A Singer Identification Technique for Content-Based Classification of MP3 Music Objects

Chih-Chin Liu and Chuan-Sung Huang
Department of Computer Science and
Information Engineering
Chung Hua University
Hsinchu, Taiwan 300, R.O.C.
ccliu@chu.edu.tw

ABSTRACT

As there is a growing amount of MP3 music data available on the Internet today, the problems related to music classification and content-based music retrieval are getting more attention recently. In this paper, we propose an approach to automatically classify MP3 music objects according to their singers. First, the coefficients extracted from the output of the polyphase filters are used to compute the MP3 features for segmentation. Based on these features, an MP3 music object can be decomposed into a sequence of notes (or *phonemes*). Then for each MP3 phoneme in the training set, its MP3 feature is extracted and used to train an MP3 classifier which can identify the singer of an unknown MP3 music object. Experiments are performed and analyzed to show the effectiveness of the proposed method.

Categories and Subject Descriptors

H.3.1 [**Information Systems**]: Information Storage and Retrieval – *retrieval model, search process, selection process.*

General Terms

Algorithms, Measurement, Documentation, Performance, Design, Experimentation.

Keywords

MP3, music classification, music databases, MP3 classification, MP3 databases, singer identification, content-based music classification, music feature extraction.

1. INTRODUCTION

As the explosive growth of the Internet, the techniques for content-based retrieval of multimedia data are getting more attention in the area of the multimedia database. MP3 (MPEG 1 Audio Layer 3) is an ISO international standard for the compression of digital audio data [12]. Since MP3 can compress audio signals at a data reduction of 1:12 while remaining CD sound quality, it has become the dominant format for audio data on the Internet. Thus, the techniques for content-based retrieval, segmentation, and classification of MP3 audio data are highly demanded.

CIKM'02, November 4-9, 2002, McLean, Virginia, USA.

Previous works on content-based retrieval of music data are reviewed in the following. Smith *et al.* proposed a method to search through broadcast audio data to detect the occurrences of a known sound [36]. Melih *et al.* [26] proposed a method for browsing and retrieving audio data. In their method, the audio signal is divided into many tracks. For each track, its start time, end time, frame number, and classification information are kept. These structural audio information can be used to support content-based retrieval. The most straightforward way to query the music databases for a naive user is to hum a piece of music as the query example to retrieve similar music objects. This approach is adopted in [1][4][6][10][13][18][33][37].

Audio signals can be roughly categorized into speech, music, and sound effects. Speaker recognition is a major research topic in the area of speech processing and a large number of speaker recognition techniques were proposed. Tutorials on speaker recognition can be found in [3][5]. Many works were also done on the discrimination between speech and music signals. Saunders proposed a technique for discriminating speech from music on broadcast FM radio [34]. Instead of the common acoustical features (e.g. tonality, bandwidth, pitch, tonal duration, and energy sequence), to reduce computational overhead, the distribution of the zero-crossing rate of the time domain waveform is used in the proposed discriminating algorithm. Other speech/music discriminating methods can be found in [7][22][23][24][27][29][35].

Previous works on sound effect classification can be found in [16][21][40][41]. Wold *et al.* proposed an approach to retrieve and classify sounds based on their content [40]. In this approach, an N-vector is constructed according to the acoustical features of an audio object. These acoustical features include *loudness*, *pitch*, *brightness*, *bandwidth*, and *harmonicity*, which can be automatically computed from the raw data. The Euclidean distance between N-vectors is adopted as the similarity measure for classification. Liu *et al.* proposed an approach to classify video scenes using their associated audio data [21]. The audio features such as volume distribution, pitch contour, frequency bandwidth, and energy ratios, are used to classify audio clips of TV programs into news reports, weather reports, advertisement, basketball game, and football game. Li used perceptual features (such as total power, brightness, bandwidth and pitch) and cepstral features (MFCCs) to classify sounds [16]. A pattern recognition method called NFL (the nearest feature line) is applied and compared with the near neighbor method [9]. According to their experiments, the error rate is reduced from 18.34% to 9.78%. Zhang and Kuo proposed a hierarchical audio classification method [41]. In this approach, audio signals are classified in two stages. In the first stage, audio signals are classified into speech, music,

environmental audio and silence. Then based on the timbre and rhythm features, the fine-level audio classification is provided by construction a hidden Markov model for each class of sounds.

As there are increasing amount of music data available on the Internet, music classification is getting more attention recently. Lambrou *et al.* proposed a music classification technique [15]. Eight acoustical features, i.e., mean, variance, skewness, kurtosis, angular second moment, correlation, number of zero crossings, and entropy are extracted from the music signals from the time and wavelet transform domains. Then, four statistical classifiers are applied to classify these music signals into three *music styles*, rock, piano, and jazz. According to their experiments, the features extracted from the adaptive splitting wavelet transform will perform better than other features. Martin developed an audio recognition system that can identify sounds generated by musical instruments [25]. The taxonomy of orchestral instrument sounds is trained based on many acoustic properties. For a given music tone, statistical classification techniques are applied to find which instrument family the tone likely belongs to. Tzanetakis and Cook proposed an audio analysis framework [39]. Many acoustic features such as pitch, harmonicity, MFCC, LPC, spectral centroid, and spectral flux are extracted from the audio signals. Then, a music/speech discriminator and an instrument identification system can be constructed based on two statistical classifiers. More acoustic features are considered in their recently work [38]. These acoustic features are extracted to represent the timbre, rhythm, and pitch characteristics of music signals. Based on these features, audio signals can be classified into *musical genres* such as classical, country, disco, hiphop, jazz, rock, blues, reggae, pop, and metal. This approach seems very promised to automatically organize the music objects on the Internet.

