PCA Summarization for Audio Song Identification using Gaussian Mixture Models

Vaia Panagiotou
Electrical and Computer Engineering Department
Democritus University of Thrace
Xanthi, Greece
panagiotou.vana@gmail.com

Nikolaos Mitianoudis
Electrical and Computer Engineering Department
Democritus University of Thrace
Xanthi, Greece
nmitiano@ee.duth.gr

***Abstract*—In an audio fingerprinting system, the song identification task should be performed within a few seconds. To address the need for fast and robust song identification system, we design fingerprints based on Gaussian Mixture Modeling (GMM) of delta Mel-frequency cepstrum coefficients (ΔMFCC) or delta chroma features (Δchroma). In order to summarize the extracted features over time, a novel implementation of Principal Component Analysis (PCA) is introduced. Experimental evaluations performed on a database of 10000 songs confirm that the proposed PCA summarization technique provides a significant increase in speed in the system's query time. Furthermore, the fingerprints prove to be quite robust against various common distortions, while by using non-distorted test song segments of 10 seconds, the system achieves high identification rates.**

***Index Terms*—audio fingerprinting, song identification, dimensionality reduction**

I. Introduction

The multi-disciplinary field of Music Information Retrieval (MIR) is focused on extracting information from music and using this information to solve a wide range of problems, among which is *automatic song identification*. This application is becoming increasingly important due to the immense growth of audio distribution channels, including radio stations, internet services, file download and exchange services (namely peer-to-peer networks).

In the search for efficient identification techniques, scientists were led to the introduction of *audio fingerprints* [1], which are unique and compact representations of the perceptually relevant part of an audio content. Thus, a song can be identified not by comparing the digitized waveforms, which is very slow and inefficient, but by comparing the audio fingerprints only. The systems that employ the above technique are known as *audio fingerprinting systems*. A typical audio fingerprinting system consists of two fundamental processes: a) fingerprint extraction and b) modeling/ matching algorithm. The system extracts acoustic relevant features from a large music collection and stores them in a database along with their respective meta-data (including song title, artist name and album title). When an unidentified song segment is presented to the system, features are extracted from the segment and matched against those stored in the database. If a positive match is found, the meta-data associated with the song are retrieved from the database.

A typical audio fingerprinting system should fulfill several requirements, including: discriminative power over a large number of fingerprints, computational simplicity, robustness to distortion and compression. Finally, the matching algorithm should be able to identify a song from a huge database only in a few seconds [1].

Audio fingerprinting can be applied to many different applications, such as identification of songs or commercials being played on radio, television or web (broadcast monitoring), finding meta-data for unidentified songs (connected audio), filtering for file sharing applications and automatic organization of large music libraries.

There are many different audio fingerprinting implementations, depending on the features that are used to generate the audio fingerprints and the matching algorithm. In [2], Ramalingam et al. extract features derived from the short-time Fourier transform (STFT) of audio signals that are modeled using Gaussian mixture models (GMM). In this paper, we describe an audio identification system, based on the structure of Ramalingam et al. Our system extracts delta Mel-frequency cepstral coefficients (ΔMFCC) and delta chroma (Δchroma) features and uses a novel summarization technique based on principal component analysis (PCA) to reduce the amount of data. The summarized features are then modeled by GMM. The proposed technique seems to accelerate the performance of the GMM-based system.

II. Related Work

Although there are several audio fingerprinting systems [2], [1], most of them feature a similar structure. Their main differences lie in the different features they use to generate the fingerprints, e.g. Mel-Frequency Cepstrum Coefficients (MFCC), Fourier coefficients, spectral flatness features, energy-based features, timbral features, spectral peaks and differences in energy between frequency bands. In this section, we will briefly examine some existing systems related to our implementation.

The Philips algorithm [2] uses the sign of the difference between energies in 33 logarithmically spaced subbands in order to form hash strings (which are referred to as *subfingerprints*). For a fast database lookup, full fingerprint comparisons are only performed between pre-selected candidates in the first

phase of the searching algorithm. The best match is determined by the bit error rate (BER) per fingerprint block.

