Automatic, Text-Independent, Speaker Identification and Verification System Using Mel Cepstrum and GMM

Automatic, Text-Independent, Speaker Identification and Verification System
Using Mel Cepstrum and GMM

Ahmad Al Marashli
Communication dep.
HIAST
Damascus, Syria
ahmadmmrs@yahoo.com

Dr. Oumayma Al Dakkak
Department of Electronic and Mechanical Systems
HIAST
Damascus, Syria
odakkak@hiast.edu.sy

***Abstract*—Our aim is to build a reliable Speaker Identification and Verification system, independent of Text, and working in a quasi real time. Such a system can be used for authentication purposes in the Access Systems, or in multimedia applications for the separation of utterances of various speakers.**

Keywords-speaker identification; speaker verification; Gaussian Mixture Model (GMM); Mel Cepstrum.

I. INTRODUCTION

Speaker Recognition is part of a wider area of speech signal processing called Speaker Classification; which refers to the process of extracting information about an individual from his/her speech [1]. From the speech itself, a listener can make some accurate guesses as to whether the speaker is male or female, adult or child. He may understand the speaker's mood and emotional state. While it is not that hard for a listener to distinguish the identity of many people from their voices, it's not easy for computers, a tutorial study of different issues of the problem are discussed in the literature, see for instance [2, 3]. The present work is an attempt to make Speaker Identification and Verification, from online speech in a quasi real-time. Part II is devoted to features extraction from speech, Mel-Cepstrum is used in our system, as it proved to be very effective in speech recognition. Part III describes the technique adopted for recognition which is based on Gaussien Mixtures Models (GMM). Part IV shows implementation issues, trying to accelerate the recognition process and reach a quasi real time. Results are presented in part V, followed by our conclusion.

II. FEATURES EXTRACTION

Speech parameterization consists of extracting feature vectors out of speech signal, in view of speech analysis, recognition or speaker recognition. The aim of this parameterization is to obtain a new representation which is more compact, less redundant, and more suitable for statistical modeling or for the calculation of a distance or any other kind of score. Most of the features used in speaker recognition systems rely on a cepstral representation of speech. We will expose the filter bank-based cepstral parameters or mel-frequencies cepstral coefficients (mfcc) used in our project. Other types of features could be taken [4, 5, 6].

A. Voice source

Speech signal is captured by a high quality microphone, in room conditions, then sampled at rate of 8000 samples/sec, and digitized by 16 bits/sample. Another voice source, without using the H.Q.microphone, will be declared when we will talk about the database used to test the system.

B. Speech detection

Important point before feature extraction is to get the speech signal uttered by a speaker out of the whole captured signal, in order to have the feature from just the speech signal, and to minimize the unnecessary processing for silence moments. This process is done by computation of both energy and zero crossing of the successive slots of the signals, to identify utterance boundaries. This simple and efficient algorithm of speech activity detection is described in [7,8]. Figure (1) shows speech signal and the resulting detection of speech activities (rectangular windows)

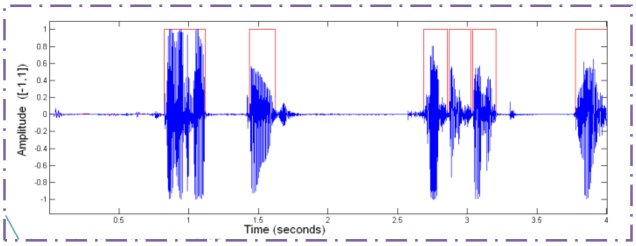


Figure 1 speech activity detection algorithm's results

C. *Preemphasizing filter:*

Pre-emphasizing the speech signal is usually a step done to eliminate the radiation effect at the lips, by enhancing the high frequency of the spectrum [9], and that is done by passing the signal into an FIR filter, equation (1) represents the difference equation of this filter.

$$x_p(t) = x(t) - a \cdot x(t-1). \tag{1}$$

The value of "a" in the above equation, is generally taken in the interval [0.95, 0.98] [9]. Our practical experiments reflect 10% decrease in error rates with pre-emphasized signal.

