using two different approaches. The first approach is based on vector quantization and the LBG algorithm, while the second approach uses self-organizing maps (SOM).**

Keywords – identification; self-organizing maps; speaker, vector quantization;

I. INTRODUCTION

Sound can be defined as each mechanical wave traveling in a material environment, resulting in oral perception. The scientific definition contains "Sound is an alteration in pressure, particle displacement or particle velocity propagated in an elastic material or the superposition of such propagated alterations. Sound is also the sensation produced through the ear by the alterations described above." [1]. The frequency of sound audible to humans ranges from 16 Hz to 20 kHz. Sounds we hear are not arbitrary phenomena, in most cases they are speech, music or singing. Speech is the basic means of communication of people. It also contains important acoustic information, which allows us to identify the speaker. By speaker identification we mean a specific process, being part of the more general problem of speaker recognition.

Speaker recognition is a term comprising all tasks related to differentiate people according to their voice. In general, we may separate speaker recognition to speaker verification and speaker identification [2]. The speaker verification process is based on the comparison of the voice model of the "unknown" speaker and the stored voice model (hereinafter referred to as "model") of a known and authenticated speaker. The speaker verification is a typical two-class decision problem [3]. Identification is the comparison of the model of the "unknown" speaker with the stored models of known speakers. The most similar speaker is selected as the one identified [4]. The speaker identification is a typical multiple-class decision problem [3]. The speaker is – in this case – only an abstract notion, represented by the corresponding model in the system. Speaker model creation is the processing phase in automatic speech recognition (ASR) systems, where a specific model and/or template is created from the input speech features [5].

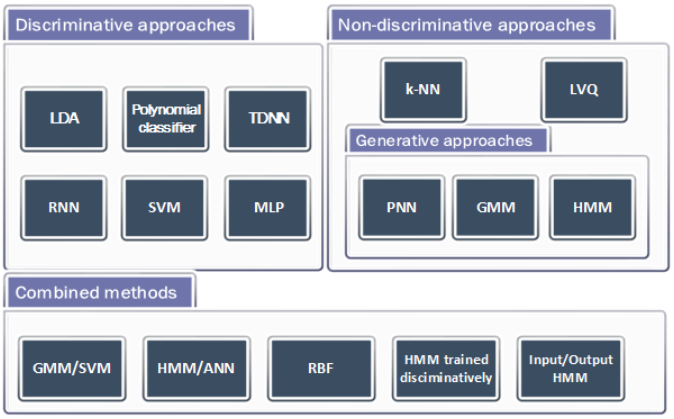

Figure 1. An overview of main classification/modeling approaches [3]

When selecting the right model we have to take the method of speaker recognition into account. In practice we distinguish text-dependent and text-independent systems. In text-dependent systems, recognition is not bound only to the characteristics of the speaker's voice, but also to the text uttered by the speaker. In text-independent recognition, the system must be capable of distinguishing the speakers only by using the speech features, regardless of the text [6]. These systems are more universal, though they have a higher error rate – i.e. false acceptance of an "unknown" and false rejection of a known speaker. During model creation, speaker recognition systems use various approaches, which may be classified as discriminative and non-discriminative/generative (Fig. 1).

II. DISCRIMINATIVE SPEAKER MODELING

In the speaker identification process, the discriminative speaker modeling approach attempts to minimize the

appearance of erroneous and/or false recognitions by determining the border parameters of the classes containing the characteristic features of the speech signals, the feature vectors of the speech signal [4]. The disadvantage of this speaker modeling approach is complicated model training (modification). If there is any change in the set of trained models, e.g. another speaker is added, all models have to be retrained (the borders have to be set up anew). On the other hand, non-discriminative methods do not have this problem, since each speaker is trained independently. Discriminative modeling uses techniques known under the names Linear Discriminant Analysis (LDA) [7], Polynomial Classifier [8], Time-Delay Neural Network (TDNN), Recurrent Neural Network (RNN) [9], Multi-layer Perceptron (MLP) or Support Vector Machine (SVM) [10][11].

III. NON-DISCRIMINATIVE SPEAKER MODELING

As non-discriminative modeling approaches we classify all methods based on template matching or the so-called generative models [2][6].

