12-2006

Towards effective content-based music retrieval with multiple acoustic feature composition

Jialie SHEN
Singapore Management University, jlshen@smu.edu.sg

John Shepherd

Ngu Ahh

DOI: https://doi.org/10.1109/tmm.2006.884618

Published in IEEE Transactions on Multimedia, 2006 December, Volume 8, Issue 6, Pages 1179-1189
https://doi.org/10.1109/tmm.2006.884618

Towards Effective Content-Based Music Retrieval With Multiple Acoustic Feature Combination

Jialie Shen, John Shepherd, and Anne H. H. Ngu

Abstract—**In this paper, we present a new approach to constructing music descriptors to support efficient content-based music retrieval and classification. The system applies multiple musical properties combined with a hybrid architecture based on principal component analysis (PCA) and a multilayer perceptron neural network. This architecture enables straightforward incorporation of multiple musical feature vectors, based on properties such as timbral texture, pitch, and rhythm structure, into a single low-dimensioned vector that is more effective for classification than the larger individual feature vectors. The use of supervised training enables incorporation of human musical perception that further enhances the classification process. We compare our approach with state of the art techniques and demonstrate its effectiveness on content-based music retrieval. In addition, extensive experimental study illustrates its effectiveness and robustness against various kinds of audio alteration.**

Index Terms—**Classification, multimedia database, music retrieval.**

I. Introduction

ADVANCES in information technology, such as digital libraries, the World Wide Web, and peer-to-peer information systems, are producing an ever-growing volume of music data. Unfortunately, the technology for generating effective music descriptors has not kept pace with the growth of music data. While the extraction of acoustic features from digital music data has a relatively long history, it has so far proved extremely difficult to determine how to effectively represent high-level semantic concepts, such as genre, using physical features from the acoustic signal. The are several reasons for this. First, there exists a large gap between high-level semantic concepts and low-level physical representation of music [2]. Second, there is a wide variety of features within a music signal (e.g. timbral texture, harmony, rhythm); thus, using a single acoustic feature may not accurately represent important characteristics of the music data. Third, human beings have an amazing and unique capability to perceive music which should be taken into account for developing effective music classification and retrieval.

In developing effective music descriptors, we are faced with the problem of producing music feature vectors that accurately mimic human music perception for a range of retrieval and classification tasks. The two subproblems here are: 1) how to combine various low-level features to effectively model human perception for a given task and 2) how to avoid producing composite feature vectors (from multiple acoustic features) that are so large as to render existing data access and machine learning methods unusable due to the "curse of dimensionality" [1], [20].

The first problem is associated with human music perception. Recent studies in music perception and cognition [4], [22] support the belief that human beings perceive music by combining different acoustic features in a "nonlinear" way. Thus, techniques assuming that each type of acoustic features contributes equally in music recognition are not supported by our understanding of the human perpetual system.

The second problem is related to computational complexity. The standard approach for dealing with this problem is to use dimension reduction methods (such as DFT, PCA, SVD, and neural networks) to prune the size of the feature vectors. However, these commonly used methods suffer from either an inability to capture non-linear correlations among raw data, which leads to significant loss of useful distance information in the reduced feature space, or very expensive training costs for tasks where machine learning is needed.

Motivated by the above, in this paper, we present a fast and robust descriptor generation method for music data, which is called *InMAF*.[1] Unlike conventional approaches, our method easily integrates various acoustic features and human musical perception to produce small feature vectors that enhance the retrieval and categorisation process. Experimental results demonstrate that our proposed method outperforms state-of-the-art approaches in some important areas. For example, it achieves around 24% improvement for genre classification accuracy on the dataset of [30], 26% improvement for artist classification accuracy, and 23% improvement for instrument classification accuracy against *DWCHs* [18], one of the best existing methods for content-based music retrieval (CBMR). It also yields nearly 27% improvement in the average precision rate against *DWCHs* for three different music retrieval query cases. In addition, real-life applications often deal with music data that suffers from noise or audio distortion. Our experimental results show that *InMAF* is robust against common types of audio alteration of music data.

The rest of the paper is structured as follows. Section II gives background knowledge and describes related work. Section III presents the architecture of the proposed system. Section IV describes a performance study and gives a detailed analysis of a comprehensive set of experiments over three large music

Manuscript received November 28, 2004; revised January 27, 2006. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Arnold W. M. Smeulders.

J. Shen and J. Shepherd are with the School of Computer Science and Engineering, University of New South Wales, Sydney NSW 2052, Australia (e-mail: jls@cse.unsw.edu.au; jas@cse.unsw.edu.au).

A. H. H. Ngu is with the Department of Computer Science, Texas State University, San Marcos, Texas 78666 USA (e-mail: angu@txstate.edu).

Digital Object Identifier 10.1109/TMM.2006.884618

[1] *InMAF* stands for **In**tegrating **M**ultiple **A**coustic **F**eatures.

databases. Finally, in Section V, we draw some conclusions and indicate future directions for this work.

II. Background

A. Music Content Representation

Various kinds of feature can be used for classifying and indexing large music collections. They include text labels for the title and performer(s)/composers and symbolic representations of melody (e.g. MIDIs and digital music scores) [10], [26]. In this paper, we focus on acoustic features.

While various systems exist for content-based speech recognition and music-speech discrimination, much less work has focused on developing compact and comprehensive music data descriptors for effective categorization and retrieval. Most of the existing work is based on spectral features of the raw music signal adapted from earlier work in speech recognition including mel-frequency cepstral coefficients (MFCCs), spectral centroid, linear prediction coefficients, spectral flux, etc. [24]. Typical examples of this approach are the work by Nam and Berger [21], who use three low-level acoustic features (spectral centroid, short time energy, and zero crossing rate) for automatic music genre classification, and work by Li and Khokhar [16], who propose nearest-feature-line methods for content-based audio retrieval and classification. In [19], Lu *et al.* studied audio classification with nine different audio features including MFCCs, zero crossing rates (ZCR), short time energy (STE), subband power distribution, brightness, bandwidth, spectrum flux (SF), band periodicity (BP) and noise frame ratio (NFR) using a support vector machine (SVM) [32] as a classifier. One of the most advanced frameworks for modelling music signals is *MARSYAS*,[2] developed by Tzanetakis *et al.* [30]. In this framework, a set of features was specifically developed to characterize different acoustic properties of music signals, including timbral texture, pitch content and rhythm. Using a linear concatenation of these features, they achieved 61% classification accuracy for a ten genre sound-data set. More recently, Li *et al.* [18] proposed using Daubechie's wavelet histogram technique (*DWCHs*) to capture local and global temporal information inside music signal. Their approach first used wavelets to decompose a music signal into different subbands. Then, a histogram for each subband was constructed. Finally, the first three moments of each histogram and energy for each subband are calculated to form *DWCHs*.[3] Due to its effective estimation of probability distribution over time and frequency via wavelets, *DWCHs* performs better than *MARSYAS*, and is currently the state-of-the-art in content-based music retrieval.

The problem with the above techniques, which rely on either single type of physical feature or a linear concatenation of many features, is that they cannot provide a "perceptually accurate" description of a music signal. The human perceptual system interprets and processes a music signal using various kinds of acoustic characteristics within a complex context. A single type of physical feature may not provide information which is rich enough to represent music data comprehensively. Also, approaches using multiple acoustic characteristics assume that a linear combination of low-level physical features can reflect how we perceive music as similar. This assumption is not supported by experimental work on human music perception.