D. *Feature vector:*

Cepstral coefficients are calculated locally on 30 ms speech segments, windowed by hamming window and 50% overlapped. Next step is the movement into frequency domain, by applying the FFT algorithm with 256 points (see equation 2). We care only about the envelope of the spectrum, that reflect the zeros and poles of the vocal tract transfer function (the main useful information to identify the speaker).

$$X(n,w) = \sum_{m=-\infty}^{+\infty} x[m].W[n-m].e^{-jwm} \tag{2}$$

$$x[\ \ ]: speech\ signal \quad W[\ \ ]: hamming\ window$$
$$W[m] = 0\ for\ m > N$$
$$n\ is\ the\ window\ number, N\ window\ length$$

Then we weigh the spectrum values by a series of filter frequency responses whose center frequencies and bandwidths roughly match those of critical auditory band filters[1], that match mel-scale bands for the range 0-4kHz[10] (see equations 2 and 3) [book], figure (2) illustrates these frequency responses.

$$E_{mel}(n,l) = \frac{1}{A_l}.\sum_{k=L_l}^{U_l} |B_l(w_k).X(n,w_k)|^2 \tag{3}$$

$$mel\ energy\ for\ frame\ n\ and\ band\ l$$
$$U_l, L_l\ \text{upper and lower imits of the band}$$
$$B_l: filter\ frequency\ response\ X: fourier\ transform\ of\ the\ signal$$

$$A_l = \sum_{k=L_l}^{U_l} |B_l(w_k)|^2 \tag{4}$$

$$normalizing\ factors$$

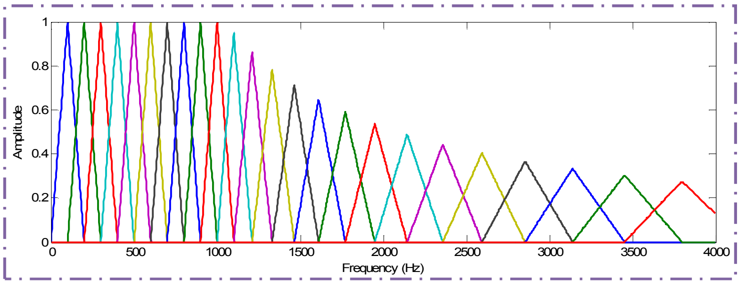


Figure (2) Mel scale bands in the range 0-4 kHz

Usually, we discard the first and last bands, because they are out of the telephony band. Hence, we just have 22 energy values (in dB) left, out of the filter-bank.

Then, an inverse Fourier transform is applied to change the spectral vector into cepstral vector (real cepstrum), which is the discrete cosine transform (DCT) of the values computed in equation 4. Equation 5 shows the computation of these mel- Cepstrum coefficients:

$$C_{mel}[n,m] = \sum_{l=1}^{L} \left(E_{mel_{dB}}(n,l)\right).cos\left[m\left(l-\frac{1}{2}\right).\frac{\pi}{L}\right] \tag{5}$$
$$m = 1,2,\dots\dots,R$$

In the above equation, R is the dimension of the final feature vector, which can be changed dynamically, in our application, to determine the number of DCT components to be used. The use of DCT helps in the de-correlation of the coefficients, to make them more suitable for probabilistic modeling [10].

III. SPEAKER RECOGNITION USING GMM

Gaussian Mixture Models have been considered as the main modeling method for speaker recognition applications [11,12], because of their probabilistic properties [13], especially when dealing with the none deterministic process of speech, especially in multi-speaker, text-independent issues. So far, we succeeded, in our application, to reach a total independency not only of the text but also of the uttered language, and of the way of speaking.

We use the multi-dimensional Gaussian probability density function to represent the speech process variability; by assigning a different probability density function (pdf), for every different sound class, and perhaps for each phoneme.