The unit for retrieving and classifying multimedia data should be carefully chosen. For video data, a shot is the basic unit for video indexing. Therefore, many shot-change detecting methods were developed. Similarly, a music object should be decomposed into a sequence of music sentences (or *phrases*) for indexing. Thus, an automatic phrase segmentation mechanism is strongly required. Melih and Gonzalez proposed an audio segmentation technique based on a perceptually derived audio representation [27]. By analyzing the properties of audio tracks in the time-frequency domain, five track types (tone, sweep, formant, noise, and unknown) can be identified. However, this approach is not suitable for music data since the whole music track is of the same type. In [19], we propose a technique to automatically segment MP3 music objects for supporting content-based retrieval of MP3 music data. In this method, an MP3 music object is segmented in three steps. First, the coefficients extracting from the output of the polyphase filters are used to compute the MP3 features. Then an MP3 music object is decomposed into a sequence of notes (or *phonemes*) based on these features. Finally, heuristic rules are applied to group these notes into music phrases.

The techniques for audio data retrieval mentioned above are developed for audio data in waveform or MIDI format. As the explosive growth of the Internet, there is a growing amount of MP3 music data available on it. However, currently only keyword-based searching mechanisms are provided by the MP3 content providers (e.g. http://www.mp3.com/) and MP3 searching engines (e.g. http://launch.yahoo.com/ and http://mp3.lycos.com/). To provide users more powerful and friendly interface to find their desired music data, two techniques are addressed. First, we have proposed an approach to retrieve MP3 music object by humming a piece of music as the query example [20][37]. The other approach is to organize the music data on the Internet into categories such that a Yahoo-like music directory can be constructed for the users to browse it. In this paper, an automatically MP3 music classification technique based on their singers is proposed. In this approach, MP3 music objects are classified in two stages. In the first stage, the coefficients extracting from the output of the polyphase filters are used to compute the MP3 features for segmentation. Based on these features, an MP3 music object can be segmented into a sequence of notes (or *phonemes*). For each MP3 phoneme in the training set, its MP3 feature vector is extracted and stored with its associated singer name in an MP3 phoneme database. On the second stage, the phonemes in the MP3 phoneme database are used as discriminators in an MP3 classifier to identify the singers of unknown MP3 music objects.

The rest of this paper is organized as follows. Section 2 presents the types of music categories and the architecture of an MP3 classification system. Section 3 describes how to segment MP3 music objects into MP3 phonemes and the phoneme MP3 feature extracting technique is addressed. The MP3 classification techniques are discussed in section 4. In Section 5, experiments are performed to show the effectiveness of the approach. Finally, Section 6 concludes this paper and presents our future research directions.

2. ARCHITECTURE OF AN MP3 MUSIC CLASSIFICATION SYSTEM

2.1 Types of Music Categories

Traditionally, music classification occurs in the CD stores when arranging the CDs for the customers to quickly find their desired CDs. Now this technique is required for automatically organizing music content on the Internet. Music objects can be classified in many ways. The most common classification methods for MP3 songs are by music styles and by artists. These methods are adopted by the MP3 search engines (e.g. http://www.yahoo.com/) and the MP3 content providers (e.g. http://www.mp3.com/).

- Music Style: MP3 songs can be classified according to their *music styles* or *music genres*. For example, MP3.com classifies MP3 songs into 16 music genres, i.e., Alternative, Blues, Children's Music, Classical, Comedy, Country, Easy Listening, Electronic, Hip Hop/Rap, Jazz, Latin, Metal, Pop & Rock, Reality, Urban/R&B, and World/Folk.
- Artist: The most straightforward way to classify songs for naïve users is according to their singers. People who love a song tend to accept the songs sung by the same singer. Since every great singer has his/her own style, we can also think an artist to be a minimal music genre. Almost all music web sites provide artist categories sorted from A to Z.

For example, Yahoo uses 12 music genres while MP3.com uses 16 music genres, which means genre classification is somewhat subjective. Yahoo also provides artist-based music directories. These two kinds of music classification methods can be mixed. For example, as Figure 1 shows, MP3.com provides two-layered music classification: In the first layer, MP3 songs are classified by music genres. For the MP3 songs in each music genre, they are further sorted according to their artists.

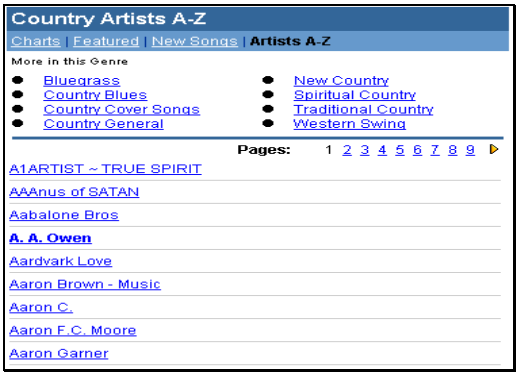

Figure 1. Two layer music classification provided by MP3.com.

2.2 System Architecture

Figure 2 shows the architecture of an MP3 classification system. The classification is performed in two stages. In the first stage, an MP3 phoneme database is constructed by segmenting a set of known MP3 training songs into phonemes and extracting their MP3 phoneme features. Every phoneme in the phoneme database will act as a discriminator to classify unknown MP3 songs. Then, another set of MP3 music objects whose singers are known is used to train the MP3 classifier for tuning the weights of discriminators. In the second stage, for every MP3 music object found on the Internet, it is segmented and its MP3 features are extracted from the MP3 bitstream. Then, the trained MP3 classifier will apply to identify the singers of these unknown MP3 music objects and categorize them into music directories.