B. Dimension Reduction Methods

Methods such as *DWCHs* and *MARSYAS* produce high-dimensional music descriptors, which render inefficient all state-of-the-art data access and machine-learning methods for searching, training and classification. Therefore, it is necessary to apply dimension reduction techniques to eliminate any redundancy amongst low-level features after the signal processing stage. The goal of a dimension reducer is to discover complex dependencies among the different features and eliminate correlated information or noise while maintaining sufficient information for discrimination between music of different classes. In order to be effective, the feature space resulting from the reduction must accurately reflect the discriminative criteria of human music perception. Currently, dimension reduction methods can be classified into two general categories: linear dimension reduction (LDR) and nonlinear dimension reduction (NLDR). Typical examples for LDR include PCA, multidimensional scaling (MDS), SVD, and DFT [5], [11], [14], [25]. These approaches assume that the variance of data can be accounted for by a small number of eigenvalues. Thus, LDR works well with data sets that exhibit some linear correlation. In our case, since acoustic features are nonlinear in nature, better performance can be expected by using NLDR. The advantage of using a neural network for NLDR is that it can learn directly from training examples (such as human prelabeled data) to form a model of the feature data. The basis for NLDR is the standard non-linear regression analysis used in the neural network approach, which has been widely studied [9], [13], [33]. Through training, the distance information of the original data source can be represented as weights between units in successive layers of the neural network. Thus, NLDR should perform better than LDR in handling feature vectors for music data. However, there is significant cost involved in training a neural network.

III. System Overview

In this section, we present a new approach to extracting descriptive information from music data. Its advantage over previous approaches, such as *MARSYAS* and *DWCHs*, lies in the fact that it can capture high level semantic concepts (such as genre) and represent those features as low-dimensional feature vectors. It achieves this by combining acoustic features and human perceptual data. Before describing the system architecture, we give a brief overview of the acoustic features that our system uses.

A. Feature Extraction

In this study, we use the *MARSYAS* framework [30] as the basis for extracting different acoustic features. *MARSYAS* classifies acoustic features into timbral texture, rhythmic content, and pitch.

- **Timbre**: Timbral texture is a global statistical music property used to differentiate a mixture of sounds. It has been widely applied for speech recognition. To extract timbral texture, we first divide each music signal into many short

[2] In this paper, we use *MARSYAS* to represent the feature set generated by *MARSYAS* framework.

[3] It means Daubechie's wavelet coefficient histograms.

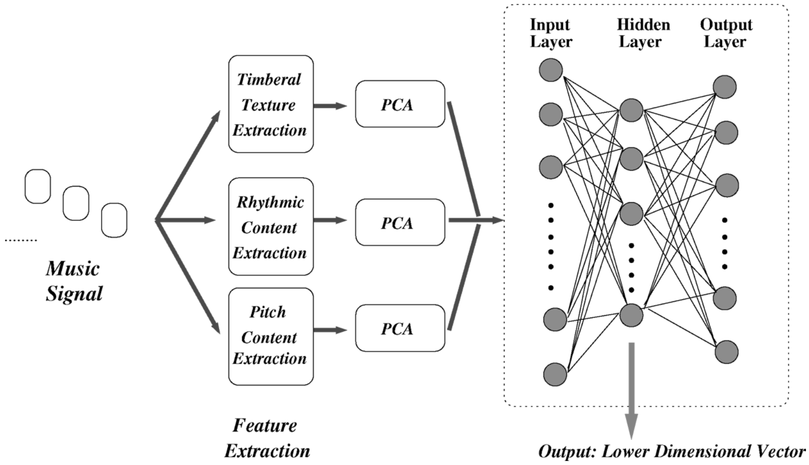

Fig. 1. Architecture of a hybrid musical feature dimension reduction scheme. The output of the scheme is the low-dimensional vectors appearing in the hidden layer of the neural network.

time-frames. Different components (mainly spectral characteristics) for this feature vector are calculated using the short-time Fourier transform (STFT). These components include spectral centroid, spectral flux, time domain zero crossings, low energy, spectral roll-off and MFCCs. This yields a 33-dimensional vector containing: mean and varianceof spectral centroid, spectral flux, time domain zero crossings, and 13 MFCC coefficients (32) plus a low-energy component (1).

- **Rhythm**: Rhythmic content indicates reiteration of musical signal overtime. It can be represented as beat strength and temporal pattern. We use the beat histogram (BH) proposed by Tzanetakis *et al.* [30] to represent rhythmic information. BH is calculated by collecting statistics on the amplitude envelope periodicities of multiple frequency bands. The specific method for their calculation is based on a discrete wavelet transform (DWT) and analysis of periodicity for the amplitude envelope in different octave frequency bands. An 18-dimensional vector is used to represent rhythmic information, containing: relative amplitude of the first six histogram peaks (divided by the sum of amplitudes), ratio of the amplitude of five histogram peaks (from second to sixth) divided by the amplitude of the first one, period of the first six histogram peaks, and overall sum of the histogram.
- **Pitch**: Pitch is used to characterize melody and harmony information in music and can be extracted via multi-pitch detection techniques. We use an algorithm proposed by Tolonen *et al.* [31]. The signal is first divided into two frequency bands (below and above 1000 Hz). Then, amplitude envelopes are extracted for each frequency and summed to construct a pitch histogram. The resulting 18-dimensional pitch vector incorporates: the amplitude and periods of the maximum six peaks in the histogram, pitch interval between the six most prominent peaks, and the overall sum of the histogram.

B. The System Architecture

InMAF utilises a two-tier hybrid architecture: dimension reduction via Principal Component Analysis followed by a nonlinear neural network using the Quick-prop learning algorithm [8]. Fig. 1 shows the overall architecture of the system. Feature vectors for timbre (33 dimensions), rhythm (18 dimensions) and pitch (18 dimensions) are first extracted from the music data. In the first tier, each acoustic feature is separately analysed by a single PCA module.[4] The variance of PCA analysis is set to be 99% and after PCA preprocessing, the dimensionalities of timbre, pitch, and rhythm are 10, 8, and 7, respectively. The outputs of each PCA module are then concatenated to form a single 25–dimensional composite feature vector as input to the three-layer perceptron feed-forward neural network.[5]

1) Principal Component Analysis: PCA is one of the most widely used dimensionality reduction methods [6]. The advantage of the PCA transformation is that any linear correlations in the data are automatically detected. It has been successfully employed for dimension-reduction in applications such as building efficient indexes for general image retrieval [15], and for image coding in specialised image databases, such as facial images [27]. In our system, PCA is used as a “pre-processing” step, where it provides small but information-rich feature vectors for the three-layer neural network, and thus speeds up the NLDR training time.

2) Neural Network: The advantage of using a neural network for NLDR is that the network can be *trained* to produce compact and effective music descriptors via pre-selected samples. In this work, a three-layer perceptron neural network with a Quick-prop learning algorithm [8], is used to perform non-linear dimensionality reduction on composite music vectors. The units in the input layer accept the composite feature vector from the PCA analysis. The number of units in the output layer correspond to the total number of classes in target data collection. However, it is the hidden layer that plays a critical role in our method. When the network has been successfully trained, the weights that connect the input and the hidden layers can be

[4] An alternative approach to implementing *InMAF* would be to first combine the feature vectors into a high-dimensional composite feature vector and then apply PCA to this composite feature vector to get a reduced vector to be used as the input of the neural network. We have also tested this approach and obtained similar experimental results to those presented below.