The input for the Gaussian probability density functions is represented by x, the feature vectors. The corresponding pdf functions $b_i(x)$ are computed by equation 6:

$$b_i(x) = \frac{1}{(2\pi)^{\frac{R}{2}}.|\Sigma_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_i)^T (\Sigma_i)^{-1}(x-\mu_i)} \tag{6}$$

$$\Sigma_i, \mu_i: the\ covarience\ matrix\ and$$
$$mean\ vector\ for\ the\ pdf\ (sound\ class)$$

Therefore, a speaker in GMM system, is represented by a group of pdfs (his/her sound classes); this group is the model (λ). Each sound class represents a state in that model, which is defined as a pdf by a mean vector and a covariance matrix. A probability of a vector x to be in the model λ (in any of the I states of λ), is calculated as the union of the different pdfs as follows:

$$p(x|\lambda) = \sum_{i=1}^{I} p_i.b_i(x) \tag{7}$$

As in previous equations, we weigh every pdf or state, with a coefficientp_i, to represent the density of feature

[1] Using of these auditory properties reflects an increase in performance in speaker recognition.

vectors related to that particular state. Figure 3 illustrates this modeling.

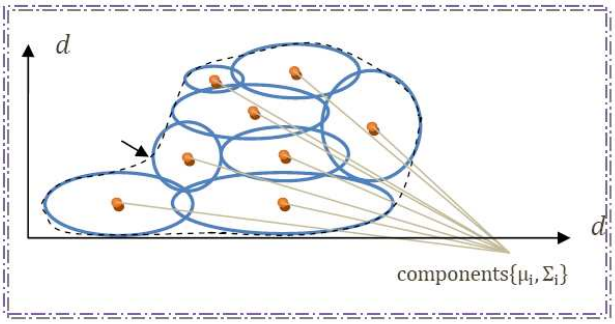


Figure 3 Gaussian mixture model for two dimensional space

In fact, we did not use a full squared but diagonal covariance matrix, due to several theoretical and practical points showing that diagonal matrix outperforms full one [13].

A. Speaker Identification:

To identify a speaker out of a closed set of previously known speakers, who have models in our codebook: $(\lambda_j)_{1\le j\le S}$, we compute the probability of each speaker model depending on every feature vector. Then, we pick the model with the highest probability (see equation 6):

$$\hat{S} = \text{Ind}\left\{ \max_{1 \le s \le S} [P(\lambda_s|x_n)] \right\} \qquad (8)$$

To find a solution for the maximum a posterior, as in previous equations; We recall the Bayes rule [10], and the problem interprets into maximizing the various $p(x_n|\lambda_j)$(see equation 9). So, with M independent feature vectors, the needed pdf to maximize is then:

$$p(\{x_1, \dots\dots, x_M\}|\lambda_j) = \prod_{m=1}^{M} p(x_m|\lambda_j) \qquad (9)$$

When dealing with logarithmic values, we write down the criterion as in equation 10:

$$\hat{S} = Ind\left\{ \max_{1 \le s \le S} \sum_{m=1}^{M} log[p(x_m|\lambda_s)] \right\} \qquad (10)$$

B. Speaker Verification:

We aim here to decide if a piece of speech belongs to a hypothesized speaker (S) or not (see figure 4), what's called a maximum likelihood detector between two hypotheses:

$$\begin{Bmatrix} H_0: (x_m)_{1\le m\le M} & belong\ to\ S \\ H_1: (x_m)_{1\le m\le M} & don't\ belong\ to\ S \end{Bmatrix}$$

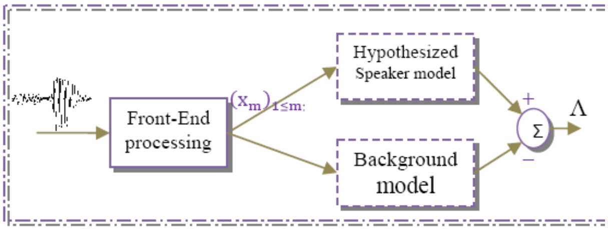


Figure 4 modeling of a speaker and its background

$$\frac{P(x_1, .., x_m|H_0)}{P(x_1, .., x_m|H_1)} = \Lambda \begin{Bmatrix} \le \theta \to accept\ \ H_0 \\ > \theta \to reject\ \ H_0 \end{Bmatrix}$$

We use here two models: λ_s : related to the hypothesized speaker, which should be already defined in the system; λ_{back} : should be built to represent other speakers (other than the hypothesized speaker).