A. *Direct template matching methods*

In previous studies written on the subject, direct template matching between the training and testing templates was proposed. This technique compares directly the templates from the set of tested and trained vectors using some similarity metric. Normally, Euclidean distance or Mahalanobis distance is used for this purpose [5]. Known methods of direct template matching are Dynamic Time Warping (DTW) and Vector Quantization (VQ).

The most popular text-dependent algorithm using the template matching approach is the DTW algorithm. DTW uses the technique of dynamic programming to process text-dependent input feature vectors. In general, DTW may be defined as an algorithm, the aim of which is to determine the similarity of two sequences, which differ in the time or speed of speech. If we focus on the problem of speaker identification, the goal of the algorithm is to remove the dependency of the speaker from the speed of speech.

The second, frequently used method is the VQ method based on modeling the speaker classes using a codebook. The book consists of multiple templates – code words. These code words are being created during training by clustering the acoustic vectors of the particular speaker. VQ is thus an effective way of compressing large feature vectors. Special algorithms are used for clustering, such as the k-means or LBG (Linde–Buzo–Gray) algorithm. When creating the codebook, these algorithms calculate the mean time information stored in the data. On one hand it may be an advantage, since it is not necessary to align the code words (which significantly simplifies the system), however, on the other hand we lose the time-dependent information included in the voice, typical for the speaker in question [12].

B. *Generative speaker modeling*

Generative models try to capture the basic distribution of training vectors as a class of centroids and the changes in the surroundings of these centroids. To put it in other words, they are trained to represent the whole distribution space in the most efficient way, using the training data as a starting point. The most widespread generative models are stochastic models, such as Gaussian Mixture Models (GMM) and Hidden Markov Models (HMM) [2].

GMM were proposed by Reynolds for speaker recognition [13]. Currently, the most widely used (text-independent) modeling technique is based on probability. The parameters of the model are calculated in the training phase in a way providing the most efficient estimation of the training vector distribution. For this purpose the Expectation Maximization (EM) algorithm may be used [14]. If appropriate input data are available, the GMM method gives excellent results. Reynolds pointed this out in another study, when he compared the efficiency of GMM and some other methods [15]. A disadvantage of this method is that it needs a lot of input data to make the recognition successful. To prevent this problem, the Universal Background Model (UBM) has been introduced for this purpose. This model is created by gathering and training large amount of voice samples from various speakers of both sexes, which makes it an independent model. The new speaker model is then produced by adapting the UBM model using the training data of the speaker in question. In this case, the adaptation occurs by means of MAP (maximum a posteriori) parameter estimation [5].

The Hidden Markov Model is based on the idea that during speech, the mutual positions of the voice organs change and thus influence the pressure passing through them. Therefore we may describe the voice canal as a series of states, which are typical to certain sounds. If we split speech into short sections of time, we may consider these to be discrete, while speech is a set of transitions between the individual sound generating states. Therefore, we may model speech as finite state automaton [16]. The theory behind the Hidden Markov Model was described by Baum et al. in a series of articles (1966, 1972). One of the first applications of the HMM was a voice recognition system. The HMM method is useful in text-dependent speaker recognition, when some password or some phrase uttered by the speaker must be modeled. It is also useful in language specific phoneme analysis [17].

IV. ANALYSIS OF THE PROBLEM OF SPEAKER IDENTIFICATION USING A VOICE RECORDING

Since the goal of the proposed solution is classification by voice, we face the problem of speaker identification. The solution to the speaker identification problem has multiple steps. The first step is to add some voice – of a certain speaker – known to the user to the system in form of a recording, e.g. in wav or mp3 format. The second step is to remove noise from the recording. The third step is to create a speaker model, which will serve as a basis for the system to identify the speaker later on. This model is created during the training phase, after which it remains unchanged. The above steps are repeated for an arbitrary number of speakers, whom we want to identify with the system. The next two steps are identical to the first and second, except for the fact that we do not know, whose voice is in the recording (the goal is to identify the

speaker). In the last step we let the ASR decide, whose voice it is.

ASR systems determine the identity of the speaker in two ways. The first is to use an open set of speakers. In this case the system may decide, if the voice in the recording belongs to one of the speakers stored in the database or it doesn't belong to any of them (i.e. it is a foreign person, not registered in the database). The second possibility is to have the system work with a closed set of speakers and even if the recording contains a voice not belonging to anybody from the database, the system shall identify the voice as one stored in the database, using the highest probability [4]. The system proposed in this article uses the latter way of identification.