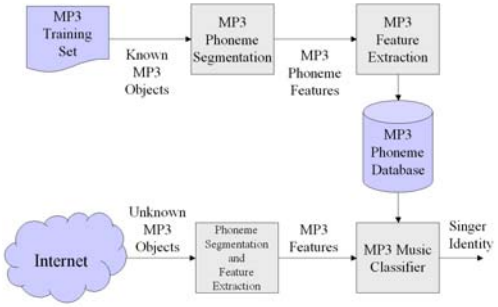

Figure 2. A genetic MP3 classification system.

3. THE CONSTRUCTION OF A PHONEME DATABASE

In this section, we describe the procedure for constructing an MP3 phoneme database.

3.1 MP3 Decoding Procedure

For content-based multimedia retrieval and classification systems, the features used should be extracted from the compressed raw data. So the MP3 decoding procedure is reviewed in short. Figure 3 shows the block diagram of an MP3 decoder. The basic unit of an MP3 bitstream is a *frame*. An MP3 frame consists of 1152 samples. A typical MP3 music object is sampled at 44.1 kHz. Therefore, there are 44100/1152 = 38.28125 frames per second. According to the standard [12], an MP3 bitstream is unpacked and dequntized, frame by frame, into *MDCT* (modified discrete cosine transform) *coefficients*. The MDCT coefficients (576 frequency lines) are then mapped into subsamples (32 subbands) using inverse MDCT. These subsamples also called *the polyphase filter coefficients*. Finally, the subsamples are synthesized into the original audio signal (PCM audio). Both the MDCT coefficients and the polyphase filter coefficients can be used to compute the MP3 features. For more information about MP3 decoding please refer to Pan [32], Brandenburg and Stoll [2], and Noll [30].

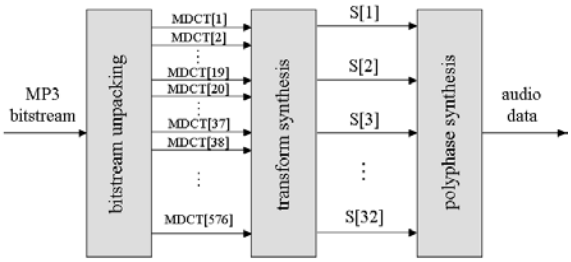

Figure 3. MP3 decoder diagram (modified from [12]).

3.2 Phoneme Segmentation

3.2.1 The Properties of Phonemes

During the decoding process, the transform synthesis module of the MP3 decoder will transform 576 MDCT coefficients into 32 subsamples (32 subbands) using inverse MDCT. Since the square of the coefficient of each subband represents the energy intensity of audio signal in this subband, for each subband, we can add the square of every subband coefficient within an MP3 frame. Let *S[i][j]* represents the *j*-th subsample at the *i*-th subband, we define

$$PC_i = \sum_{j=1}^{36} (S[i][j])^2 \qquad (1)$$

To segment an MP3 music object, the features should reflect the energy changes among frames. For example, Figure 4 shows the waveform of the song "It is sunny" by Zu-Yung Hsu. There are obvious energy changes between notes. Therefore, we can define the *frame energy* (*FE*) of a frame as the summation of its 32 subband energy PC_i.

$$FE = \sum_{i=1}^{32} PC_i \qquad (2)$$

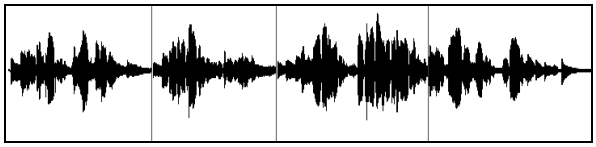

Figure 4. The waveform of the music example "It is sunny".

An MP3 phoneme represents a note in the staff or a syllable in a music sentence. It is typically generated with quasi-periodic excitation of resonance structures by singers and orchestral musical instruments. Figure 5 shows the waveform of four phonemes. We can find there is an energy gap (also called *MP3 phoneme break*) between two phonemes. Thus, the goal of the MP3 phoneme segmentation is to develop a technique to automatically identify the locations of these energy gaps from the raw MP3 bitstreams.

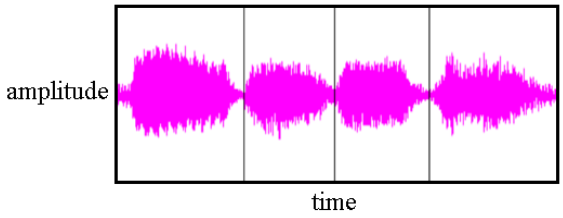

Figure 5. The waveform of four phonemes.

The envelope of the waveform of a phoneme has a four-stage structure, i.e., attack, decay, sustain, and release stages. This is known as the ADSR parameters set that is used in synthesizers to control the start, duration and ending of a note. For example, Figure 6 shows the waveform of two phonemes extracted from the pop song "It is sunny". Their envelopes are denoted with dotted lines. The ADSR stages can be easily observed. The energy gap is located between the release stage of a phoneme and the attack stage of its adjacent phoneme.

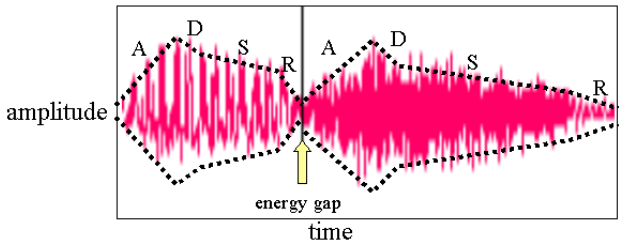

Figure 6. The waveform of two phonemes.