There are two types of background: universal background model and personalized background model (different from one speaker to another). However, the universal background, which is common to all speakers, needs large databases of speech, and involves much more computations than the other one. Therefore, most applications uses the other one, forming the background for each speaker from the models of all (or some) of the other speakers known to the application [12].

C. Models training:

The training process involves building the speaker model out of his speech. This means defining each pdf $(\{\mu_i, \Sigma_i\})$and weight (p_i) for the I states, depending on the M feature vectors.

This ask is usually performed using Expectation-Maximization (EM) estimation algorithm. It is an iterative algorithm, which aims to refine GMM parameters to satisfy increasing conditional probability $p(X_{train}|\lambda)$, on each step, for the feature vectors to be in the model. Where X_{train} are the training vectors

$$X_{train} = \{x_1, \dots\dots, x_{M_{train}}\}.$$

We develop the model k+1 from the model k, at each iteration, so that equation 11 is valid:

$$p(X_{train}|\lambda^{k+1}) \ge p(X_{train}|\lambda^{k}) \qquad (11)$$

Better initial conditions for the algorithm have a tiny effect on the final score for detection. The effect of the initial conditions is studied in [14]. We initiate our models with a k-means vector quantization [15] VQ, with the centers of VQ as mean vectors, and initial weights reflects feature vectors distribution over VQ sectors, and constant values for the covariance matrix.

EM steps and theoretical analysis are well defined in [10,13,14]. Depending on the conditional probability

$p(i_n = i|x_n, \lambda^k) = \frac{p_i^k . b_i^k(x_n)}{\sum_{i=1}^{I} p_i^k . b_i^k(x_n)}$; EM could be resumed in a collection of equations as is shown in equations 12:

$$
\begin{aligned}
p_i^{k+1} &= \frac{1}{M} \sum_{n=1}^{M_{train}} p(i_n = i|x_n, \lambda^k) \\
\mu_i^{k+1} &= \frac{\sum_{n=1}^{M_{train}} p(i_n = i|x_n, \lambda^k). x_n}{\sum_{n=1}^{M_{train}} p(i_n = i|x_n, \lambda^k)} \\
\Sigma_i^{k+1} &= \frac{\sum_{n=1}^{M_{train}} p(i_n = i|x_n, \lambda^k). x_n\, x_n{}^T}{\sum_{n=1}^{M_{train}} p(i_n = i|x_n, \lambda^k)} - \mu_i^{k+1} \left(\mu_i^{k+1}\right)^T
\end{aligned}
\qquad (12)
$$

Figures 5 and 6, show respectively an example of training points in a two dimension space, and the result of GGM modeling.

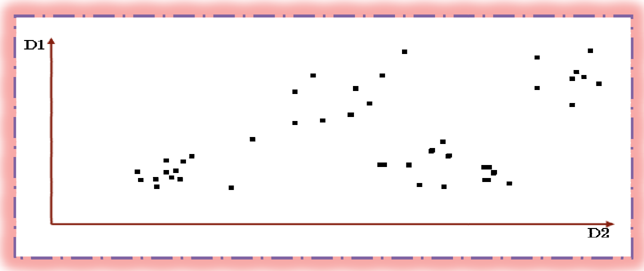


Figure 4 example of training points in 2 dimension space

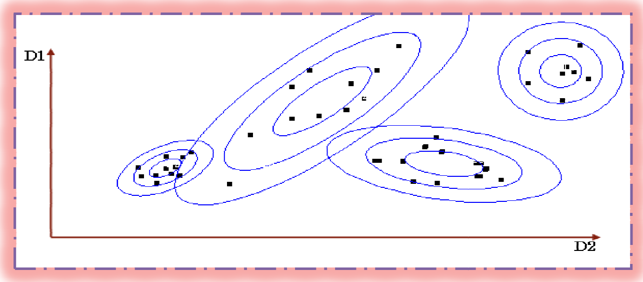


Figure 5 results of GMM of the points of figure 5

IV. APPLICATION IMPLEMENTATION

A. Implementation:

The system was first built, using Matlab, to simulate the various algorithms, and fulfill the main objectives of the project in Identification and Verification.