[5] In fact, we use total variance of PCA to control dimensionality. Based on our experiments, a 25-dimensional feature vector was a good tradeoff between neural network training cost and effectiveness of dimension reduction. Further discussion and analysis of this issue is presented in Section IV-E-1.

```
Input          : $seed\_set$, $MaxSamSize$, $classNum$
Output         : $ts$
Initialization : setup three individual indexes based on
                 timbre, pitch and rhythm feature;
$ts = \emptyset$;
1  for each subset $s_i$ in $seed\_set$ do
2      $ts_i = \emptyset$ ;
3      while $|ts_i| <$ MaxSampleSize/(classNum) do
4          if $s_i == \emptyset$ then
5              // New seed for class i is picked up by human
               expert.
6              $s_i$ = SelectNewSeed(DomainExpert, i);
7          end
8          for each music $m$ in $s_i$ do
9              $lr_m$ = Knn(rhythmIndex, $m$);
10             $lt_m$ = Knn(timbreIndex, $m$);
11             $lp_m$ = Knn(pitchIndex, $m$);
12             $lf_m$ = $lr_m \bigcap lt_m \bigcap lp_m$;
13             $ts_i$ = $ts_i \bigcup$ Pick(DomainExpert, $lf_m$);
14             $s_i$ = $s_i - m$;
15         endFor
16     end
17     $ts = ts \bigcup ts_i$;
18 endFor
19 retrun($ts$)
```

Algorithm 1. Training sample selection

```
Input          : $neural\ network,\ cutoff\ value\ \psi\%$
                 $timbre\ feature\ matrix\ M_t$
                 $rhythm\ feature\ matrix\ M_r$
                 $pitch\ feature\ matrix\ M_p$
Output         : $trained\ system$
Initialization : Initialize neural network with small random
                 values;
1  for each type of feature matrix do
2      Apply the PCA analysis with predefined cutoff value $\psi\%$;
3  end
4  Select the training samples using Algorithm 1;
5  Construct the composite feature vectors $\mathbf{z}_k$ for training
   examples from PCA-preprocessed timbre, rhythm and pitch
   feature vectors via linear concetanation;
6  Prepare the training patterns $(\mathbf{z}_k, c_k)$, where $c_k$ is the class
   number and corresponding the composite feature vector $\mathbf{z}_k$;
7  Present the training patterns $\mathbf{z}_k$ as input and $c_k$ as output to
   the network;
8  Use the Quick-prop learning algorithm to update the weights
   of the network;
9  Test the convergence of the network;
10 if the convergence condition is satisfied then
11     Stop the training process;
12 else
13     Go back to step 7 and do retraining;
14 end
```

Algorithm 2. Training algorithm for neural network component in InMAF

treated as entries of a transformation that maps "raw" feature vectors to more compact vectors. Thus, when a high-dimensional feature vector is passed through the network, its activation values in the hidden units form a lower-dimensional vector. Each lower dimensional vector preserves only the most discriminating information from the original feature vector.

C. Human Musical Perception Integration

The training process in our system has two stages: 1) construct a training set incorporating human musical classification and 2) use the training examples to generate compact music descriptors.

1) Training Sample Selection: The training stage begins with the incorporation of human music perception by defining a classification scheme for music data, determined by the specific application (e.g. classification by genre, artist, or instrument). This classification scheme is used to choose training examples; examples are required from each class. In our experiments, we consider three classification schemes (genre, artist, instrument) as a basis for determining similarity of music data. Note that each classification task requires its own feature descriptor, and thus needs its own training set.

The training set ts is generated by the procedure shown in Algorithm 1. Its main goal is to construct a set of trained examples with size specified by the parameter $MaxSamSize$, which represents classification criteria based on human music perception for subsequent training. categorise music into classes. The classification scheme determines the number of classes $classNum$. For example, in the genre similarity experiment, we consider ten genres ($classNum = 10$). After manually choosing a set s_i of examples for each class, we combine them to form a single $seed\_set$. Each $s_i \in seed\_set$ contains music samples selected with the help of a domain expert to represent the corresponding class. In our work, we used "ground truth" classifications from professional reviews and similarity judgements from western mainstream online music services.[6] To build the training samples for each class, for each music sample $m \in s_i$, we make a k nearest neighbour search based on three single features including timbre, rhythm and pitch (lines 9∼11). Then, a single list lf_m is obtained via intersection of the three result lists. The music in lf_m is similar in timbre, in rhythm and in pitch (line 12). Next, with the aid of the domain expert, we pick up all music pieces belonging to the same class m from lf_m and add them into ts_i which contains trained examples for class i (line 13). In the following step, if the number of training examples for one class reaches a predefined threshold which is equal to $MaxSamSize/classNum$, those examples will be added into ts (line 17).

In some cases, not all $s_i \in seed\_set$ may generate sufficient training examples. In this case, we ask the domain expert to generate a new set of seeds for each class i where $|ts_i| < MaxSamSize/classNum$, and then repeat the above procedure until each subclass is large enough.

2) System Training: As shown in Algorithm 2, in order to train the system, we first set up a PCA dimension reducer for each type of raw feature vector (lines 1–3). Note that we use the entire data set, and not just the training set, in determining the principal components (PCs). This has the advantage that the covariance matrix for each type of feature vector contains the global variance of music in the database. The number of PCs is determined by the cutoff value ψ. In this study, ψ is set so that the minimum variance retained after PCA dimension reduction is at least 99%. Based on our experiments, the cutoff value ψ significantly influences the training cost. We discuss this later.

The neural network is initialized by setting the weight of each link, connecting two units in the network, to a random small value. We used Algorithm 1 to obtain a training sample

[6] In this paper, we use service from http://www.allmusic.com (AMG).

(line 4). The training then proceeds by iterating over the music data items in the training set, choosing one item from each subclass in turn (lines 7–8). For each item, we construct a composite feature vector using a linear concatenation of the PCA-reduced timbre, pitch, and rhythm feature vectors (lines 5–6). The composite feature vector and the class number is then presented to the neural network. Finally, we test the convergence of the network (line 9). If the convergence condition is satisfied, the training process halts (line 11). Otherwise, we continue to present training examples, one at a time (line 13) until convergence.

IV. A Performance Study

In this section, we demonstrate the effectiveness of our approach by comparing it with the current best approaches in the areas of classification and similarity retrieval, and also consider robustness against different kinds of audio distortion. The study examines a range of possible methods for generating music descriptors, including our proposed method *InMAF* and state-of-the-art existing approaches like *DWCHs*, *MARSYAS* (denoted by MAR)[7] and two other dimension reduction methods including PCA and neural networks (denoted by NN) [29]. For each of these (except *DWCHs*), we consider three different combinations of low-level features ($\mathrm{Rhythm}+\mathrm{Timbre}+\mathrm{Pitch}$ denoted by RTP; $\mathrm{Timbre}+\mathrm{Rhythm}$ denoted by TR; $\mathrm{Timbre}+\mathrm{Pitch}$ denoted by TP). In our results, a system configuration denoted by "xxxx-yy" contains feature extraction method "xxxx" with feature combination "yy". For example, "InMAF-RTP" denotes a configuration using our proposed method with rhythm, timbre, and pitch features. The size of feature vectors generated by pure neural network (NN) and *InMAF* is 10, which is equal to the number of neurons in the hidden layer of the multilayer perceptron. Also, the number of neurons in the input layer is 25, equal to the size of the PCA-preprocessed feature vector with 99% total variance. All of the experiments were conducted on a Pentium III machine with 450-MHz CPU, 256-MB RAM running under Linux.