Microsoft's Robust Audio Recognition Engine (RARE) [2], [7] constructs an optimal set of features, using both original and distorted versions of songs as training data, in order to provide extra robustness against distortions. This system applies a two-layer oriented principal component analysis (OPCA) on the log power spectrum of the modulated complex lapped transform (MCLT) coefficients of the signal to reduce dimensionality of features. In the recognition stage, RARE uses a new method based on redundant bit vector indices, in order to avoid brute-force linear scans, which are slow.

Fraunhofer's AudioID [2] uses a set of psychoacoustic features including loudness, spectral flatness measure (SFM) and spectral crest factor (SCF) as feature vectors. The extracted features are then modeled by vector quantization (VQ) so that each song is encoded by a codebook. In the identification stage, feature vectors of the unidentified song are extracted and approximated by all stored codebooks, using some distance metric. The codebook that scores the smallest accumulated approximation error is considered the correct match.

AudioDNA [2] extracts MFCCs and converts them into a sequence of acoustic events-genes by using Hidden Markov Models (HMM). In the identification stage, feature vectors from an unidentified song are extracted and compared with the database fingerprints using an approximate matching algorithm. The database song that is closer to the fingerprint of the query song is chosen as the correct match.

Shazam's fingerprints [5] are based on spectrogram peaks. In this system, nearby peaks are combined into pairs, which are termed landmarks. Each pair is uniquely identified by two time values and two frequency values. These values are combined through a hash function, thus forming the final fingerprints. In the identification phase, the fingerprints from the unidentified song are matched against a large set of fingerprints derived from database. The candidate matches are subsequently evaluated and the database song that most closely matches with the unidentified song is used to perform identification.

III. Proposed Audio Fingerprinting Scheme

The proposed audio identification system is based on the implementation of Ramalingam et al. as described in [2]. From the block diagram shown in Fig. 1, two operating modes can be identified, namely the *training mode* and the *identification mode*. During the training mode, fingerprints from all available songs are created and stored in the database. In the identification mode, the stored fingerprints are compared with the fingerprint of an unknown song in order to find the closest match.

More specifically, the songs are first preprocessed and their features are extracted. After this procedure, we introduce a novel method to reduce the number of the feature vectors. For this purpose, Principal Component Analysis (PCA) is applied to summarize the extracted feature vectors. Traditionally, PCA is employed to reduce the dimension of the feature vectors

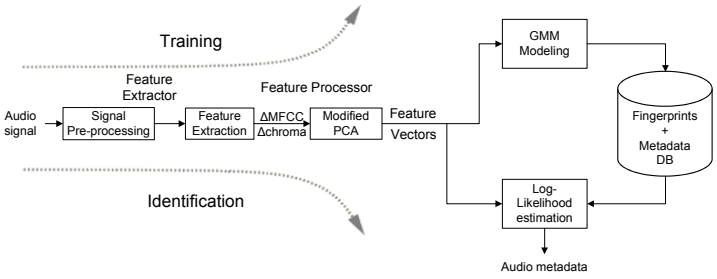

Fig. 1. The proposed Audio Identification system.

in many signal processing applications. In our method, the feature vectors dimension is retained, while PCA is used to reduce the number of vectors along time. Once the feature vectors are summarized over time, each song is modeled by Gaussian Mixture Modeling (GMM). In the identification phase, features from an unknown song are extracted and summarized in a similar manner as in the training stage. Consequently, they are used to evaluate the log-likelihood of all models in the database. The model that gives the highest log-likelihood is identified as the correct song. The proposed system will be described in more detail in the following sections:

A. Pre-processing

In the pre-processing stage, the audio signal is first converted to a general uncompressed format (16-bit, PCM). Then, it is converted to mono (if necessary) by averaging the right and left channels and finally is downsampled to 11025 Hz.