This first implementation under Matlab, reveals the problem of time consumption: execution time was unacceptable; Model training took about 40 seconds[2], for a three-second utterance, with feature vector dimension 20, 64 states per model, and 10 iterations for learning. While to identify a speaker out of 17 using a two-second utterance, with the same parameters as above, took about 45 seconds. These numbers are far away from real time issues.

Efforts were done to decrease algorithms' complexity, only half the execution time was saved.

The identification technique strategy was then changed: use only 40% of the speech segment to find out the best five speakers, with the maximum probability. Then, Pick the best speaker using rest of the speech segment among those five speakers. This change made a remarkable effect on execution time; no more linear dependencies of this time on the number of speakers. The identification using 2 second utterances took less than 10 sec.

A step toward real time was done by implementing the project in the C language. The execution time was decreased to 5% the previous one and we reached real time processing. The implementation is ready to be performed on signal processors.

B. Real time processing:

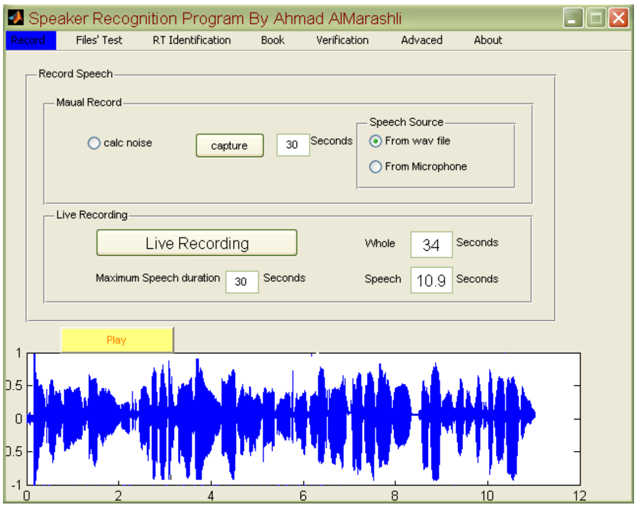
In the application and after minimizing the processing time with C functions, real time process is done by splitting time into periods or time slots, speech is recorded during a slot, while at the same time speech of the previous time slot is under process starting from deletion of silence to identification or verification on speech parts. Identification or verification decision is updated every time slot, and process ended when a final decision is taken when it remains stable for a number of steps.

In identification, the result is always shown in the application window, as the name of the speaker, and is updated every time slot. For verification the process is done as a login request and stops when it is stable on allow or refuse. And for both processes the interface shows always a mark for the detected speaker, which indicates the percentage of the recorded speech that fully matches the model of that speaker as the best model (highest probability), of the whole speech.

As far as processing time is less than the recording time, the real time processing is fulfilled. If it is not the case, we loose small shanks of speech which do not affect the application results.

With 100 speakers and 32 classes in a model, our application is rarely to loose any tiny time of speech. Used time slot is 2 seconds and could be changed dynamically through the application graphical user interface.

[2] All execution times numbers are measured using a computer with win XP running over 1.8Ghz CPU



30 Seconds

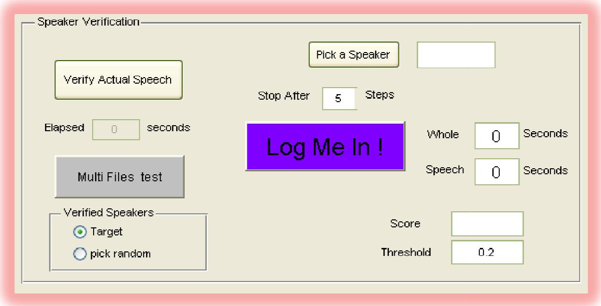
Speech Source: ● From wav file ○ From Microphone