The flexibility of the proposed system is ensured by the possibility to add or remove speakers any time, without the necessity to retrain the existing speakers, i.e. it allows updates to the existing speaker. The number of persons (voice recordings) processed in real-time is limited by the hardware running the system. The time required for the identification depends on the total number of persons in the system.

Applying the above steps, the system of the activity consists of four main phases (Fig. 2). These are:

- Voice recording import;
- Recording analysis;
- Model creation;
- Speaker identification.

A. *Voice recording import*

This phase allows the system to work with the selected voice recording to analyze it and display its waveform. The supported input formats are wav and mp3. Any input audio recording is converted to wav audio format. The raw data of the recording were encoded as float values (which allows sufficient precision) and temporarily stored in binary form on the disk.

B. *Recording analysis*

The analysis of the recording consists of signal preprocessing and the extraction of features. Preprocessing should adapt the input signal to the appropriate form for further processing. The s(n) speech signal may contain noise – a component not containing any information, but which occurs due to the transfer channel, the sound card, etc., . However, when determining certain features of the speech signal (Short Term Zero Crossing Rate, Short Term Energy Parameter) [19], it may have an adverse effect on further processing, especially during the calculation of short term energy, therefore this signal has to be centered out. If the length of s(*n*) is *n* samples, we may calculate the direct component as the mean (1), which we detract from all s(*n*) samples.

$$\bar{s} = \frac{1}{N}\sum_{i=1}^{N} s(i). \qquad (1)$$

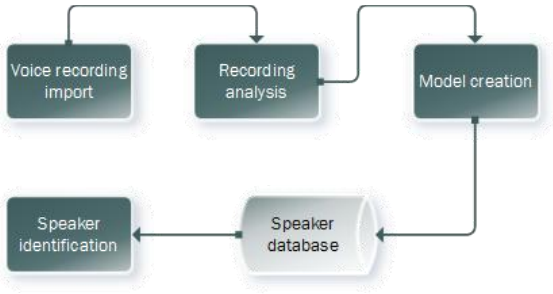

Figure 2. The logical structure of the system

To emphasize the higher frequency components of the voice signal and level out its natural fadeout, we apply a pre-emphasis filter, implemented as a single coefficient Finite Impulse Response (FIR) filter. The calculation is as follows:

$$y(n) = s(n) - \alpha * s(n-1). \qquad (2)$$

Where, s(n) is the nth speech sample, y(n) is the corresponding pre-emphasized sample and α is the pre-emphasis factor reflecting the degree of pre-emphasis, typically having a value between 0.95 and 0.97. We have set this parameter to 0.97 in the proposed system, because 0.97 de-emphasizes lower frequencies more than 0.95. Pre-emphasis ensures that in the frequency domain all the formats of the speech signal have similar amplitude so that they get equal importance in subsequent processing stages [20]**Error! Reference source not found.**.

Further processing occurs in frames, representing relatively short periods of time (10 ms), during which the speech is considered stationary (i.e. does not change dramatically) [2].

The speech recording contains in most cases so-called silent places, i.e. parts not containing speech, just noise. This silence is unwanted during the processing. This noise was removed using speech detector using frame energy accumulation as its basis. In the following processing phase the calculation of acoustic features is being done. A calculation using mel-frequency cepstral (MFC) coefficients based on the first 13 coefficients has been used. More coefficients model the properties of the voice better, however the first 13 coefficients are sufficient to capture the characteristic features of the voice.

The robustness of the system has been increased by capturing signal dynamics. For this purpose, delta coefficients were used, which were derived from the existing MFC coefficients. We have used first grade and second grade coefficients (acceleration coefficients) – these may be calculated as stated at [21].