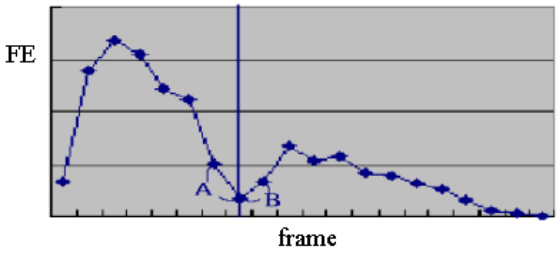

Figure 7. The FEs of two MP3 phonemes.

We encode the piece of the song shown in Figure 6 into MP3 format and extract its MP3 features. The result is shown in Figure 7. Two properties can be observed. First, the energy gap is located between segment A and segment B, which is consistent with the result in Figure 6. Thus, segment A is part of the release stage of the former phoneme while segment B is part of the attack stage of the latter phoneme. Second, the FEs of the frames which are close to the energy gap are lower than the average FE of all frames. Based on these properties, an MP3 phoneme segmentation technique can be developed. This technique is formally presented in the following.

3.2.2 The Technique for MP3 Phoneme Segmentation

Rule 1 *Assume frame A, frame B, and frame C are three continuous frames of an MP3 music object whose MP3 features are FE_A, FE_B, and FE_C, respectively. If an MP3 phoneme break is located at frame B, we have*

$$\begin{cases} FE_B \le FE_A \\ FE_B < FE_C \end{cases} \tag{3}$$

$$FE_B \le FE_C \tag{4}$$

This is because the energy of frame B should be lower than or equal to that of frame A if frame A and frame B are parts of the release stage of a phoneme. Similarly, the energy of frame B should higher than that of frame C if frame B and frame C are parts of the attack stage of a phoneme.

However, not all MP3 phonemes possess standard ADSR envelopes. Noises may disturb this property. To reduce the noise disturbance, we use the summations of the FEs of n continuous frames instead of the original FEs as the new MP3 features for Rule 1.

$$FE' = \sum_{i=1}^{n} FE_i \tag{5}$$

where FE_i represents the MP3 feature of the i-th frame, and FE' is its corresponding new MP3 feature. According to the experiments, we find $n = 4$ will get the best result. Since the phoneme break should occur at lower energy area, to increase the precision of the phoneme segmentation, we can apply the following rule.

Rule 2 *Assume frame A, frame B, and frame C are three continuous frames of an MP3 music object whose MP3 features are FE_A, FE_B, and FE_C, respectively. The average frame energy of an MP3 music object is FE_{avg}. If an MP3 phoneme break is located at frame B, we have*

$$k \cdot Max(FE_A, FE_B, FE_C) < FE_{avg} \tag{6}$$

where $Max(FE_A, FE_B, FE_C)$ represents the maximal frame energy among FE_A, FE_B, and FE_C; k is a constant between 0 and 1.

3.3 MP3 Phoneme Feature Extraction

As we have mentioned, both the MDCT coefficients and the polyphase filter coefficients can be used to compute the MP3 features. In [20][37], we measure the similarity between a hummed music query and an MP3 music object based on FPCV. However, since the bandwidth of a subband is too large to discriminate MP3 phoneme, in this paper, the MP3 features of MP3 phonemes are extracted from the MDCT coefficients instead. Let *MDCT[i]* represents the coefficient at the i-th frequency line of an MP3 frame, we define

$$MC_i = (MDCT[i])^2, \; i = 1, 2, \ldots, 576 \tag{7}$$

$$FMCV = (MC_1, MC_2, \cdots, MC_{576}) \tag{8}$$

A 576-dimenstional feature vector *FMCV* (MDCT Coefficient Vector) is derived for each MP3 frame. Since an MP3 phoneme consists of a sequence of MP3 frames, we can define the phoneme feature vector *PMCV* (Phoneme MDCT Coefficient Vector) for an MP3 phoneme of n frames as

$$PMCV = (\sum_{i=1}^{n} MC_1, \sum_{i=1}^{n} MC_2, \cdots, \sum_{i=1}^{n} MC_{576}) \tag{9}$$

Perceptual normalization technique [19] can be further applied to make the feature vector more consistent with human auditory system.

4. THE MP3 CLASSIFICATION TECHNIQUE

4.1 Phoneme Discriminator Training

As we have discussed in previous section, the first stage of the proposed MP3 classification approach is to construct an MP3 phoneme database for the training set of MP3 songs whose singers are known. The number of different phonemes that a singer can sing is limited and the singers with different timbre possess their own unique phoneme sets. Therefore, the phonemes of an unknown MP3 song can be associated with the similar phonemes of the same singer in the phoneme database. Thus in the second stage of the classification procedure, each phoneme in the MP3 phoneme database is used as a *discriminator* to identify the singers of unknown MP3 songs.

Since the discriminating ability (or uniqueness) of every phoneme is unequal, to measure the uniqueness of a phoneme for a certain singer, we define the *discriminating radius* γ_f of a phoneme f to be the minimal distance between the phoneme feature vector of f and the closest phoneme feature vector corresponding to a phoneme f' sung by another singer. That is,

$$\gamma_f = Min\left(dist\left(PMCV_f, PMCV_{f'}\right)\right) \quad (10)$$

where $PMCV_f$ and $PMCF_{f'}$ are the phoneme MDCT coefficient vectors of f and f', respectively, and $dist(PMCV_f, PMCF_{f'})$ is the Euclidean distance between $PMCV_f$ and $PMCF_{f'}$.