Whole 34 Seconds

Maximum Speech duration 30 Seconds

Speech 10.9 Seconds

Figure 6 main application interface



Stop After 5 Steps

Elapsed 0 seconds

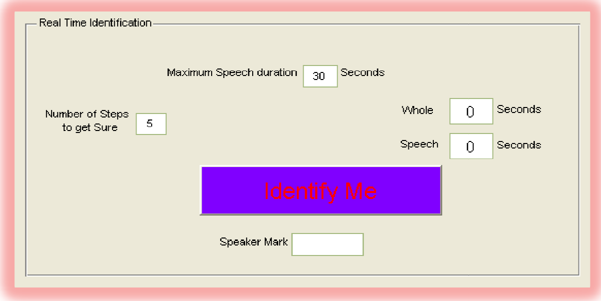
Whole 0 Seconds

Speech 0 Seconds

● Target ○ pick random

Threshold 0.2

Figure 7 application interface for verification



Maximum Speech duration 30 Seconds

Number of Steps to get Sure 5

Whole 0 Seconds

Speech 0 Seconds

Figure 9 Application interface for identification

C. DATABASE:

Our database contains two types of speech sources: the **first** is the practical and the one to use with the real application for the system as previously described, about 30 Syrian people including males, females, and few children; talking through a high quality microphone in a room conditions, mainly most of the records are in Arabic but some are not. Some speakers trained the system in Arabic then tested it with French and English. Others, tested the system with totally different accents, and with sounds like hums. The system succeeded more than once in such circumstances.

The **second** source adds lots of speakers, which is needed to calculate exact numbers to describe the system, like error rates and other quantities, and to test its ability, to hold more speakers, and still acts in real time.

Since it's not easy to have an appropriate speech database to use especially with Arabic language, and with a variety of accents to testify total independence of text; we created our own database. We recorded speech from satellite channels, received by a PCI-card, and recorded with the application, on a laptop which is connected to the receiver computer. Speech is extracted from conversations, news, documentary programs, or some songs; but each is for a single speaker without any background sound or music. Sound quality differs from good for some records to poor for others, most of the records were in Arabic language and with many accents, some in English, and a few in French.

For both sources, the sampling rate is 8000 samples/sec, at total the whole database contains 100 speakers, with an average of 35 seconds of pure speech for each speaker; added to 10 seconds used for learning, the whole database is about 4,500 seconds.

V. RESULTS

We reached 100 speakers with perfect real Identification, in more than 90% of test cases reached a right decision, and large number of them had a stable decision since the beginning of the test, and with a hundred percent mark. The average for obtained marks is about 87%, when testing on 2-second speech chunks. All of that using 32-class models and performing the training on just ten seconds of speech.

In what follows, we present sets of tests to find out the effects of various parameters of the system on the identification rates. The tests were performed on a number of speakers (SP) equals 100, taking 20 MFCC features (FEA) for each vector. The test duration (TSD), which is our time step for real time processing, is in seconds, and the training duration (TRD) is also in seconds. The number of classes for GMM is referred to as (CL). In these conditions, (CORI) is the rate of correct identifications hits, during TSD, and the rate of overall correct real time identification is (RTIR).

TABLE I. THE EFFECT, OF THE TRAINING DURATION, ON THE IDENTIFICATION RATE, FOR 32 MFCC.

SP	FEA	TSD	TRD	CL	CORI	RTIR
100	20	2	10	32	87%	93%
100	20	2	5	32	76%	69%

TABLE II. THE EFFECT, OF THE TEST DURATION, ON THE IDENTIFICATION RATE, FOR 16 MFCC.

SP	FEA	TSD	TRD	CL	CORI	RTIR
100	20	2	10	16	85%	91%
100	20	4	10	16	94%	93%

TABLE III. THE EFFECT, OF THE NUMBER OF MFCC ON THE IDENTIFICATION RATE, FOR 32-CLASS MODELS.

SP	FEA	TSD	TRD	CL	CORI	RTIR
100	10	2	10	32	86%	92%
100	20	2	10	32	87%	94%

TABLE IV. THE EFFECT, OF THE NUMBER OF CLASSES WITH 20 MFCC.

SP	FEA	TSD	TRD	CL	CORI	RTIR
100	20	2	10	8	85%	87%
100	20	2	10	16	85%	91%
100	20	2	10	32	87%	94%

These tables suggest some the following conclusions:

- Increasing the number of features (MFCC) increases the identification rate, which is predicted.
- Increasing the number of classes of GMMs increases the identification rate, however over-fitting has bad effects. 32 seems to be the best choice for this parameter.
- Increasing the test duration also increases the identification rates. 2 seconds gives a trade-off between the time response of the system and the identification rate.