B. Feature Extraction

The feature extraction process is designed to determine representative feature vectors, which may have reduced information compared to the original audio signals, but are able to uniquely represent each song. The chosen feature set should be robust against a wide range of distortions and at the same time, the computational load should be low enough to allow feature extraction in real-time applications. Most fingerprint extraction algorithms follow a similar procedure [1]. The audio signal is segmented into frames of constant length and a window function is applied to each frame, to minimize the signal discontinuities at frame boundaries. Furthermore, to increase robustness to shifting, the frames are overlapped. The frame lengths range from 10-500 ms and the overlap varies from $50\% - 98\%$. Subsequently, a set of features is computed for each frame. In our system, we use spectral features, which have been widely exploited in music retrieval. They are extracted by performing STFT on the windowed audio segments and calculating two possible spectral features sets: either Mel-Frequency Cepstrum Coefficients (MFCC) or chroma features.

1) *Mel-Frequency Cepstrum Coefficients (MFCC)*: the MFCCs are the most popular acoustic features with extensive success in music retrieval tasks, mainly because they closely approximate the human auditory system.

They are perceptually motivated features based on the STFT and they are briefly calculated in the following way [2], [3]: The power spectrum of each extracted frame is calculated using the FFT. The power spectrum is then mapped to the Mel-scale, using a Mel-filter bank. The Mel-filter bank consists of a set of triangular overlapping band-pass filters, which are uniformly spaced on the Mel scale. The above distribution provides greater emphasis on the actual low frequencies, where the information is more important for the human ear. The log of the power at each Mel-frequency band is calculated. The Discrete Cosine Transform (DCT) is then applied, in order to reduce the correlation between coefficients and compress information into the lower-order coefficients. The MFCCs are then the amplitudes of the resulting spectrum. In this work, we keep the first 22 coefficients, excluding the zero-th order coefficient. This choice was shown to be optimal in our experiments.

2) *Chroma*: Chroma features, also called *pitch class profile* (PCP), have been already used in several music retrieval applications. Based on Shepard's helix model, which decomposes the human auditory systems perception of pitch into tone height (octave number) and chroma (pitch class), this feature takes into consideration only the chroma information. The entire frequency spectrum is projected to a 12-dimensional real-valued vector, which represents the 12 distinct semitone classes involved in a single octave, with all octaves folded together (helix). A generalized version of 24 or 36 dimensional representation is often used for higher resolution. Although the chroma representation discards the original octave information, it can still provide useful musical information about the song. In our system, we calculate the chroma features using an implementation by Ellis [6]. In this approach, the magnitude spectrogram of a music signal is extracted by applying the STFT. From this representation, only spectral peaks are mapped to the chroma spectrum, thus giving a 12-dimensional vector. Each vector element corresponds to the intensity of a semitone class (chroma).

3) *Delta coefficients*: The noise vulnerability of MFCC and chroma features significantly degrades the performance of the recognition system in noisy conditions. The first time derivatives of the above features give a new set of dynamic characteristics, known as *delta* features, which are more robust to distortions [3]. Thus, in order to enhance the system performance in noisy conditions, it is advantageous to use the delta features, instead of the original features. First time derivatives of MFCC (often known as delta-MFCC or ΔMFCC) are well known in literature and have been used in many applications. In a similar manner to ΔMFCC, we introduce the *delta-chroma* or Δchroma features, which replace the original chroma in our system, in order to increase robustness to noise. Both delta features are computed using the following formula:

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta(c_{t+\theta} - c_{t-\theta})}{2\sum_{\theta=1}^{\Theta} \theta^2} \quad (1)$$

where d_t is a delta coefficient at time t, c is the static coefficient (MFCC or chroma) and Θ is a time window in which the derivative is calculated. In our experiments, we used a time window of $\Theta = 9$.

C. Feature Summarization using Principal Component Analysis

A typical problem in these systems is the large number of features extracted from each song. This results into huge computational load and thus long time for training the database and most importantly long time for song queries. Thus it is essential to reduce the number of feature vectors without losing important information and thus accelerate the system's operation. Principal Component Analysis (PCA) is a traditional dimensionality reduction method. Here, we adapt PCA to summarize audio features over time.

The main concept of PCA is to reduce the dimensionality of a data set in which there are a large number of correlated variables, while retaining as much as possible of the variance (i.e. energy) present in the data set. This reduction is achieved by a linear transformation to a new set of variables, the principal components, which are uncorrelated and are ordered according to their importance in representing the original variables. This implies that most information can be retained by keeping only the first few principal components . Computation of the principal components is simplified to an eigenvalue decomposition problem of the data covariance matrix. Here, we propose to use PCA not to reduce the feature vectors' dimension, but to summarise and replace a group of feature vectors by their principal components.

