CONTENT-BASED RETRIEVAL OF MP3 SONGS BASED ON QUERY BY SINGING

Wen-Nung Lie and Chen-Kang Su

Department of Electrical Engineering
National Chung Cheng University, Chia-Yi, 621, Taiwan, ROC.
Phone : 886-5-2720411 ext. 33211, Fax : 886-5-2720862
E-mail : wnlie@ee.ccu.edu.tw

ABSTRACT

With the growing of multimedia in Internet, content analysis of multimedia plays an important role for humanistic management. In this paper, we investigate the content-based retrieval of MP3 songs based on the interface of query by singing. In our method, the MDCT spectral coefficients were directly used to represent the tonic characteristic of a short-term sound. This spectral profile is used for detailed matching between two audio segments. Perceptual features were also computed from MDCT coefficients for audio classification. Two pre-stages based on SVM and *k*-means classifications were used to remove incorrect (or noisy) segment candidates and speed up following matching process. On the other hand, the schemes of exponential key-scaling and time-warping techniques were developed to overcome key difference and tempo variation between different singers. Experiments show that the retrieving probability of our design can achieve up to 76 % among the top 5 out of a total of 114 excerpts in the database.

I. INTRODUCTION

Traditional organization of audio excerpts using song's name, singer, and other text-based managements is not enough for modern users. In recent years, content-based audio analyses and managements are getting more attention in the area of multimedia databases. This kind of systems would help people access the interesting music conveniently. Especially, humming or singing constitutes the most convenient, but also the most challenging, way in human-machine interface.

Several researches of audio content analysis were based on PCM (pulse code modulation). The MP3 (MPEG-1 audio layer III) songs are widely popular in the World Wide Web. Therefore, we now focus on the content-based analysis of the MP3 songs, which stand in the compressed domain.

Nowadays, the research in content-based audio analysis and management can be generally put into two main categories: (1) audio segmentation and classification, and (2) audio retrieval.

Audio segmentation/classification [2] concerns about the use of features to segment and classify audios into several basic types such as speech, music, song, environmental sounds, speech with background music, and environmental sounds with background music. Jiang *et al*. [3] presented the octave-based spectral contrast feature to classify the music into Baroque, Romantic, Pop, Jazz, and Rock classes. Some other researches [4] focused on MPEG bit-stream and used features in compressed domain for classification. Some researches about audio classification focused on using the pattern classifier to discriminate audios. For examples, neural network [5], kNN classifier, and support vector machines (SVM) [6].

Methods of interface can be query by audio excerpt [6] and query by humming or singing [7-8]. Humming a tune and singing a song have now become convenient and effective interfaces for audio retrieval. Most of the recent works was based on melody and rhythm extraction. Depending on the audio form stored in the database, MIDI or polyphonic audio, different strategies were adopted. Working on the MIDI database, it was assumed that the pitch, melody, and rhythm information have been extracted [7] and stored in the symbolic format. Working on the polyphonic audio databases may suffer from the fact that phoneme segmentation and melody or rhythm extraction from polyphonic audio are difficult-to-solve problems due to the mixing of many sources of musical instruments.

Feature matching constitutes the core technology in audio retrieval, while audio classification or clustering is helpful to the speedup of the search (matching of audio segments with quite different classifications can be bypassed). Here, both the classification and matching techniques for audio segments were investigated. For example, each audio segment was classified into the types of silence, pure music, or singing-mixed music.

By the way, we focused on the retrieval of polyphonic music stored in the MP3 form. Due to these two limitations (polyphonic and compressed), we did not extract the phoneme and melody information from audio clips in the database. Instead, some perceptual features were calculated from MDCT coefficients for audio classification (based on the SVM and *k*-means classifiers). Besides, MDCT coefficients were directly used for tone representation which is then matched to those obtained from the query signals. For robust matching between the stored MP3 music and the query signals, the exponential frequency-scaling and dynamic time-warping techniques were used to overcome the problems of different key references and tempo variations for different singers.