Concerning the verification part of the system, we got an equal error rate (miss hit error and false alarm) of 6.5%. See figure (9), for the adopted parameters of the system (20 MFCC, 32 classes). This equal error is equal to 8.4% for 16 classes.

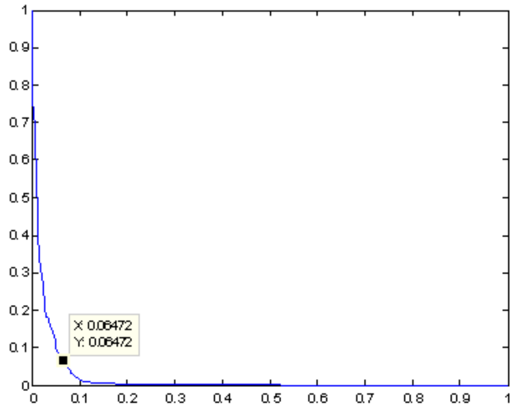


Figure 10 Detection Error Tradeoff for speaker verification (horizontal is false alarm probability, vertical miss hit probability).

VI. CONCLUSION

Based on the various tests of the effects of each parameter for the system, we adopted the following parameters:

- 20 MFCC for the features vectors,
- 32 classes for the GMMs,
- 2 seconds for the time step for the testing,
- 10 seconds for the learning phase.

Our system does not only work on the speech database of the 100 speakers, but can also work correctly, as a real time identification system, in on-line recording conditions.

Ten seconds of pure speech are sufficient for the above mentioned performances of the system. The system works independent of the text, the language and the way of speaking

Based on the various tests of the effects of each parameter for the system, we adopted the following parameters:

REFERENCES

[1] J. Campbell "Speaker Recognition: A Tutorial", proceedings of IEEE , vol.Nb. 9, sep. 1997

[2] R. D. Rodman, "Speaker recognition of disguised voices", Department of Computer Science North Carolina State University Raleigh, North Carolina, U.S.A.

[3] R. L. Klevans, R. D. Rodma, "Voice Recognition", 1997, Poston, London

[4] J. P. Campbell, D.A.Reynold and R. P. Dunn, "Fusing High- and Low-Level Features for Speaker Recognition", EUROSPEECH 2003 - GENEVA

[5] C. Y. Espy-Wilson, S. Manocha and S. Vishnubhotla, "A New Set of Features for Text-Independent Speaker Identification", Interspeech 2006, ICSLP.

[6] Haton, J. P. et al. "Reconnaissance automatique de la parole: Du signal a son interpretation", DUNOD, 2006.

[7] L.R. Rabiner and M.R. Sambur, "An Algorithm for Determining the Endpoints for Isolated Utterances", The Bell System Technical Journal, Vol. 54, N.2, Feb 1975, pp.297-315

[8] Y. Harba, Isolated words recognition system using Hidden Markov Models HMMs linked to a robot acquiring vocal commands. Internal report, 2005, HIAST.

[9] F. Bimbot, et al. "A Tutorial on Text-Independent Speaker Verification" EURASIP Journal on Applied Signal Processing 2004:4, pp.430–451

[10] T. Quatierie, "Discrete time Speech Signal Processing: Principles and practice" pearson Education, 2002.

[11] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models", M.I.T. Lincoln Laboratory, Digital Signal Processing 10, 19–41 (2000)

[12] D. A. Reynolds. W. Campbell, T. Gleason, C. Quillen, D. Sturim, P. Torres-Carrasquillo and A. Adami "The 2004 MIT Lincoln Laboratory Speaker Recognition System", Acoustics, Speech, and Signal Processing, 2005. Proceedings, Volume 1, Issue , March 18-23, 2005 Page(s): 177 - 180

[13] A. Al Marashli, Automatic text-independent speaker recognition system, internal report, 2007, HIAST.

[14] Y. Itaya, H.Zen, Y. Nankaku, K. okuda, T. Kitamura and C. Miyajima, "Deterministic Annealing Em Algorithm In Parameter Estimation For Acoustic Model", INTERSPEECH ICSLP 2004

[15] A.W. Moore "K-means and Hierarchical Clustering", 2001, on the we site www.cs.cmu.edu/~awn.