For example, Figure 8 illustrates 18 phoneme features of three singers in the feature space (only two dimensions are shown). The discriminating radius of f_1 is much larger than that of f_2, which means the timbre of phoneme f_1 is very unique and f_1 has higher discriminating ability. Therefore we say f_1 is a *good* discriminator since it can discriminate three phonemes while f_2 is a *bad* discriminator since it can discriminate none in this example.

The other factor that influences the discriminating ability of a phoneme is the number of neighbor phonemes of the same singer. If a phoneme has more neighbors, it means this phoneme really be a distinguishing characteristic of the singer. Thus, we can define the *frequency* ω_f of a phoneme f to be the number of phonemes of the same singer located within its discriminating radius.

For example, Figure 9 shows two discriminators f_3 and f_4 with the same discriminating radius. However, the frequency of f_3 is six while that of f_4 is only one, which means the discriminating ability of f_3 is stronger than f_4.

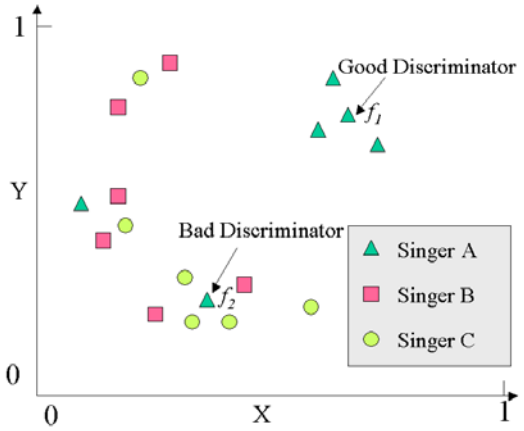

Figure 8. Good and bad discriminators.

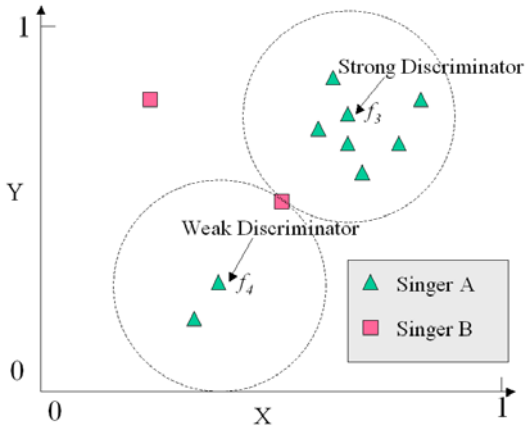

Figure 9. Strong and weak discriminators.

The discriminating radius of every discriminator in the phoneme database can be set by computing the distance between its phoneme feature vector and the feature vector of its nearest neighbor phoneme sung by different singer. Note that the nearest neighbor phoneme of a phoneme is not necessary to be the same note or syllable as long as it is sung by a different singer.

To find the frequency of every discriminator in the phoneme database, another training set is used. The frequency of each discriminator is initialized to be zero. For each phoneme in the second training set, it is compared with every discriminator of the same singer in the phoneme database. If the distance between the two phoneme feature vectors is within the discriminating radius, the frequency of the corresponding discriminator is increased by one.

After the discriminating radius γ_f and the frequency ω_f of every discriminator f in the phoneme database are computed, its discriminating function $D(f)$ can be defined as

$$D(f) = \gamma_f \cdot \log_2(\omega_f + 1) \quad (11)$$

4.2 MP3 Music Classification

The *k*NN classifier [9] is used to classify the unknown MP3 songs. For each unknown MP3 song, it is first segmented into a sequence of phonemes. For performance consideration, only the first N phonemes of an unknown MP3 song are used for classification. For each phoneme in the first N phonemes, it is compared with every discriminator in the phoneme database and the k closest neighbors are found. For each of the k closest neighbors, if its distance is within the threshold, it will give a weighted vote whose value is given by the discriminating function. Otherwise, its vote is set to zero. The $k*N$ weighted votes are accumulated according to the singers and the unknown MP3 song is assigned to the singer with the largest score.

5. EXPERIMENTS

To show the effectiveness of the proposed method, a series of experiments are performed and analyzed.

5.1 Experiment Set-up

Ten male Chinese singers (S.K. Wu, H.J. Zhou, Z.X. Lin, Y. Zhang, Y.S. Zhang, X.Y. Zhang, H.M. You, T.P. Xiong, Q. Qi, and D.H. Liu) and ten female Chinese singers (F. Wang, X.Q. Xiong, Y.Z. Sun, H.M. Zhang, Y.Q. Liang, W.W. Mo, R.Y. Xu, F.

Wan, Y.H. Zhao, and R.Y. Liu) were chosen. For each singer 30 songs were randomly picked: 10 songs were used for constructing the phoneme database, another 10 songs were used for training, and the rest 10 songs were used as the test set for classification. Therefore, there are totally 600 MP3 songs used to perform the following experiments.

The effectiveness of the MP3 music classification technique is measured by *the precision rate* which is defined as the number of MP3 songs correctly classified to a music class divided by the total number of MP3 songs in the music class. In general, a music class contains a single singer. However, we can also group several singers with similar voices (timbre or phonemes) into a class. This will provide user the ability to find songs which are similar to their favorite singers.

5.2 Experiment Results

There are three factors dominating the results of the MP3 music classification method: the setting of k in the kNN classifier, the threshold for vote decision used by the discriminators, and the number of singers allowed in a music class.