II. AUDIO FEATURE SELECTION

A. Tone transcription vector

A collected set of MDCT coefficients were directly used as the tone transcription vector in this paper. In order to eliminate the influences of low-frequency background music (like the drum beating) and high-frequency homonymic sounds of musical instruments, only the 5^{th}~25^{th} MDCT coefficients were chosen for the query signals and the 5^{th}~50^{th} MDCT coefficients for the database music. The reason why a double amount of coefficients was selected for the database songs is the key-scaling purpose (discussed later). By the way, considering the matching speed and noise reduction, two audio frames are combined as an audio

segment and coefficients in a segment are averaged. The definition of tone transcription vector is given below:

$$HTT_n = \frac{(s_n^5, s_n^6, s_n^7, \cdots, s_n^{24}, s_n^{25})}{HM_n}, \qquad n = 1, 2, 3, \cdots, M \tag{1}$$

$$DTT_n = \frac{(s_n^5, s_n^6, s_n^7, \cdots, s_n^{49}, s_n^{50})}{DM_n}, \qquad n = 1, 2, 3, \cdots, N_D \tag{2}$$

where HTT_n is for the query signal, DTT_n is for the database music, s_n^i is the averaged i-th frequency component, n is the audio segment index, M and N_D are numbers of segments in the query and database audio excerpt, respectively, and HM_n and DM_n are normalization factors.

B. Subband energy distribution and energy variation

Subband energy can be calculated from MDCT coefficients. We define 6 subbands whose frequency ranges are 1~50, 51~100,101~150, 151~200, 201~250, and 251~300, all in terms of MDCT coefficient index. Energy of each subband is calculated by summing the squares of coefficient amplitudes, normalized by the total energy from the 1st~300-th coefficients. On the other hand, the spectrum variation of the singing-mixed music is larger than that of pure music. Energy variation of a subband can be also used as a discriminator.

C. Spectrum centroid [1]

Generally speaking, the spectrum centroid of singing-mixed music is lower than that of pure music.

D. Spectrum spread [1]

The spectrum spread of the singing-mixed music is larger than the pure music. It is defined as the square root of the spectrum magnitude-weighted average of the squared difference between the frequency and the spectrum centroid.

E. Spectrum flux [1]

The spectrum flux was defined as the average variation of spectrum between two adjacent audio segments. The variation of spectrum between adjacent segments of the singing-mixed music is more distinguishable than that of pure music.

III. K-MEANS CLASSIFICATION

We consider the problem of query by singing as searching audio segments from the database that are similar to the query signal. This is time-consuming for matching in a high-dimensional feature space. Efficient classifications will make query systems work faster and more practical.

The *k*-means clustering technique was adopted to partition the audio segments in all of the database audio clips into *c* clusters. In our system, the dimension of each *DTT* vector is 46, which made us choose *c* to be 50 for sufficient partitioning.

By *k*-means clustering, the cluster means and labels for all *DTTs* can be obtained. By the way, we computed mutual distances between cluster means and constructed a look-up table for later use. This table was used in the query process for quick reference to speed up the matching.

IV. SVM CLASSIFICATION

Generally, we can categorize a pop song into parts of singing-mixed music and pure music. If each audio segment can be first classified into these two types of sounds before matching, the search time and the matching performance can be improved significantly. In this paper, the SVM technique, which is often a two-class classifier, was used to achieve the purpose.

Perceptual features introduced in Section II.B~E were used to discriminate the singing-mixed music from pure music by SVM. A 16-dimensional feature vector was constructed for each audio segment. We used the Gaussian Radial Basis kernel function [1] to train the SVM classifier, where the variance σ is optimally set to be 0.5 by trials on current database.

V. SPECTRUM SCALING

Normally, different singers may have different reference keys in singing. That is, for the short-time spectrum, there may be a scaling of the major-peak frequency between the query and database signals. This frequency scaling could be consistent along the whole audio. Hence, key of the query signal should be up- or down-shifted to be matched with that of the database clips.

In our system, components of *HTT* were scaled in frequency axis before matching to those of *DTT*. The scaling is basically exponential, according to the property that the frequency interval between notes changes with an exponent of $2^{1/12}$. For simplicity, a step size of $2^{1/10}$ was adopted, regarding the computing speed. The range of scaling was selected to be within $2^{-5/10} \sim 2^{10/10}$. A scale larger than $2^{0/10}$ means that the key of the query signal is lower than that of the database excerpt. Our goal is to find a scale, among $2^{-5/10} \sim 2^{10/10}$, that makes the distance between the scaled *HTT* and *DTT* a minimum.

VI. AUDIO SEGMENT MATCHING

A. The overall flow of processing

In this section, several techniques, such as SVM and *k*-means classification, spectrum scaling, and time warping, are integrated for audio segment matching. Among them, the classification techniques are used to exclude impossible candidates to speed up the matching process. Spectrum scaling is for overcoming the key difference between different singers. Here, we propose a sliding-window-based time warping technique to solve the problem of tempo variation between different singers.