C. *Methods of speaker model creation*

The way of speaker modeling depends on the selected method of speaker model creation. We implemented two different approaches of speaker model creation. The first approach is based on creating a codebook, thus it is based on modeling using VQ (see section II.A) and the LBG algorithm. The second approach is based on creating a model using

Kohonen self-organizing maps (SOM). The advantage of the second approach – compared to LBG-based modeling – is the capability to capture the correlation of an input feature vector and multiple centroids simultaneously. This is due to the fact that SOMs use the method of competitive learning. A best matching unit (BMU) is selected for each input vector – using Euclidean distance – the weights of which are influenced most. However, then the weights of the neighboring units are updated in a specified surrounding, with a decreasing tendency [22]. This feature of the map should result in the creation of a better codebook and also to better results of the final identification, compared to the LBG method.

D. Methods of speaker identification

Since we have decided to model the speakers in the proposed system using a codebook, the identification of the unknown sample was related to the calculation of the distance to each model in the database. The calculation was based on the total Euclidean distance between all feature vectors of the input recording and the given speaker model. The computation result was then stored in a data structure containing information on the speaker with the least distortion and the best matching score.

The database of speaker models is being accessed intensely only in this phase. The speaker model consists of two parts. The first part is a logical model stored in the database, containing information on the speaker as a physical person (name, surname, etc.), along with a reference to the storage location of the second model – the codebook – on the disk. This means that only the logical model is loaded into the memory and the codebook is loaded only on demand.

To increase the recognition capabilities, we have used weighting. The idea behind this is to weight the code words according to their efficiency in the decisive process by calculating the similarity of the respective codebooks. For each code word of the book we are calculating the weights for we calculate the distance to the closest word of the other books. The smaller the distance between the words, the more they are similar, thus the final weight of the given code word will have a smaller value. The higher code word weights mean that these words have a good discriminative capability in regard of the other codebooks (classes). These weights are then used to calculate the similarity score of the input acoustic feature vector in comparison to the given book. The equation used for the calculation of similarity is therefore the following:

$$s_w\left(X, C_i\right) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{d\left(x_t, c_{min}^{i,t}\right)} * w\left(c_{min}^{i,t}\right). \qquad (3)$$

V. EXPERIMENTAL PROOF OF THE FUNCTIONALITY OF THE PROPOSED SYSTEM

To prove the functionality of the proposed system (fig. 3) we have selected samples coming from two sources. The first is the author's own database containing samples recorded using a microphone and a PDA device. The second source consisted of the recordings available at the VoxForge speech corpus [23]. These are recordings of various quality (studio recordings, as well as recordings created by means of a microphone and a

Figure 3. User interface of the designed system

PDA device), mostly single channel, with a sampling frequency amounting to 8 or 16 kHz. The sound samples were available in wav and mp3 format.

The average length of the available recordings was approx. 25 s, recorded at a sampling frequency of 8 and 16 kHz. The sound samples were then subsequently converted to an 8 kHz frequency wav format. Like this a set recordings was created, containing a total of 393 samples of 100 various speakers, coming from both sexes.

The set of samples was separated into two groups: 100 training samples (all from different speakers) have been used to create a database of speakers. Then, we have used the remaining 287 samples of 94 speakers as testing samples. To find out the overall score of the individual algorithms, we have used a full database of 100 speakers and tried to associate 94 testing samples to them.

The initial setting of the system parameters is stated in table no. 1.

TABLE I. THE INITIAL SYSTEM PARAMETER SETTINGS

Parameter	Value
Pre-emphasis filter coefficient	0.97
Frame size (ms)	10
Window size (ms)	25
The alpha coefficient of the window function	0.16
Window function type	Hamming window
MFCC count	13
Linear filter count	13
Log filter count	27
Bottom frequency limit	133
Top frequency limit	4000
Max. no. of iterations	100

A. *Achieved results: Test no. 1*

With the first test we tried to find out the overall score of the system when identifying speakers. The success rate of the algorithm was calculated as the ratio of the correctly identified samples to the total number of testing samples. The result achieved by SOM was 98.93% (93 correctly identified speakers of 94) and 96.8 % for LBG (91 of 93).

B. *Achieved results: Test no. 2*

The second test was to show the impact of the size of the acoustic model of the result of the identification with the aim of specifying its optimum size, which would produce acceptable results in a reasonable time required for processing. For this purpose we have selected 35 correctly identified samples and their sizes (16 – series1, 64 – series2, 128 – series3, 256 – series4) and compared them mutually. The results achieved by LBG are in fig. 4, while the results achieved by SOM in fig. 5. As it is evident: with the increasing model size, the ratio of matches goes up, too. There is a significant difference between sizes 16 and 64. Further increases of the difference in model size are not that significant (e.g. series3 vs. series4). The average matching score of the algorithms during identification for the individual acoustic model sizes is stated in table no. 2.