The goal of the first experiment is to find the best setting of k in the kNN classifier (i.e., k nearest neighbors). Table 1 shows the precision rates for various k values and k=100 will give the better result. To find more accurate k values, more precise settings are tested as Figure 10 shown and the best setting of k is 80, which will result in 90% precision rate (threshold=0.2, number of singers in a class is 5).

Table 1. Precision vs. Threshold.

k	1	50	100	500	1000	10000
Precision	10%	60%	80%	50%	20%	10%

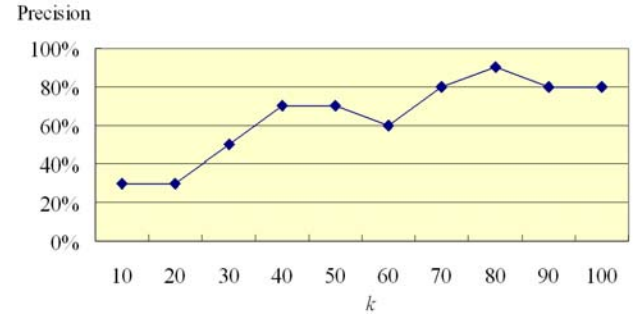

Figure 10. Precision for various k values used in the kNN classifier.

The second experiment illustrates the effect of the threshold setting. The threshold is used by the k nearest discriminators to decide whether to vote in the kNN classifier. As Figure 11 shows, if the threshold is set to 0.01, no discriminator will vote. The classification is performed as randomly, in which only 1/20 = 5% of the test MP3 songs are correctly classified. On the other hand, if the threshold is set to 1, each phoneme in the test set will let k nearest discriminators vote even though this phoneme is not similar to any discriminator. According to our experiment, we found the best threshold setting is 0.2 (k=80, number of singers in a class is 2).

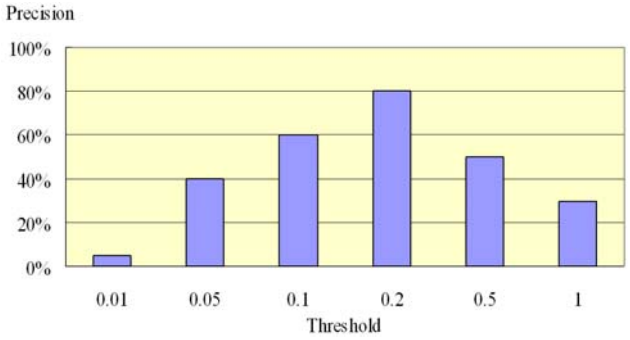

Figure 11. Precision vs. Threshold.

As we have mentioned, more than one singer can be allowed in a music class. The larger the number of singers in a music class is allowed, the higher the precision rate will be achieved as shown in Figure 12. We also find the average precision rate of male singers is a little higher (9%) than that of female singers.

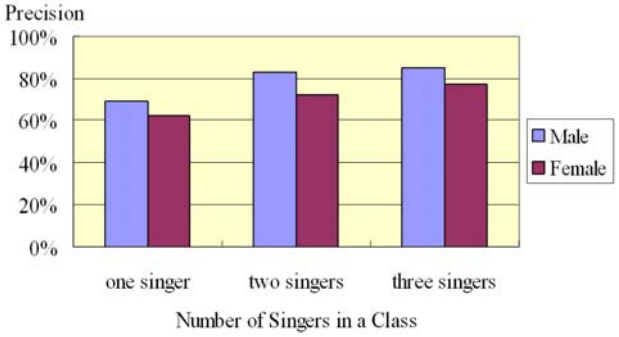

Figure 12. Precision vs. Threshold.

The experiment result for each singer is examined in more detail. As Figure 13 shows, songs sung by H. M. You result in the highest precision rate. It is consistent with our empirical conclusion: H. M. You's songs are very unique. On the other hand, only 50% of W.W. Mo's songs can be correctly classified since her voice is very common.

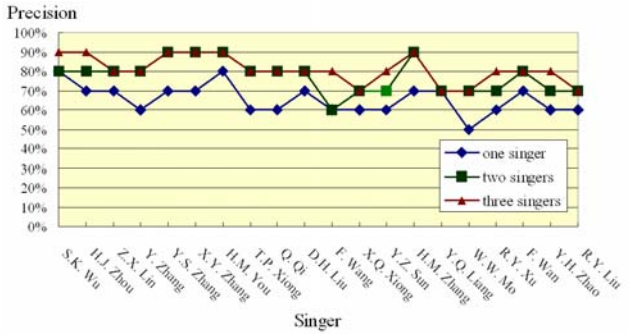

Figure 13. Precision vs. Threshold.

To investigate which singers have similar voices, the confusion matrices for the male and female singers are illustrated in Table 2 and Table 3, respectively. According to our experiments, none of the 100 songs by male singers will be classified into classes of female singers, and so for female singers. For male singers, two T.P. Xiong's songs are misclassified as Q. Qi's and three Q. Qi's songs are misclassified as T.P. Xiong's. We also found H.J. Zhou

and Y.S. Zhang possess similar timbre. These results agree with our empirical experience.

Table 2. The confusion matrix for ten male singers.