A. Datasets

Three separate music databases were used in this performance study. The first, Dataset I, is used for testing the performance of different kinds music descriptors in genre classification. It contains 1000 music data items covering ten genres with 100 songs per genre. This dataset was used in [18], [30]. To ensure variety of recording quality, the excerpts of this dataset were taken from radio, compact disks, and MP3 compressed audio files. Each item in the collection belongs to exactly one of ten music genre categories: Classical, Country, Dance, Hip-hop, Jazz, Reggae, Metal, Blues, and Pop. The second dataset, Dataset II, is used for testing the performance of music descriptors generated by different methods on artist classification. It contains 1000 songs covering 20 different artists. This dataset was constructed from the CD collection of the first author and his friends. It includes ten male singers (such as Van Morrison, Michael Jackson, Elton John, etc.) and ten female singers (such as Kylie Minogue, Madonna, Jennifer Lopez, etc.), with 50 songs for each singer. Dataset III contains 1000 sounds covering 10 different solo instruments such as piano, guitar, violin, etc, and there are 100 music items for each instrument category. This dataset is used for instrument-based classification and similarity search. The length of each music item in all three datasets is 30 s and each item is stored as a 22050-Hz, 16-bit, mono audio file.

B. Automatic Music Classification

In this section, we compare the performance of music descriptors produced using our approach against descriptors produced via the *DWCHs* method, currently the best known approach. We perform the comparison for three separate music classification tasks: genre-based classification, artist-based classification and instrument-based classification. The classifers used in this study include support vector machines (SVMs), K-nearest neighbour (KNN), Gaussian mixture models (GMM), decision tree and linear discriminant analysis (LDA).

1) Effectiveness of Classification: Table I[8] shows the results of our experiments to test the accuracy of genre classification and artist classification for different classifiers using a variety of music descriptors as input. The classification problems were carried out on different data sets (the ones described in Section IV-A). For each of the classifiers, we used tenfold cross validation to calculate classification accuracy [20]. This means the whole dataset is divided into ten disjoint subsets of (approximately) equal size. For testing, we trained classifiers on nine of these ten disjoint subsets and then tested on the remaining one, each time leaving out a different subset. The above process was repeated for each approach to generate music descriptors, including our *InMAF* approach (with different combinations of acoustic features), pure neural network, PCA, *DWCHs*, and *MARSYAS* with linear concatenation.

The bottom three rows of the Table I indicate how the different classifiers performed if only individual raw acoustic features were used in the descriptor. The poor accuracy observed in this experiment (between 30% and 50% for all classifiers) verifies the claim that effective music classification cannot be achieved by considering only a single low-level acoustic feature. The same conclusion was reached by [30]. Some improvement in accuracy can be obtained by considering a combination of low-level features. We considered all different linear concatenations of combinations of the timbre, pitch and beat vectors. The best linear combination (MAR-RTP) uses all three low-level features and achieves accuracy rates of 70.1% for genre classification, 69.2% for artist classification and 67.3% for instrument classification with the best classifier (SVMs). Similar observations can be obtained in case of using pure PCA as the dimension reduction method. The performance with *DWCHs* is better than any linear concatenation of acoustic features and better than pure PCA. This is because *DWCHs* provide a good estimation of probability distribution over time and frequency which leads to a better feature representation.

In comparison with PCA, *DWCHs* and *MARSYAS*, constructing music descriptors with *InMAF* results in a significant improvement in classification accuracy for all of the different

[7] Note that in MARSYAS, linear concatenation is used to construct a composite feature vector as input to different machine-learning based classifiers.

[8] SVM1 and SVM2 denote Support Vector Machine with one-versus-the-rest and pairwise approach. Dec. tree denotes decision tree.

TABLE I

Classification Accuracy of Different Learning Methods With Various Music Descriptor Construction Methods. G Denotes Accuracy of Genre Classification on Dataset I, A Denotes Accuracy of Artist Classification on Dataset II and I Denotes Accuracy of Instrument Classification on Dataset III. For the Bottom Three Rows, Results are Obtained Using Raw Feature Vectors

Feature Extraction Methods	Classification Methods																	
	SVM1(%)			SVM2(%)			GMM(%)			KNN(%)			Dec. tree(%)			LDA(%)		
	G	A	I	G	A	I	G	A	I	G	A	I	G	A	I	G	A	I
InMAF-RTP	89.7	90.7	88.2	90.2	89.6	87.3	81.4	81.3	80.7	85.3	86.5	83.2	81.7	80.4	79.2	84.5	80.5	81.2
InMAF-TR	80.1	81.6	79.5	82.6	79.6	78.2	70.6	70.3	72.3	73.4	75.4	74.1	70.8	73.6	74.5	73.5	75.6	74.2
InMAF-TP	79.5	80.6	80.1	80.5	78.3	79.5	71.7	72.7	73.4	73.8	72.5	75.2	71.2	70.9	73.6	72.3	74.7	75.6
NN-RTP	91.9	91.7	89.1	92.2	90.4	86.2	83.1	83.5	80.8	87.1	86.5	83.7	83.5	82.1	78.5	84.4	80.2	82.5
NN-TR	79.5	78.8	80.1	82.8	83.1	79.2	71.1	70.7	72.8	74.9	75.1	75.2	72.5	73.2	76.2	73.7	75.5	74.8
NN-TP	79.5	78.8	80.2	82.8	83.1	80.2	72.1	73.5	73.8	72.9	71.8	75.1	71.5	72.3	73.2	73.1	74.5	76.8
DWCHs	75.5	73.4	72.4	75.2	76.2	74.5	68.2	68.8	67.4	68.3	68.6	67.3	71.2	69.7	70.3	71.3	69.3	70.2
MAR-RTP	68.7	69.5	65.7	70.1	69.2	67.3	61.7	63.1	62.7	59.5	59.9	58.2	68.1	67.8	65.4	69.4	63.2	61.2
MAR-TR	65.1	64.7	62.4	65.7	65.1	63.5	60.5	59.8	58.2	61.7	61.4	54.3	67.3	68.1	63.2	68.2	61.5	59.8
MAR-TP	68.2	69.7	65.3	68.2	71.2	65.2	60.7	61.2	57.4	61.3	58.9	54.2	68.4	68.9	62.9	67.5	59.4	58.7
PCA-RTP	67.9	70.7	73.1	69.2	71.4	68.2	67.1	67.5	65.8	67.9	69.7	65.3	68.3	67.2	66.2	63.7	64.5	67.8
PCA-TR	61.9	63.7	67.1	62.2	66.4	63.4	60.1	63.5	60.8	61.4	61.7	59.1	63.2	60.4	61.2	57.1	58.9	60.1
PCA-TP	59.2	61.5	66.8	62.0	61.4	62.9	59.7	62.9	60.7	61.2	61.3	60.4	64.0	58.4	61.5	58.2	58.5	59.8
Beat	30.5	29.9	34.5	32.1	30.8	31.7	36.6	36.9	33.5	31.3	33.2	32.5	37.8	39.5	37.4	24.9	31.3	30.4
Timbre	49.9	48.2	46.3	50.7	52.3	52.8	45.1	46.9	44.7	47.2	50.1	49.3	49.5	49.1	48.3	55.3	43.5	45.7
Pitch	33.8	31.2	32.6	35.7	37.5	36.8	38.2	37.8	39.4	32.1	34.2	32.3	37.2	39.2	38.5	31.2	30.5	33.9