Figure 1 shows the processing flow of proposed audio retrieval system. At the database side, the tone transcription vector is extracted for each audio segment in each MP3 excerpt. The *k*-means clustering is then applied to the 46-D *DTT* space to construct an indexing structure (indicating the nearest cluster mean to each audio segment) for on-line use. On the other hand, the perceptual features in 16-D space are also extracted for each audio segment, which are then SVM-classified into singing-mixed music ("S") or pure music ("M"). Also, the classification label of each audio segment is stored.

For the query signal, the processing is similar. The tone transcription vectors are extracted and then scaled in the frequency axis (i.e., key-scaling). The scaled *HTT* is classified by calculating the distances to the 50 stored cluster means. Based on these classification labels, we can calculate a rough distance of this query signal to the audio clips in database.

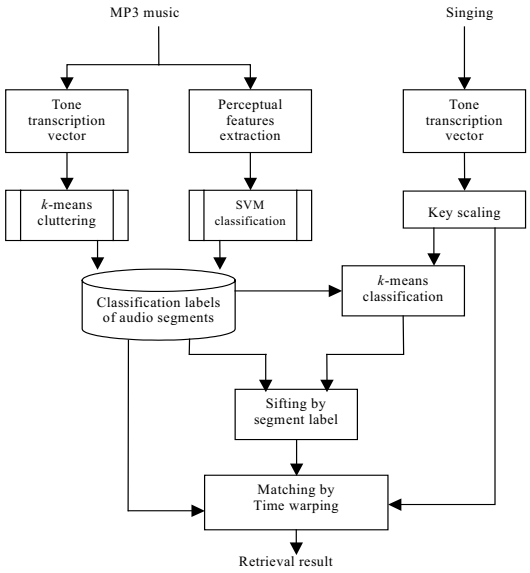

Fig. 1. Overview of proposed audio retrieval system

B. Location of singing-mixed audio portions by SVM classification

Basically, we only need to do matching between the query signal and the audio segments that are classified to be singing-mixed music. A sliding window (Fig.2), whose length is *M*, was proposed to locate audio portions that contain singing-mixed music signal. Audio portions mostly composed of "M" labels is unlikely to match the query signal. According to our experiments, the proposed SVM has a classification rate of 85.14% in separating singing-mixed music and pure music. To compensate this inaccuracy, a threshold value was used for robust recognition:

$$\sum_{i=1}^{M} Bool_i(L = "S")/M \geq Thd\,, \qquad (3)$$

where *Bool*(.) returns +1 if the classified label is "S" and 0 if else. Experimentally, *Thd* is set to be 0.5 ~ 0.85.

C. Matching based on k-means classification

Each audio segment, disregarding from the query or the database, can be classified to one of the 50 clusters. Given a scaling *s*, we would like to compare the scaled *HTT* with the *DTT*. When they are assigned to the same cluster, a similar tonic characteristic is meant. Otherwise, the distance between their cluster means indicates the degree of difference between them.

We adopted a modified manner for *k*-means classification of a scaled *HTT*: computing the distance by considering only the 21-D subspace out of the original 46-D space:

$$D'_j = \sqrt{\sum_{i=5}^{25}(q^i - \mu_j^{si})^2}\,, \qquad 1 \leq j \leq 50\,, \qquad (4)$$

where $\{q^i\}$ are the MDCT coefficients in *HTT*, $\{\mu_j^{si} | i = 5 \sim 25\}$ is the set of mapped coefficients of the *j*-th cluster means. Each scaled *HTT* is assigned with a cluster label that minimizes D'_j.

After each scaled *HTT* is labeled, we compute the distance between a scaled *HTT* and a given *DTT*:

$$D_{hk} = \sqrt{\left|\boldsymbol{\mu}_{L(h)} - \boldsymbol{\mu}_{L(k)}\right|^2}\,, \qquad (5)$$

where $\boldsymbol{\mu}_j$ represents the *j*-th mean vector, *L*(.) defines a *k*-means classification function that returns a cluster label nearest to *h* or *k*, where *k* is a *DTT* and *h* is a scaled *HTT*. The distance between a *DTT* and a scaled *HTT* is figured out by computing the distance between their associated cluster means. Summing the distances between a set of scaled *HTT*s and *DTT*s thus determines the similarity to the audio portion that is windowed. Audio portions passing the similarity check are considered for detailed matching.