The overview of our proposed implementation of PCA is shown in Fig. 2. First, the feature vectors, which correspond to the columns of the illustrated table, are divided into groups with 50% overlap along the time axis. Then, the original PCA method is applied to each group separately, so that each group can be replaced by a new set of vectors. The difference between the original PCA method and our method is that the latter does not reduce the dimension of the vectors, but instead reduces their number along the time, while retaining their dimensionality. For example, a group consisting of 16 22-dimensional vectors can be replaced even by only one 22-dimensional vector, without any significant loss of information. Of course, it is possible to reduce both the dimension and the number of feature vectors at the same time, but in this paper we chose to reduce only their number along time, since this lead to higher identification rates in our experiments.

Two important factors in the implementation of the above process are the choice of the number of feature vectors contained in each group and the choice of the number of principal components which represent each group. Obviously, the greater the number of principal components that are selected to represent each group, the greater the amount of

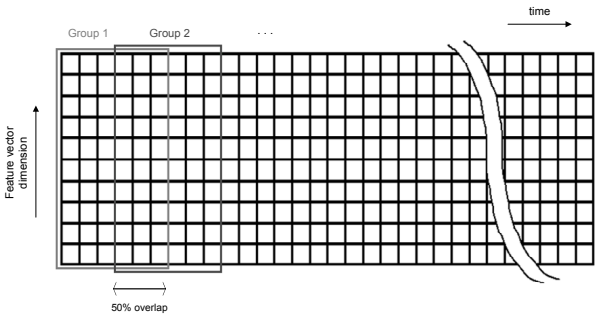

Fig. 2. Proposed implementation method of PCA for feature summarization.

the final data. In contrast, the greater the number of vectors that constitute each group, the smaller the amount of the final data, but more information will be summarized and then may be lost. Concluding, the proposed implementation of PCA for feature summarization aims firstly at accelerating the computational process both at the training and searching stages and, on the other hand, at reducing the memory requirements of the system.

D. Gaussian Mixture Modeling

Gaussian mixture modeling has been successfully employed for a variety of audio classification and retrieval tasks. In this paper, GMM is used to model an audio fingerprint as a probability density function (PDF), using a linear combination of Gaussian component PDFs (mixture). A Gaussian mixture density is defined as a weighted sum of M component densities and is given by the equation [4]:

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^{M} a_i b_i(\mathbf{x}) \tag{2}$$

where $\mathbf{x}$ is a D-dimensional random vector, $b_i(\mathbf{x})$ are the component densities and a_i are the mixture weights. Each component density is a D-variate Gaussian function of the form:

$$b_i(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma_i|}} e^{\left(-0.5(\mathbf{x}-\mathbf{m}_i)^T \Sigma_i^{-1}(\mathbf{x}-\mathbf{m}_i)\right)} \tag{3}$$

with a $D \times 1$ mean vector $\mathbf{m}_i$ and a $D \times D$ covariance matrix Σ_i. The mixture weights a_i are positive numbers that satisfy the constraint that: $\sum_i a_i = 1$. Although the general model form supports full covariance matrices, i.e. a covariance matrix with all its elements, usually only diagonal covariance matrices are used in order to reduce the computational complexity. Each GMM can be represented by the mean vectors, covariance matrices and mixture weights from all component densities. Therefore, a GMM can be described by $\lambda = \{a_i, \mathbf{m}_i, \Sigma_i\}, \forall i = 1, \ldots, M$.

Given a set of training vectors $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N\}$ and a GMM configuration, the parameters of the GMM, which best matches the distribution of the training vectors, are estimated using the *Expectation-Maximization* (EM) algorithm [4]. The equations for training a GMM using EM can be found in [4]. The EM is shown to be dependent on the initialisation of the GMM parameters. Commonly, the initial GMM model is typically derived by the k-means algorithm.