TABLE II

Training Time of Different Classifiers With Various Music Descriptor Construction Methods for Music Classification

Feature Extraction	Genre Classification					Artist Classification					Instrument Classification				
	SVMs	GMM	KNN	DT	LDA	SVMs	GMM	KNN	DT	LDA	SVMs	GMM	KNN	DT	LDA
InMAF-RTP	2.91s	2.44s	2.43s	0.41s	2.93s	2.15s	2.07s	2.44s	0.26s	2.13s	2.15s	2.07s	2.44s	0.39s	2.25s
InMAF-TR	2.92s	2.52s	2.42s	0.42s	2.82s	1.94s	1.58s	1.53s	0.28s	1.85s	2.10s	2.12s	2.57s	0.43s	2.08s
DWCHs	4.22s	4.43s	4.68s	0.98s	3.92s	2.62s	2.93s	2.97s	0.61s	2.74s	4.09s	4.01s	4.23s	1.12s	4.08s
MAR-RTP	4.76s	5.15s	5.12s	1.49s	4.56s	2.83s	3.15s	3.07s	1.08s	2.53s	5.01s	5.14s	5.05s	1.58s	4.01s
MAR-TR	4.41s	4.31s	4.16s	1.18s	4.36s	2.82s	2.77s	2.68s	0.71s	2.67s	4.56s	4.24s	4.01s	1.21s	4.39s

TABLE III

Training Cost of Dimension Reduction Methods

Dataset	Training Cost of Dimension Reduction Methods(min:sec)							
	InMAF-RTP	InMAF-TR	InMAF-TP	InMAF-PR	NN-RTP	NN-TR	NN-TP	NN-PR
I	21:34	18:23	18:52	18:07	26:02	23:32	22:39	20:34
II	20:15	18:45	19:05	18:09	24:43	24:05	21:55	22:04
III	21:30	17:54	19:06	18:08	25:15	22:55	23:15	21:22

classifiers. For example, in case of genre classification, the range of the improvement with *InMAF* against *DWCHs* is from 18% to 24%, depending on the learning method used. For classification by artist, the improvement range is from 15% to 26% and 13% to 23%. Among all classification methods, SVMs give the best results, whatever kind of music descriptor is used. The accuracy achieved with the one-versus-the-rest SVMs to classify music by genre, artist and instrument is 89.7%, 90.7%, and 88.2%, respectively. On the other hand, based on Table I, the pure neural network approach yields lower classification error rates than other approaches. However, for *InMAF*, use of PCA-reduced low-level feature vectors as input to the neural network does not significantly reduce categorisation accuracy, but does provide a great reduction in the classification time. In the next section, we study the efficiency of music classification with various music descriptor generation methods.

2) Efficiency of Classification: Using a large input feature vector can make the learning process for any classifier very inefficient in terms of training time. Using a small but well-discriminating feature vector generated by *INMAF* not only provides superior classification accuracy but also saves a large amount of training time. To further illustrate the performance advantage of using *InMAF*, we computed the actual training time for different learning methods with music descriptors generated by various methods. The results in Table II indicate that the speedup due to our proposed method is significant. For example, training the SVMs with *MARSYAS*[9] and *DWCH*s required 4.76 s and 4.22 s for genre classification, respectively. In contrast, our proposed approach needed only 2.91 s, nearly 38% and 31% saving.

On the other hand, although it can be seen that superior classification accuracy can be achieved using pure neural network from Table I, the approach suffers from very long learning time. This is because time required for a typical learning algorithm, such as back-propagation (BP), grows at super-linear rate with number of inputs. Thus, compression of data through certain kinds of transformation potentially yields a significant advantage in terms of time complexity. Based on this principle, *InMAF* uses PCA as the first layer of the hybrid architecture to preprocess raw music feature vectors. Results from Tables I and III show that this approach does not lose significant classification accuracy, but substantially improves the network training cost; training a neural network to achieve 91.9% with SVMs for genre classification on dataset I required about 26 min to finish. In contrast, our *InMAF* approach require

[9] *MARSYAS* uses linear concatenation of three acoustic features to construct input feature vectors.

about 21 min to complete learning process and results in 89.7% classification accuracy. There is a significant saving on training time and similar observation can be obtained for classification with the other two similarity notions. Thus, we can conclude that the *InMAF* is a highly *effective* and *efficient* technique of generating music descriptors for automatic music classification.

C. Music Retrieval

In this section, we present the results of an experiment to verify the effectiveness of our approach for content-based music retrieval. Music retrieval can be informally defined as: the user submits a query music clip and the system retrieves a list of music pieces from the database that are most similar; the list of "matching" pieces is displayed in order starting from the most similar. There are clearly many different notions of "similarity" for music; each notion of similarity corresponds to one kind of query. In this study, we consider the following types of similarity.

- **Type I:** find music that has similar genre from the database constructed using dataset I.
- **Type II:** find music performed by the same artist from the database constructed using dataset II.
- **Type III:** find music with the same instrument from the database constructed using dataset III.

Under the *InMAF* approach, we need to build one music descriptor generator for each of the above similarity notions. Once we are able to generate descriptors for a particular similarity notion, we use them to build a hybrid-tree [12] indexing structure as the basis for similarity retrieval, using the well-known Euclidean distance as the similarity metric. Finally, we set up three different indexes to facilitate three types of similarity search as described above respectively.

In our experiment, we randomly selected 10% of the target dataset as query examples, where these query examples uniformly cover all subclasses. This test was repeated 1000 times. Also, we evaluated the top 100 sounds ranked in terms of similarity measurement. Since not all relevant sounds are examined in this experiment, the concepts of normalised precision (P_n) and normalised recall (R_n) [28] were used to evaluate the performance of similarity retrieval for different query types. They can be defined by

$$P_n = 1 - \frac{\sum_{i=1}^{R}(\log rank_i - \log i)}{\log\left(\frac{N!}{(N-R)!R!}\right)}$$

$$R_n = 1 - \frac{\sum_{i=1}^{R}(rank_i - i)}{(N-R)!R!}$$

where N is the number sounds in the dataset, R is the number of relevant sounds, and the rank order of the ith relevant music is denoted by $rank_i$.

1) Effectiveness of Similarity Search: One of our conjectures is that it is possible to obtain effective retrieval from a low-dimensional feature space if the feature vectors are carefully constructed. In this framework, we build relatively small music descriptors from high-dimensional "raw" feature vectors. Furthermore, by incorporating human musical perception, more discriminating information can be incorporated into a smaller size of feature vector which leads to superior performance for similarity search.