	S.K. Wu	H.J. Zhou	Z.X. Lin	Y. Zhang	Y.S. Zhang	X.Y. Zhang	H.M. You	T.P. Xiong	Q. Qi	D.H. Liu
S.K. Wu	8	0	0	1	0	0	0	0	1	0
H.J. Zhou	0	7	0	2	2	0	0	0	0	1
Z.X. Lin	1	0	7	0	0	1	1	1	0	0
Y. Zhang	0	0	0	6	0	0	0	0	0	0
Y.S. Zhang	0	3	1	0	7	0	1	1	0	0
X.Y. Zhang	0	0	0	0	0	7	0	0	0	0
H.M. You	0	0	0	1	0	2	8	0	0	2
T.P. Xiong	0	0	1	0	0	0	0	6	3	0
Q. Qi	0	0	1	0	0	0	0	2	6	0
D.H. Liu	1	0	0	0	1	0	0	0	0	7

Table 3. The confusion matrix for ten female singers.

	F. Wang	X.Q. Xiong	Y.Z. Sun	H.M. Zhang	Y.Q. Liang	W.W. Mo	R.Y. Xu	F. Wan	Y.H. Zhao	R.Y. Liu
F. Wang	6	2	1	0	1	2	3	1	1	2
X.Q. Xiong	0	6	1	0	0	1	0	1	0	1
Y.Z. Sun	0	0	6	1	0	0	0	0	2	0
H.M. Zhang	0	0	1	7	0	1	0	0	0	1
Y.Q. Liang	1	2	1	0	7	1	0	0	0	0
W.W. Mo	0	0	0	0	0	5	0	0	0	0
R.Y. Xu	3	0	0	0	0	0	7	0	0	0
F. Wan	0	0	0	0	0	0	0	7	0	0
Y.H. Zhao	0	0	0	0	0	0	0	0	6	0
R.Y. Liu	0	0	0	2	2	0	0	1	0	6

6. CONCLUSION

Today, large amounts of MP3 music data are available on the Internet. However, traditional keyword-based searching techniques are not enough to provide user to find out the desired MP3 music. Thus issues about content-based music data retrieval and classification are getting more attention recently. In this paper, a phoneme-based MP3 classification technique is proposed to automatically organize the music objects on the Internet into categories. Thus a Yahoo-like music directory can be constructed in which MP3 songs are categorized by their singers. Users can browse this music directory to find out their desired songs or the songs similar to their favorite singers.

For the future work, three research issues are addressed. First, more music features such as pitch, melody, rhythm, and harmonicity will be considered for music classification. Second, we will try to represent the MP3 features according to the syntax and semantics defined in the MPEG7 standard and investigate how to process MP3 queries in the Internet environment. Finally, since the illegal distributions of MP3 music objects can be found elsewhere, we plan to develop an *MP3 identifier* to automatically authenticate the copyright of MP3 music objects on the Internet.

7. ACKNOWLEDGMENTS

This work was partially supported by the Republic of China National Science Council under Contract No. NSC-91-2213-E-216-003.

8. REFERENCES

[1] Bakhmutova, V., V. D. Gusev, and T. N. Titkova, "The Search for Adaptations in Song Melodies," *Computer Music Journal*, Vol. 21, No. 1, pp. 58-67, Spring 1997.

[2] Brandenburg, K., and G. Stoll, "ISO-MPEG-1 Audio: A Generic Standard for Coding of High Quality Digital Audio," *Journal of the Audio Engineering Society*, Vol. 42, No. 10, Oct 1994, pp. 780-792.

[3] Campbell, J.P., Jr., "Speaker Recognition: a Tutorial**,"** *Proceedings of the IEEE*, Vol. 85, No. 9, Sept. 1997 pp. 1437 –1462.

[4] Chen, J. C. C. and A. L. P. Chen, "Query by Rhythm: An Approach for Song Retrieval in Music Databases," *In Proc. of 8th Intl. Workshop on Research Issues in Data Engineering*, pp. 139~146, 1998.

[5] Chibelushi, C.C., F. Deravi, and J. S. D. Mason, "A Review of Speech-Based Bimodal Recognition," *IEEE Trans. On Multimedia*, Vol. 4, No. 1, pp. 23-37, March 2002.

[6] Chou, T. C., A. L. P. Chen, and C. C. Liu, "Music Databases: Indexing Techniques and Implementation," *in Proc. IEEE Intl. Workshop on Multimedia Data Base Management Systems*, 1996.

[7] Chou, W., and L. Gu, "Robust Singing Detection in Speech/Music Discriminator Design," in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 2, pp. 865-868, 2001.

[8] Foote, J., "Content-Based Retrieval of Music and Audio", *Multimedia Storage and Archiving systems II, Proc. SPIE, Vol.3229,* pp. 138-147.

[9] Fukunaga, K., *An Introduction to Statistical Pattern Recognition*, San Diego, CA, Academic Press, 2nd ed., 1990.

[10] Ghias, A., Logan, H., Chamberlin, D., and Smith, B. C., "Query by Humming: Musical Information Retrieval in an Audio Database," *in Proc. of Third ACM International Conference on Multimedia*, pp. 231~236, 1995.

[11] Hsu, J.L., C.C. Liu and A.L.P. Chen, "Discovering Non-Trivial Repeating Patterns in Music Data," *IEEE Transactions on Multimedia*, Vol. 3, No. 3, pp. 311-325, 2001.

[12] ISO/IEC 11172-3:1993, "Information Technology — Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s — Part 3: Audio."

[13] Kosugi, N., Y. Nishihara, S. Kon'ya, M. Yamamuro, and K. Kushima, "Music Retrieval by Humming," *in Proceedings of IEEE PACRIM'99*, pp. 404-407, 1999.

[14] Kosugi, N., Y. Nishihara, S. Kon'ya, M. Yamamuro, and K. Kushima, "A Practical Query-By-Humming System for a Large Music Database," *In Proc. ACM Multimedia*, 2000.