GMM can be used for automated song identification in the following manner. A set of feature vectors is extracted, as described in the previous section, from each song in the database. These feature vectors are used to train the GMM model λ_i, which is then stored in the database. Once the models for each song are stored in the database, automated song identification can be performed. A song segment is presented to the system, requiring identification. A set of the same feature vectors $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n, \ldots, \mathbf{y}_N\}$ are extracted from the song segment. In order to identify the song in the database that fits better to these feature vectors, we calculate the log-probability $\log p(\mathbf{Y}|\lambda_i) = \sum_{n=1}^{N} \log p(\mathbf{y}_n|\lambda_i)$ for all models λ_i in the database. The song model that gives the highest score indicates the song in question.

In the case of relatively low log-probability scores (below a threshold), the algorithm is unsure which song fits best to the query and is an indication that the song does not exist in the library. In this case, the algorithm sends a message that the song can not be identified.

IV. Experimental Results

To evaluate the proposed techniques, we used a database consisting of 10000 popular songs, with total duration of about 626 hours and 51.6 GB storage. The songs were in mp3 format (16-bit, 44.1 kHz, stereo) and covered 14 different musical genres including classical, country, hip hop, jazz, pop and rock music. As for the query set, 250 musical excerpts of various genres and lengths (5, 10 and 15 sec) were chosen from random parts of database's songs. All the experiments have been performed in MATLAB, on a standard PC (Intel Core 2 Quad, 2.33 GHz, 4GB RAM).

A. Evaluation of the proposed PCA summarization technique

In this section, we perform an experiment, in order to evaluate the effectiveness and efficiency of the proposed PCA-based summarization technique. In this experiment, we compare two systems: one that uses the proposed PCA summarization technique in both training and identification phases and another that does not. Of course, the above process is performed individually for each type of features. For the implementation of the proposed technique, the extracted features of each song were divided into groups consisting of 16 vectors and each group was represented by only the principal vector. These parameters were chosen after a series of experiments, in which the above combination led to higher identification rates. The audio clips that are used in the identification phase have a duration of 10 seconds and don't suffer from distortion problems.

The results shown in Fig. 3 and Table I reveal the benefits of using the proposed PCA method. When ΔMFCCs are used as features, the mean query time decreases from 3.28 min without PCA to 13.5 sec with PCA, thus giving an improvement of about 14.5 times in search time. However, when Δchroma are

used as features, the query time is much smaller compared to ΔMFCCs, because Δchroma features are of lower (12) dimension. Thus, in the case of Δchroma features, the mean query time decreases from 18.1 sec without PCA to 4.7 sec with PCA, giving an improvement of about 3.8 times in search time. Since the search time is a factor of paramount importance in real-time applications, the proposed PCA technique gives great benefit to the system, especially if the amount of the extracted data is very large. The most important aspect of this method is that it is able to reduce the search time, without reducing the identification rate, which instead is increased slightly for both features.

B. Robustness to distortions

In real-life conditions, songs may suffer from various kinds of distortions. Therefore, it is imperative that the system can achieve high identification rates in as many distortions as possible. In order to test the systems robustness, three representative distortions were applied to the query audio clips:

- Noise addition: Additive white Gaussian noise of varying SNR (Signal-to-Noise Ratio) levels (10, 15 and 20 dB) was added to the queries in order to simulate many real-world applications.
- Filtering resulting from many different processes: 1) bass boost filter, 2) bass cut filter, 3) telephone bandpass filter [300-3400] Hz, 4) bandstop filter [750-1800] Hz.
- Echo addition: This effect is used to simulate the echo produced by the reflection of sound waves on surfaces or objects.

This set of experiments assumes that the query songs exist in the database. Thus, in the identification phase, the model that gives the highest log-likehood is obtained as the correct match, without using any rejection criterion. The identification rates achieved, using both ΔMFCCs and Δchroma, are shown in Table II, for different length of queries and distortions. PCA summarization was used, as described earlier. Using ΔMFCCs instead of Δchroma seems to increase the performance in almost all cases. Also, as it was expected, the shorter the query clips, the smaller the recognition rates that are achieved, regardless of the extracted features and the distortion that was applied to them. The identification rates remain at sufficiently high levels for ΔMFCCs, even when 5-sec clips are used, while the same does not happen for Δchroma features. The latter yields high recognition rates only when audio clips of 10 and 15 seconds are used. The performance for both feature sets drops with the increase of noise and distortions. However, the ΔMFCCs behave better than Δchroma features.