The experiments verify our claim. Fig. 2 summaries query effectiveness of the *MARSYAS*, *DWCHs* and *InMAF* techniques for three different query types. It is shown that *MARSYAS* with linear concatenation for feature vector construction is the worst in terms of recall and precision rates. Furthermore, although the *DWCHs* technique achieves better performance than *MARSYAS*, improvement is limited. This is because *DWCHs* only captures low level physical characteristics of the music signal. In fact, the experimental results clearly demonstrate that *InMAF* significantly outperforms the two other approaches. For example, Fig. 2(a) shows that compared to *DWCHs*, the *InMAF* method improves the retrieval precision from 57.5% to 71.6% for Type I queries, 53.8% to 74.2% for Type II queries and 58.7% to 77.2% for Type III queries. In addition, on average, around 39% improvement can be observed against the *MARSYAS* approach for all kinds of query types. From Fig. 2, we also note that integrating additional acoustic features into *InMAF* can bring significant improvement in accuracy for all kinds of query types. For example, by considering pitch, retrieval accuracy improvement for query Type I, Type II, and Type III is 16.7%, 14.3%, and 14.7%, respectively. In contrast, with additional feature incorporation, there is no big improvement in term of query effectiveness when using *MARSYAS*.

D. Robustness

Humans have an impressive capability to identify and classify sound or music, from a very small sample and even in the presence of moderate amounts of distortion. This property is potentially useful in real-world music database applications, where the query sound may have its origins in a process such as low-quality live recording. In order to evaluate the robustness of *InMAF*'s query performance against various audio alterations, we ran the same set of tests as described in Section IV-C for three datasets. However, each music item was distorted before using it in the query and the results were compared against the results obtained from using a non-distorted query. This was repeated for varying levels of distortion. There was no distortion in training data of *InMAF*. Experimental results clearly demonstrate that compared with other approaches, *InMAF* emerges as the most robust technique on all distortion cases. Fig. 3 summarizes the results for the different descriptor generators under various distortions for query type I. Note that the similar observation can be made for other kinds of query. It can bee seen that *InMAF* is fairly robust to different kinds of noise and acoustic distortion with various query types. For example, with query type I, *InMAF* is robust to echo with 9-s delay time, 45-dB SNR white background noise, 9-s cropping, 50% volume amplification, 75% volume deamplification, and 60-dB SNR pink background noise.[10] In contrast, *DWCHs* can only tolerate echo with 10-s delay time, 60-dB SNR white background noise, 11-s cropping, 37% volume amplification, 90% volume deamplification, and pink background noise with SNR 65 dB.

Instead, since *InMAF* is being trained to reduce the dimensionality of raw acoustic feature vectors, this suggests that we

[10] We use equation $SNR_{dB} = 10log_{10}(S/N)$ to calculate signal-to-noise ratio, where S is signal power, N is noise power and its unit is decibles (dB).

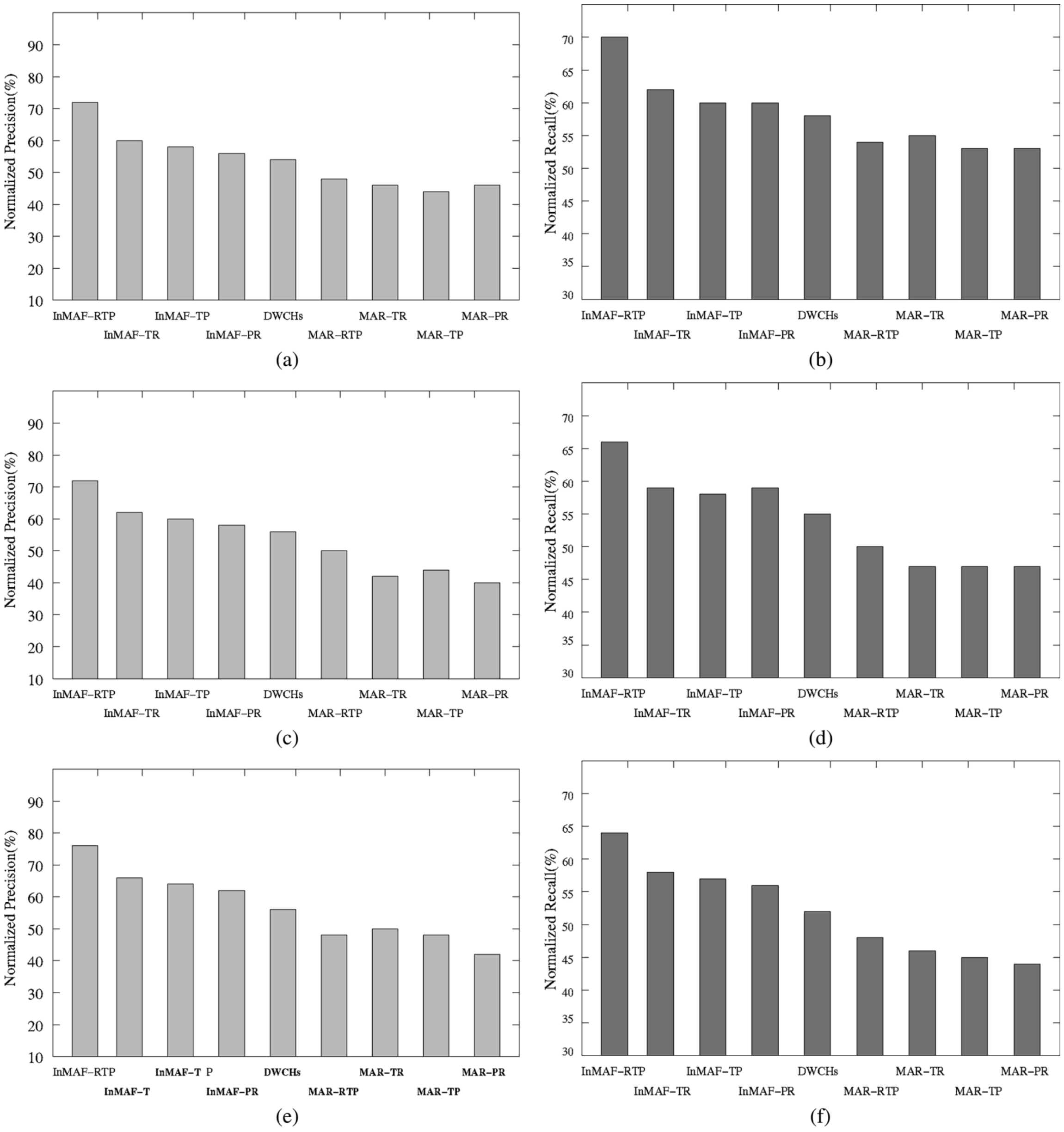

Fig. 2. Query effectiveness of InMAF, DWCHs, and MARSYAS. (a) Precision—query type I. (b) Recall—query type I. (c) Precision—query type II. (d) Recall—query type II. (e) Precision—query type III. (f) Recall—query type III.

can enhance robustness of the framework by training it using not only the original music, but also a copy of the music item which has been altered with noise or distortion. We modified music data items with different kinds of distortion as learning examples for training purpose and carried out a series of experiments to test the performance of our system in the presence of moderate amounts of noise and other kinds of distortion. During this test, we randomly chose 20% of music items from each category in the training data, applied a number of effects to each item, and included all of the distorted versions of the item, as well as the original item, in the training data. The neural network was then trained using all of this data; the aim was to train it to recognize not only exact version of the original music data, but to allow it to be robust to distortions. Fig. 3 shows the effects of extra learning examples on performance improvement for query type I. The results demonstrate that integrating additional distorted examples into training data further improved the robustness of *InMAF* for all the distortion types. In particular, there is a significant gain in the case of pink noise and cropping. For example, with query type I, *InMAF* with extra learning examples is robust to 6 s cropping and 40 dB pink noise. This is a significant improvement over *InMAF* trained by clean data, which can only tolerate 9 s cropping and 60 dB pink noise.