D. Detailed matching – time-warping to solve tempo variation problem

Traditionally, the problem of tempo variation between different singers was solved by using time warping, or dynamic programming (DP), technique. Here, a DP scheme that takes advantages of the results from *k*-means and SVM classifications was developed to speed up the matching process.

As shown in Fig.2, each stage represents an audio segment in the query signal and each node represents one audio segment in the database. The node cost is defined to be the matching error between one *HTT* and one *DTT*. When the window slides in the database clip, the SVM labels of the *DTT*s are first checked (Eq.(3)) and then the cluster labels were compared between *HTT*s and *DTT*s (Eq.(5)). On failure of either test, the corresponding nodes (the 45° diagonals in Fig.2) are marked as "null" and the node costs are set to be infinite. Notice that not only the nodes are nullified, but also their linkages to other nodes are nullified.

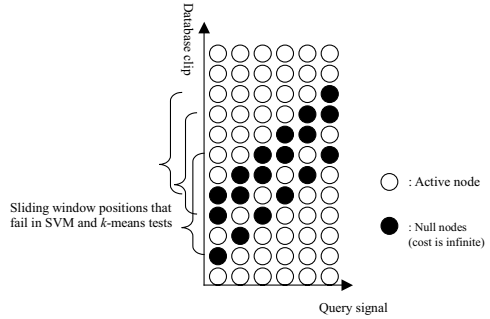

Fig. 2 The multi-stage topology of the DP matching process.

For active (not-nullified) nodes, the node cost $d_s(i, j)$ is :

$$d_s(i, j) = \sum_{n=5}^{25}(DTT_j(s \cdot n) - HTT_i(n))^2\,. \qquad (6)$$

The edge cost for the linkage between two nodes ((*i*, *j*) and (*i*-1, *k*)) is set to be 1 (*j*=*k*-1), 0 (*j*=*k*), 1 (*j*=*k*+1), 1.5 (*j*=*k*+2), 2.5 (*j*=*k*+3), and ∞ (others). That is, nodes far away could not be linked and a range of –1 ~ +3 could be allowed. Since we take the key scaling into consideration, the DP processing is performed once for a given scale. A total of 16 DP processes (16 scales) is performed for one query signal matching to one database music clip.

After applying the DP processes, an optimal path and its corresponding scaling and cost can be found out. Audio retrieval is to rank these optimal costs among MP3 excerpts in the database.

E. Greedy matching

A greedy matching method is proposed for comparison with the DP-based matching in speed and retrieving probability. The greedy matching method is also based on the sliding window process. Denote the query signal as $\{HTT_1, HTT_2, ..., HTT_M\}$ and

the windowed portion of current database clip as $\{DTT_\alpha, DTT_{\alpha+1}, ..., DTT_{\alpha+M}\}$. As the SVM and *k*-means classification succeed, search of assignments starts from HTT_1. By default, DTT_α is assigned to HTT_1. Then we try to find the assignment to HTT_2. Only *DTT*s at positions $\alpha, \alpha+1, \alpha+2$ are searched. That is, a tolerance of two audio segments is allowed in edge linking. Iterating this searching process for HTT_{i+1} ,…, HTT_M will find all assignments to $\{HTT_1, HTT_2, ..., HTT_M\}$. The total cost of the passed nodes thus forms the matching cost of the audio portion starting from DTT_α.

VII. EXPERIMENTAL RESULTS

We built a database of MP3 audio excerpts. The database is composed of 114 Taiwan-pop songs. Each song is about 60 seconds long. Fifty queries by singing were conducted. Each singing is about 5~10 seconds long. The query signal could be in any portion of an audio. We assumed a unified scaling for all the audio segments in the query signal.

Table I shows the retrieving probability (see also Fig.3) and average searching time with different combinations of methods. "Top-*i*" means that the correct answer can be found from the top *i* retrieved music clips. The methods include : 1) *Scheme* 1: DP matching without key-scaling, 2) *Scheme* 2: DP matching with exponential key-scaling, 3) *Scheme* 3: DP matching with exponential key-scaling and *k*-means classification, 4) *Scheme* 4: DP matching with exponential key-scaling, SVM and *k*-means classification, 5) *Scheme* 5: Greedy matching with exponential key-scaling, SVM and *k*-means classification.