C. False Positive Analysis

In previous sections, it was assumed that the query clips existed in the database. However, it is possible that some songs may not be present in the database. Consequently, there should be a criterion to reject songs that can not be recognized. This can be achieved by determining a suitable threshold in each case. Hence, in order to perform a correct identification,

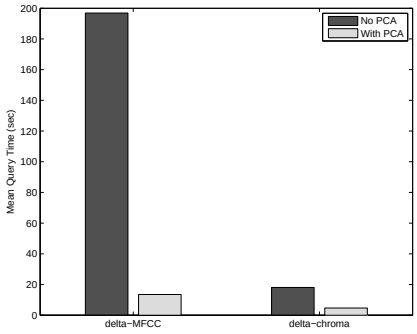

Fig. 3. Search time (in sec) for ΔMFCC and Δchroma features with and without using PCA.

the absolute difference between the first and third higher log-likelihoods observed in the system, should be above a certain threshold. After experimentation, it was observed that most query songs featured greater differences between the log-likelihoods mentioned above, compared to most of the songs that did not exist in the database. Thus, unknown songs whose difference in log-likelihoods is below the threshold can not be recognized.

By varying the threshold, different false positive and false negative rates can be achieved. The false positive or *false acceptance rate* (FAR) occurs when the system declares different songs as same, whereas the false negative or *false rejection rate* (FRR) occurs when the system rejects songs that exist in the database. A robust system is one that shows both low FAR and FRR. However, in some applications it is more important to eliminate wrong identifications, while in others it is more important to avoid rejecting registered songs.

The query set used in this experiment, consists of 250 database's excerpts and 100 excerpts that are not present in the database. Each of them has a duration of 10 seconds. The PCA summarization scheme was used, as described earlier. Figs. 4, 5 show the FRR and FAR for ΔMFCC and Δchroma, respectively. In the case of ΔMFCCs, without using a threshold, the system can not reject unregistered songs and achieves an identification rate of 96%. By introducing a threshold, the songs that are not included in the database can now be rejected. Fig. 4 shows a plot of FAR and FFR for various threshold values. The optimal threshold choice of 38 gives a false acceptance rate of 10.9% and a false rejection rate of 10.85%, while the identification rate decreases to 89.14%. When Δchroma are used as features, the system gives an identification rate of 86.4%, without using a rejection criterion. Fig. 5 shows a plot of FAR and FFR for this case. By choosing a threshold of 15, the system achieves a false acceptance rate of 21.5% and a false rejection rate of 18.52%, while the identification rate decreases to 80.57%. Optimal threshold values can be estimated by training the system with various FAR and FFR curves for various audio content.

TABLE I
IDENTFICATION RATES AND SEARCH TIME (IN SEC) WITH AND WITHOUT USING PCA.

Feature processing	ΔMFCC		ΔChroma	
	Ident. rate (%)	Mean query time (sec)	Ident. rate (%)	Mean query time (sec)
No PCA	95.6	196.9	84	18.1
With PCA	96	13.5	86.4	4.7

TABLE II
IDENTICATION RATES FOR UNDISTORTED AND DISTORTED QUERIES

Distortion Type	Identification rate (%)					
	ΔMFCC			ΔChroma		
	5 sec	10 sec	15 sec	5 sec	10 sec	15 sec
No distortion	93.2	96	98	50.8	86.4	93.2
Noise (10 dB)	45.2	59.6	66.4	46	79.2	89.6
Noise (15 dB)	66.4	82	86	44.8	80.8	90.8
Noise (20 dB)	79.2	89.6	94	45.6	81.6	91.2
Bass boost filter	88	94.4	96.8	38	76	88
Bass cut filter	88.4	94.4	97.6	51.6	86.4	92.4
Bandpass filter [300-3400] Hz	67.6	85.6	90	44.4	79.2	88.8
Bandstop filter [750-1800] Hz	87.2	93.6	97.6	11.2	25.2	44.4
Echo	53.6	73.6	80	23.2	59.6	78.8