E. Discussion

In this section, we discuss issues related to the performance of this hybrid method in light of the use of PCA as an additional pre-processing step.

1) Effect of PCA on Training Process: In our system, the inputs to the neural network are not the original feature vectors, but vectors which have been reduced via PCA. PCA is used for

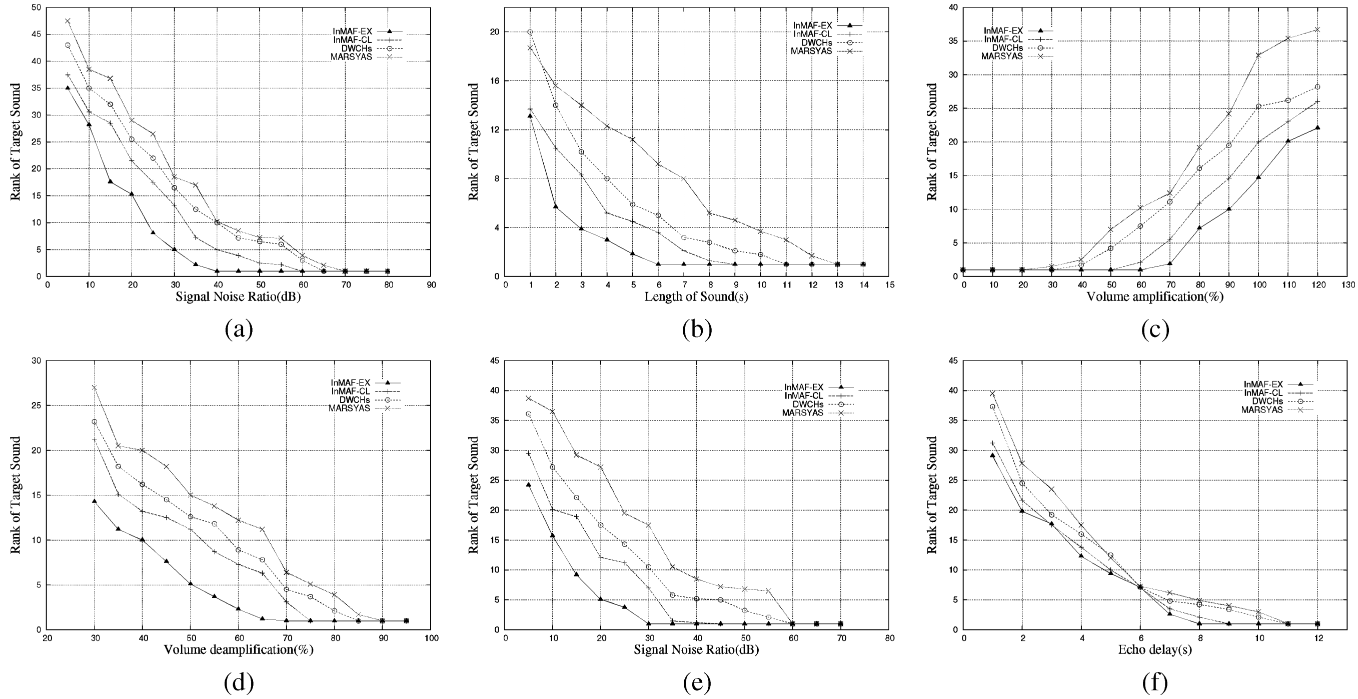

Fig. 3. Robustness of different music descriptor generation methods for query type I. InMAF-CL and InMAF-EX denote InMAF with clean training data or with training examples containing both clean and distorted data individually. (a) Pink noise addition. (b) Cropping. (c) Volume amplification. (d) Volume deamplification. (e) White noise addition. (f) Echo.

TABLE IV
TRAINING TIME OF *InMAF* FOR DIFFERENCE VARIANCE OF PCA PREPROCESSING (THE UNIT OF TRAINING TIME IS min:sec)

Variance of PCA(%)	DataSet I		DataSet II		DataSet III	
	Training Cost	Input Size	Training Cost	Input Size	Training Cost	Input Size
99.9	21:05	44	20:20	45	20:55	43
99.7	20:49	37	20:14	38	20:34	37
99.5	20:42	30	20:02	30	19:57	30
99	21:34	25	20:15	25	21:30	25
97	27:02	21	25:51	21	26:02	21
95	31:51	19	33:55	19	34:02	19
93	51:23	14	49:45	15	50:23	15
88	70:24	11	62:03	11	63:45	12
83	119:33	7	125:27	7	117:47	7

preprocessing to speed up the training time of the neural network component, and also used to remove redundant aspects of the "raw" feature vectors. There is clearly a trade-off involved here: taking many PCs which account for all of the variance ought to result in more effective retrieval/classification performance, but will also result in higher training costs; using less PCs will make training more efficient, but will not account for variance as well and will also result in less effective retrieval/classification performance. In this subsection, we give the results of using different numbers of PCs for three collections of music data. The network training condition is the same as that mentioned in Section IV for 10 hidden units.

From Table IV, it can be seen that the number of PCs for the best network training in our application depends on their total variance. There are no significant differences in the time required for network training using input vectors with size from 25 to 45 since they account for more than 99% of the variance. Moreover, input vectors with variance exceeding 99.7% do not require extra training time. However, if we use PCA-preprocessed feature vectors which acocunt for less than 95% of the variance, then the differences are significant. For example, with dataset I, it takes 1 h and 10 min for input vectors with size of 11 that account for 88% of the variance to complete learning, which is far greater than the time needed for vectors which account for 99% variance or more.

TABLE V
TRAINING TIME OF *InMAF* VERSUS NUMBERS OF HIDDEN UNITS

Size of Hidden Unit	Training Cost (min:sec)		
	DataSet I	DataSet II	DataSet III
20	12:52	12:45	13:05
17	14:45	15:34	14:23
14	18:25	17:01	16:47
10	21:34	20:15	21:30
8	26:50	27:45	26:02
6	51:45	47:51	55:56
4	50:23	57:56	65:34

2) Parameters of the Neural Network: A wide variety of parameter values were tested in order to find an optimal choice for the network learning algorithm in the above experiments. However, in practice, it is often undesirable or even impossible to perform a large number of random parameter tests. Moreover, different applications may require different sets of parameters of the network. In our case, the optimal parameters for the Quick-prop algorithm are step size of 1.75 and learning rate of 0.9.

The number of the hidden units used can also affect the network convergence and learning time. Table V summaries the learning time of neural network with various numbers of hidden units for different datasets. The size of input layer is 25. We can observe that the more hidden units the neural network has, the less training cost is required to complete the learning process. This is because more hidden units can keep more information. However, since the network serves as a dimension reducer, the number of hidden units is restricted to a practical limit.