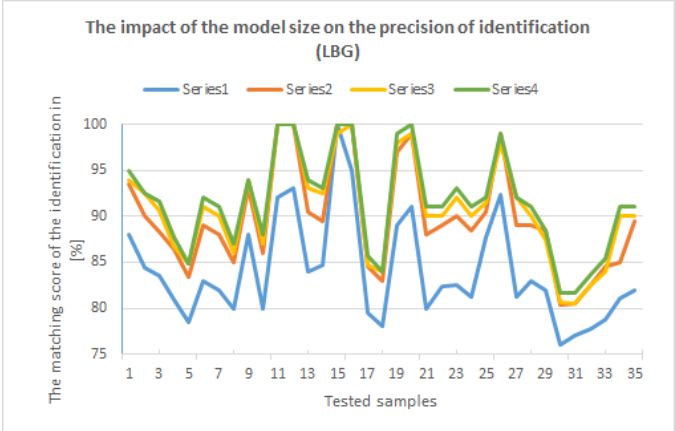

Figure 4. Matching score : LBG

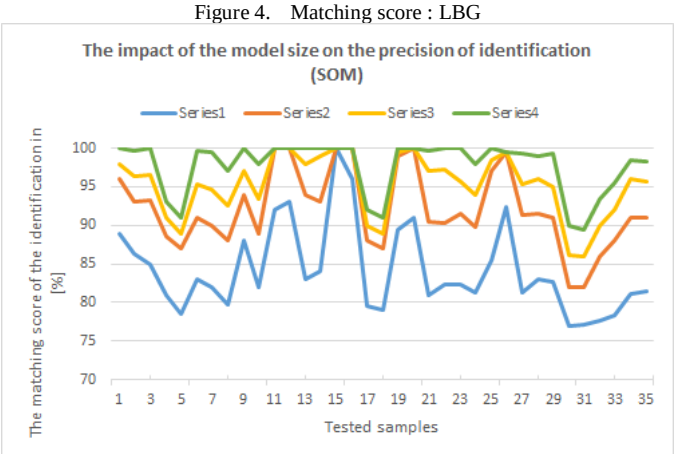

Figure 5. Matching score : SOM

TABLE II. THE AVERAGE MATCHING SCORE IN % ACHIEVED BY THE RESPECTIVE ALGORITHMS WITH MODELS OF VARIOUS SIZES.

	16	**64**	**128**	**256**
LBG	84	89,71	90,73	91,49
SOM	84,18	92,09	95,26	97,73

VI. CONCLUSION

The presented paper discusses the problems of speaker voice analysis and recognition with the aim of identification. In accordance with the proposed solution we have described the phases of processing required for speaker identification. As a means of speaker modeling and identification we have chosen a method based on creating a codebook with vector quantizing and the LBG algorithm. As an alternative to this method we have used codebook creation using SOMs. With the aim of increasing the matching score we have proposed an optimization, in the form of code words weighting in the code book. To measure the similarity of the voices at the input of the system and the model in the database we have chosen the Euclidean distance. The proposed system was implemented as a standalone application created using the .NET framework in C#, version 4 using WPF technology to build the graphical user interface. The architecture of the application was based on the Model-View-ViewModel architectural pattern with the aim to minimize the dependence of the application and the presentation layer. The efficiency of the system was verified by a series of tests. For this purpose we have selected two speaker databases. We have focused on determining the overall matching score of the respective implemented methods and to follow the correlation of the matching score and the model size. Best results have been achieved using a codebook of 256 code words; however the average difference between correct speaker identification using a codebook of 256 code words and a codebook of 64 code words was only 3%.

ACKNOWLEDGMENT

This work was supported by the Slovak Research and Development Agency under the contract No. APVV-0008-10.

REFERENCES

[1] H.F. Olson, Elements of Acoustical Engineering. 1957.

[2] N. Zheng, "Speaker Recognition Using Complementary Information from Vocal Source and Vocal Tract", The Chinese University of Hong Kong, 2005.