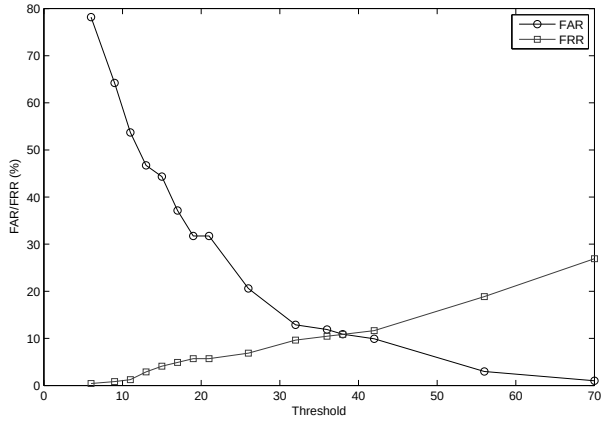

Fig. 4. FAR and FRR curves for ΔMFCCs for threshold varies between 6 and 70.

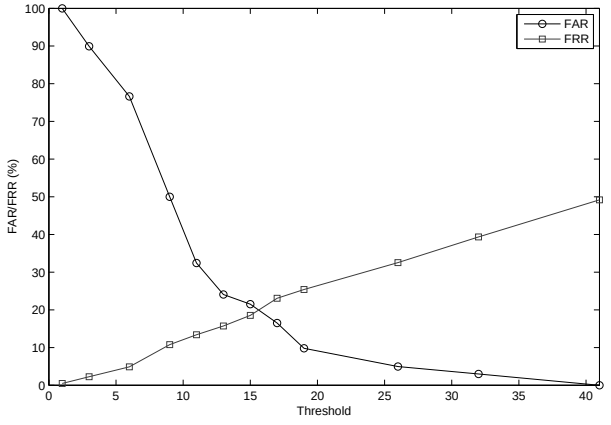

Fig. 5. FAR and FRR curves for Δchroma for threshold varies between 1 and 41.

V. CONCLUSIONS

In this paper we have presented an audio fingerprinting technique based on Gaussian Mixture Modeling of spectral features. The proposed scheme is based on the original PCA method and summarizes the extracted features over time. The scheme proves to be very efficient in achieving a significant speed-up in both training and identification phases. Using ΔMFCCs and Δchroma features, the recognition performance of the system was evaluated in a series of experiments using a database of 10000 songs. Results show that the system provides robustness against various distortions, especially when ΔMFCCs are used as features. Finally, using false acceptance and false rejection rates for various song databases, the system can learn an optimal threshold to reject queries that do not belong to the database. Future work may focus on incorporating more features that are robust to distortion in the feature vector, while retaining a relative low query speed for the system.

REFERENCES

[1] P. Cano, E. Batlle, T. Kalker and J. Haitsma, *A review of audio fingerprinting*, J. VLSI Signal Process., Vol. 41, No. 3, pp. 271-284, 2005.

[2] A. Ramalingam and S. Krishnan, *Gaussian mixture modeling of short-time Fourier transform features for audio fingerprinting,* IEEE Trans. Inf. Forensics and Security, vol. 1, no. 4, pp. 457-463, Dec. 2006.

[3] F. D. Leon and K. Martinez, *Enhancing timbre model using MFCC and its time derivatives for music similarity estimation,* Proc. EUPISCO, pp. 2005- 2009, Bucharest, Romania, Aug. 2012.

[4] D. A. Reynolds and R. C. Rose, *Robust text-independent speaker identification using Gaussian mixture speaker models,* IEEE Trans. Speech Audio Process., vol. 3, no. 1, pp. 72-83, Jan. 1995.

[5] A. Li and C. Wang, *An industrial strength audio search algorithm,* Proc. ISMIR03, pp.7-13, Baltimore, USA, 2003.

[6] D. Ellis, *Chroma feature analysis and synthesis*, http://labrosa.ee.columbia.edu/matlab/chroma-ansyn/, Columbia University, 2007.

[7] C. J. Burges, J. C. Platt and J. Goldstein, *Identifying audio clips with RARE*, Proc. ACM MM03, pp. 444-445, New York, USA, 2003.