V. Conclusion

In this paper, we present a novel music descriptor construction technique for effective content-based music retrieval. Unlike previous approaches, which were based solely on automatically derived acoustic features, our approach incorporates (via training) similarity information based on human music perception, to produce descriptors that are both efficient (low-dimensional) and effective (well-discriminating). We are not aware of any other work that integrates notions of human music perception in developing music classification criteria as our work does. We have also developed a learning algorithm for training the system to generate descriptors representing different similarity notions. The approach is fully implemented and a series of experiments has been carried out to compare this method against state-of-art approaches in the areas of classification (using a variety of machine learning approaches), similarity retrieval and robustness against audio distortion. Moreover, our approach can integrate new low-level acoustic features without difficulty.

Acknowledgment

The authors would like to thank Prof. G. Tzanetakis of the University of Victoria, Canada, for kindly sharing his dataset.

References

[1] C. Böhm, S. Berchtold, and D. A. Keim, "Searching in high-dimensional spaces: index structures for improving the performance of multimedia databases," *ACM Comput. Surv.*, vol. 33, no. 3, pp. 322–373, September 2001.

[2] D. Byrd and T. Crawford, "Problems of music information retrieval in the real world," *Inform. Process. Manag.*, vol. 38, no. 2, pp. 249–272, 2001.

[3] C.-C. Chang and C.-J. Lin, LIBSVM: A Library for Support Vector Machines 2001 [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[4] M. Clynes, *Music, Mind and Brain: The Neuropsychology of Music*. New York: Plenum, 1982.

[5] K. Chakrabarti and S. Mehrotra, "Local dimensionality reduction: A new approach to indexing high dimensional spaces," in *Proc. VLDB Conf.*, 2000.

[6] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: Wiley, 2000.

[7] S. Downie and M. Nelson, "Evaluation of a simple and effective music information retrieval method," in *Proc. ACM SIGIR Conf,*, 2000, pp. 73–80.

[8] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ: Prentice-Hall, 1999.

[9] R. Hecht-Nielsen, "Replicator neural networks for universal optimal source coding," *Science*, vol. 269, pp. 1860–1863, 1995.

[10] D. Huron, Humdrum Toolkit Reference Manual 2004 [Online]. Available: http://dactyl.som.ohio-state.edu/Humdrum/guide.toc.html

[11] K. V. Ravi Kanth, D. Agrawal, and A. Singh, "Dimensionality reduction for similarity search in dynamic databases," in *Proc. ACM SIGMOD Conf.*, 1998, pp. 166–176.

[12] S. M. K. Chakrabarti, "The hybrid tree: An index structure for high dimensional feature spaces," in *ICDE*, 1999, pp. 440–447.

[13] J. Karhunen and J. Joutsensalo, "Representation and separation of signals using nonlinear pca type learning," *Neural Netw.*, vol. 7, pp. 113–127, 1994.

[14] J. B. Kruskal and M. Wish, *Multidimensional Scaling*. Thousand Oaks, CA: Sage, 1977.

[15] D. Lee, R. W. Barber, W. Niblack, M. Flickner, J. Hafner, and D. Petkovic, "Indexing for complex queiries on a query-by-content image," in *Proc. SPIE Storage and Retrieval for Image and Video Databases III*, 1993, pp. 24–35.

[16] G. Li and A. A. Khokhar, "Content-based indexing and retrieval of audio data using wavelets," in *Proc. IEEE Int. Conf. Multimedia and Expo (II)*, 2000, pp. 885–888.

[17] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *Proc. Int. Symp. Music Information Retrieval*, 2000.

[18] T. Li, M. Ogihara, and Q. Li, "A comparative study on content-based music genre classification," in *Proc. ACM SIGIR Conf.*, 2003, pp. 282–289.

[19] L. Lu, H. Zhang, and S. Z. Li, "Content-based audio classification and segmentation by using support vector machines," *Multimedia Syst.*, vol. 8, no. 6, pp. 482–492, 2003.

[20] T. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

[21] U. Nam and J. Berger, "Addressing the same but different—different but similar problem in automatic music classification," in *Proc. Int. Symp. Music Information Retrieval*, 2001.

[22] J. Pierce, *The Science of Musical Sound*. New York: W. H. Freeman, 1992.

[23] J. R. Quinlan, *C4.5: Programs for Machine Learning*. New York: Morgan Kaufman, 1993.

[24] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition*. Upper Saddle River, NJ: Prentice-Hall, 1993.

[25] D. Rafiei and A. O. Mendelzon, "Efficient retrieval of similar time sequences using DFT," in *Proc. 5th Int. Conf. Foundations of Data Organization (FODO'98)*, Kobe, Japan, 1998.

[26] E. Selfridge-Field, *Beyond MIDI: The Handbook of Musical Codes*. Cambridge, MA: MIT Press, 1997.

[27] L. Sirovich and M. Kirby, "A low-dimensional procedure for the identification of human faces," *J. Opt. Soc. Amer.*, vol. 4, no. 3, p. 519, 1987.

[28] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York: McGraw-Hill, 1983.

[29] J. Shen, A. H. H. Ngu, and J. Shepherd, "InMAF: indexing music databases via multiple acoustic features," in *Proc. ACM SIGMOD'06*, Chicago, IL, Jun. 2006.

[30] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, Sep. 2002.

[31] T. Tolonen and M. Karjalainen, "A computationally efficient multipitch analysis model," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 4, pp. 708–716, Nov. 2000.

[32] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[33] J. Wiles, P. Bakker, A. Lynton, M. Norris, S. Parkinson, M. Staples, and A. Whiteside, "Using bottlenecks in feedforward networks as a dimension reduction technique—an application to optimization tasks," *Neural Comput.*, vol. 8, no. 6, pp. 1179–1183, 1996.

Jialie Shen is currently pursuing the Ph.D. degree in the School of Computer Science and Engineering, University of New South Wales (UNSW), Sydney, Australia.

He was an Associate Lecturer at UNSW from 2002 to 2005. His main research interests are multimedia databases, information retrieval, and data mining.

John Shepherd received the Ph.D. degree in 1990 from the University of Melbourne, Melbourne, Australia.

He is a Senior Lecturer in the School of Computer Science and Engineering, University of New South Wales, Sydney, Australia. His main research interests are query processing for both relational and nonrelational (e.g., multimedia) databases, information organization/retrieval, and applications of information technology to teaching and learning.

Dr. Shepherd has served on the Program Committees of conferences such as VLDB, WISE, and DASFAA.

Anne H. H. Ngu received the Ph.D. degree from the University of Western Australia in 1990.

She is an Associate Professor in the Computer Science Department, Texas State University, San Marcos, and an Adjunct Associate Professor at the University of New South Wales, Sydney, Australia. Her main research interests are scientific and business process automation, large-scale data access and integration, and multimedia databases. She has worked internationally as a Research Scientist at the Institute of Systems Science (Singapore), Tilburg University (The Netherlands), Telcordia Technologies, Lawrence Livermore National Laboratory (Berkeley, CA), and MCC (Austin, TX) and has authored or coauthored over 80 refereed technical papers.

Dr. Ngu has served as 2002 Publication Chair and as 1999 Organization Chair of the IEEE International Conference on Data Engineering. She has also served as the Program Co-Chair of the 2005 Web Information System Engineering (WISE) Conference. She has been a Program Committee Member for over 20 national and international conferences.