[15] Lambrou, T. *et al.*, "Classification of Audio Signals Using Statistical Features on Time and Wavelet Transform Domains," in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 6, pp. 3621-3624, 1998.

[16] Li, S. Z., "Content-Based Audio Classification and Retrieval Using the Nearest Feature Line Method," *IEEE Transactions on Speech and Audio Processing*, Vol. 8, No. 5, pp. 619-625, Sept. 2000.

[17] Liu, C. C., A. J. L. Hsu, and A. L. P. Chen, "Efficient Theme and Non-Trivial Repeating Pattern Discovering in Music Databases," in *Proc. of IEEE Intl. Conf. on Data Engineering*, pp. 14-21, 1999.

[18] Liu, C. C., A. J. L. Hsu, and A. L. P. Chen, "An Approximate String Matching Algorithm for Content-Based Music Data Retrieval," *in Proc. of IEEE Intl. Conf. on Multimedia Computing and Systems,* 1999.

[19] Liu, C. C., and Wei-Yi Kuo, "Content-Based Segmentation of MP3 Music Objects," *in Proc. of the Workshop on the 21st Century Digital Life and Internet Technologies*, 2001.

[20] Liu, C. C. and Po-Jun Tsai, "Content-Based Retrieval of MP3 Music Objects," in *Proc. of the ACM Intl. Conf. on Information and Knowledge Management (CIKM 2001)*, 2001.

[21] Liu, Z. *et al.*, "Audio Feature Extraction and Analysis for Scene Classification," in *Proc. IEEE First Workshop on Multimedia Signal Processing*, pp. 343-348, 1997.

[22] Liu, Z. and Q. Huang., "Classification of Audio Events in Broadcast News," in *Proc. IEEE Workshop on Multimedia Signal Processing*, pp. 364-369, 1998.

[23] Lu, G.J. and T. Hankinson, "A Technique Towards Automatic Audio Classification and Retrieval," in *Proc. IEEE Intl. Conf. on Signal Processing*, Vol. 2, pp. 1142-1145, 1998.

[24] Lu, G.J. and T. Hankinson, "An Investigation of Automatic Audio Classification and Segmentation," in *Proc. IEEE Intl. Conf. on Signal Processing*, Vol. 2, pp. 776-781, 2000.

[25] Martin, K. D., and Y. E. Kim, "2pMU9. Musical instrument identification : A pattern-recognition approach," *in the 136th meeting of the Acoustical Society of America*, October 13, 1998.

[26] Melih, K., and R. Gonzalez, "Audio Retrieval Using Perceptually Based Structures", in *Proc. of IEEE International Conference on Multimedia Computing and system*, pp 338-347, 1998.

[27] Melih, K., and R. Gonzalez, "Audio Source Type Segmentation Using a Perceptually Based Representation," *in ISSPA 99, Brisbane, Australia, 22-25 August, 1999.*

[28] Mo, J. S., C. H. Han, and Y. S. Kim, "A Melody-Based Similarity Computation Algorithm for Musical Information," *in Proc. of Knowledge and Data Engineering Exchange Workshop（KDEX'99）*, pp. 114 –121, 1999.

[29] Moreno, P.J. and R. Rifkin, "Using The Fisher Kernel Method for Web Audio Classification," in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 4, pp. 2417-2420, 2000.

[30] Noll, P., "MPEG Digital Audio Coding," *IEEE Signal Processing Magazine*, Vol. 14, No. 5, pp. 59-81, Sept. 1997.

[31] Painter, T. and A. Spanias, "Perceptual Coding of Digital Audio," *Proceedings of the IEEE* , Vol. 88, No. 4 , pp. 451–515, April 2000.

[32] Pan, D., "A Tutorial on MPEG/Audio Compression," *IEEE Multimedia Magazine,* Vol. 2, No. 2, pp. 60-74, Summer 1995.

[33] Rolland, P. Y., G Raskinis, and J. G. Ganascia, "Musical Content-Based Retrieval: an Overview of the Melodiscov Approach and System," In *Proc. ACM Multimedia 99*, pp. 81-84, 1999.

[34] Saunders, J., "Real-Time Discrimination of Broadcast Speech/Music," in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 2, pp. 993-996, 1996.

[35] Scheirer, E. and M. Slaney, "Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator," in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 2, pp. 1331-1334, 1997.

[36] Smith, G., H. Murase, H. Kashino, "Quick Audio Retrieval Using Active Search", in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing,* Vol. 6, pp. 3777-3780, 1998.

[37] Tsai, Po-Jun and Chih-Chin Liu, "An MP3 Search Engine on the Internet", *in Proc. of 2000 Workshop on Internet & Distributed Systems,* Vol. 1, pp. 18-27, 2000.

[38] Tzanetakis, G., G. Essl, and P. Cook, "Automatic Musical Genre Classification of Audio Signals," in *Proc. Int. Symposium on Music Information Retrieval (ISMIR)*, Bloomington, Indiana, 2001.

[39] Tzanetakis, G., and P. Cook, "A Framework for Audio Analysis Based on Classification and Temporal Segmentation," *in Proc. EUROMICRO Conf.*, Vol. 2, pp. 61-67, 1999.

[40] Wold, E., T. Blum, D. Keislar, and J. Wheaton, "Contented-Based Classification, Search, and Retrieval of Audio", *IEEE Multimedia Vol. 3, No. 3,* pp. 27-36, Fall 1996.

[41] Zhang, T. and C.-C.J. Kuo, "Hierarchical Classification of Audio Data for Archiving and Retrieving," in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 6, pp. 3001-3004, 1999.