Scheme 1 has the worst performance. Scheme 2 is better, but with an increased searching time. Scheme 3 has the same performance as scheme 2, but the searching time is reduced (due to the power of *k*-means classification). However, scheme 4 is better than scheme 3. The search time is also decreased. Thanks to the SVM classifier. Finally, scheme 5 (greedy matching) is slightly worse than scheme 4, but is the speediest (except scheme 1). Since the greedy search is not optimal, performance degradation is expectable. However, it is still competitive to schemes 2 & 3.

VIII. CONCLUSIONS

We have worked on content-based retrieval of MP3 pop songs in a manner of query by singing. In the proposed system, two sifting stages (SVM and *k*-means classification) were adopted to speed up the matching process and simultaneously improving the retrieving probability. The schemes of exponential key-scaling and time-warping techniques were developed to overcome the differences in reference keys and tempo variation between different singers.

In our method, all the perceptual features (for classification) and tone transcription vectors (for detailed DP matching) are all extracted from the MDCT coefficients.

By experiments, the retrieving probability can achieve up to 76 % among the top 5 matched, out of a total of 114 excerpts in the database. This is comparable to other literature, though it is now difficult to make a fair comparison, due to the lack of a common database.

Table I. The retrieving accuracy and average searching time with different schemes

%	Top-1	Top-3	Top-5	Top-10	Top-15	Top-20	Time (sec)
Scheme 1	10	18	22	30	42	50	2.98
Scheme 2	42	64	74	76	80	84	31.28
Scheme 3	42	64	74	76	80	84	12.03
Scheme 4	48	72	76	80	88	92	8.29
Scheme 5	48	66	70	82	88	90	6.48

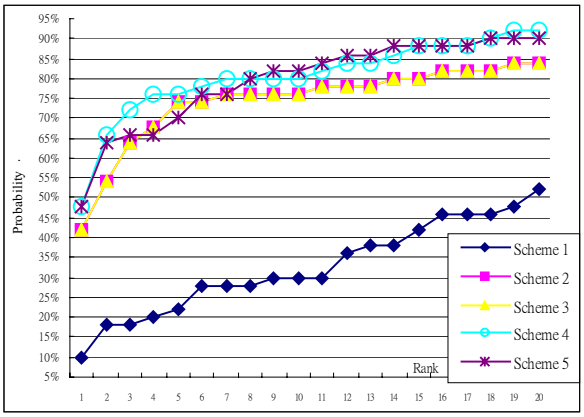

Fig.3 Retrieving probability with different selections of top ranked MP3 excerpts.

REFERENCES

[1] Lie Lu, Hong-Jiang Zhang, and Stan Z. Li, "Content-Based Audio Classification and Segmentation by Using Support Vector machines," *ACM Multimedia Systems Journal* 8 (6), pp. 482-492, 2003.

[2] Tong Zhang and C.-C. Jay Kuo, "Audio Content Analysis for Online Audiovisual Data Segmentation and Classification," *IEEE Trans. on Speech and Audio Processing,* Vol. 9, No. 4, pp. 441-457, May 2001.

[3] Dan-Ning Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai, "Music Type Classification by Spectral Contrast Feature," *Proc. of IEEE Int'l Conf. on Multimedia and Expo (ICME02),* pp.113-116, 2002.

[4] Roman Jarina, Noel Murphy, Noel O'connor, and Sean Marlow, "Speech-Music Discrimination from MPEG-1 Bitstream," *Advances in signal processing, robotics and communications*, pp. 174-178, 2001.

[5] M. De Santo, D. Percannella, C. Sansone, and M. Vento, "A Neural Multi-Expert Classification System for MPEG Audio Segmentation," *Proc. of Int'l Conf. on Advances in Pattern Recognition,* Brazil, pp. 50-59, March 2001.

[6] Guodong Guo and Stan Z. Li, "Content-Based Audio Classification and Retrieval by Support Vector Machines," *IEEE Trans. on Neural Networks,* Vol. 14, No. 1, pp.209-215, 2003.

[7] Lie Lu, Hong, and Hong-Jiang Zhang, "A New Approach To Query By Humming in Music Retrieval," *Proc. of IEEE Int'l Conf. on Multimedia and Expo,* pp. 776-779. 2001.

[8] Jyh-Shing Roger Jang and Ming-Yang Gao, "A Query-by-Singing System Based on Dynamic Programming," *Proc. Of Int'l Workshop on Intelligent Systems Resolutions (the 8th Bellman Continuum),* Taiwan, pp. 85-89, Dec 2000.