[3] T.D. Ganchev, "Speaker Recognition", Department of Computer and Electrical Engineering University of Patras Greece, 2005.

[4] M.S. Chavan, S.V. Chougule, "Speaker Features And Recognition Techniques: a Review" in International Journal Of Computational Engineering Research (IJCER), Vol. 2, Issue No.3, 2012, pp. 720-728. ISSN: 2250–3005.

[5] H.S. Jayanna, S.R. Mahadeva Prasanna, "Analysis, Feature Extraction, Modeling and Testing Techniques for Speaker Recognition" in IETE Technical Review, Vol. 26, Issue 3, 2009; pp. 181-190.

[6] T. Kinnunen, H. Li, "An Overview of Text-Independent Speaker Recognition: from Features to Supervectors" in Speech Communication, Vol. 52, Issue 1, 2010, pp. 12-40.

[7] L. Lu, "Probabilistic Linear Discriminant Analysis for Acoustic Modelling" in IEEE Signal Processing Letters, Vol. 21., No. 6, 2014, pp. 702-706.

[8] W.M. Campbell, K.T. Assaleh, "Speaker Recognition With Polynomial Classifiers" in IEEE Transactions on Speech and Audio Processing, Vol. 10, No. 4., 2002, pp. 205-212.

[9] R.L.K.Venkateswarlu, R. Vasantha Kumari, G.Vani JayaSri, "Speech Recognition By Using Recurrent Neural Networks" in International Journal of Scientific & Engineering Research Vol. 2, Issue 6, 2011, pp. 1-7

[10] A. Khan, M. Farhan, A. Ali, "Speech Recognition – Increasing Efficiency of Support Vector Machines" in International Journal of Computer Applications, Vol. 35, No. 7, 2011, pp. 17-21.

[11] A. Ganapathiraju, J. Picone, "Applications of support vector machines to speech recognition" in IEEE Transactions on Signal Processing, Vol. 52, No. 8, 2004, pp. 2348-2355.

[12] J. P. Campbell, "Speaker Recognition: A Tutorial" in Proceedings of the IEEE, Vol. 85, No 9. 1997, pp. 1437-1997.

[13] D.A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models" in Speech Communication , vol. 17, pp. 91-108, 1995.

[14] A. Dempster, N. Laird, D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm" in Journal of the Royal Statistical Society 39(1) (1977) 1-38

[15] D.A. Reynolds, R.C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models" in IEEE Trans. Speech Audio Process. , vol. 3, pp. 72-83, Jan. 1995.

[16] M. Gaspierik, "Standard methods for voice recognition" [Štandardné metódy rozpoznávania hlasu]. [online, cit. 07-July-2014] Ref. http://keeper.blog.matfyz.sk/p15244-standardne-metody-rozpoznavania-hlasu

[17] K. M.O Nahar, M. Elshafei, W. G. Al-Khatib, H. Al-Muhtaseb, Statistical Analysis of Arabic Phonemes for Continuos Arabic Speech Recognition, International Journal of Computer and Information Technology, Vol. 01, Issue 02, 2012, pp. 49-61

[18] Gy. Györök, M. Makó, "Acoustic Noise Elimination by FPAA, 3rd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence, SACI 2006, pp. 571-577.

[19] Short Term Time Domain Processing of Speech [online, cit 07-July-2014]. Ref. http://iitg.vlab.co.in/?sub=59&brch=164&sim=857& cnt=1.

[20] B. Singh, V. Rani, N. Mahajan, "Preprocessing In ASR for Computer Machine Interaction with Humans: A Review" in International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, Issue 3, March 2012, pp. 396-399.

[21] Delta, Acceleration and Third Differential Coefficients [online, cit. 07-July-2014] Ref. http://www1.icsi.berkeley.edu/Speech/docs/HTKBook3.2/node67_mn.html

[22] Machine Learning: Self Organizing Feature Maps [online, cit 07-July-2014]. Ref. http://dame.dsf.unina.it/machine_learning.html#sofm

[23] VoxForge: Speech Corpus [online, cit 07-July-2014]. Ref. http://www.repository.voxforge1.org/downloads/SpeechCorpus/Trunk/