Combining Acoustic and Multilevel Visual Features for Music Genre Classification

MING-JU WU and JYH-SHING R. JANG, National Taiwan University

Most music genre classification approaches extract acoustic features from frames to capture timbre information, leading to the common framework of bag-of-frames analysis. However, time-frequency analysis is also vital for modeling music genres. This article proposes multilevel visual features for extracting spectrogram textures and their temporal variations. A confidence-based late fusion is proposed for combining the acoustic and visual features. The experimental results indicated that the proposed method achieved an accuracy improvement of approximately 14% and 2% in the world's largest benchmark dataset (MASD) and Unique dataset, respectively. In particular, the proposed approach won the Music Information Retrieval Evaluation eXchange (MIREX) music genre classification contests from 2011 to 2013, demonstrating the feasibility and necessity of combining acoustic and visual features for classifying music genres.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Music genre classification

ACM Reference Format:
Ming-Ju Wu and Jyh-Shing R. Jang. 2015. Combining acoustic and multilevel visual features for music genre classification. ACM Trans. Multimedia Comput. Commun. Appl. 12, 1, Article 10 (August 2015), 17 pages.
DOI: http://dx.doi.org/10.1145/2801127

1. INTRODUCTION

With the rapid growth of digital music and online music services (e.g., Spotify, Grooveshark, 7digital, and Pandora), music information retrieval (MIR) has recently emerged as a popular field of research. In particular, music genre classification is becoming more relevant, because genres provide useful descriptions of music [Tzanetakis and Cook 2002]. A key factor in genre classification is the use of effective features for classification. For example, Mel-frequency cepstrum coefficients (MFCCs) [Tzanetakis and Cook 2002], octave-based spectral contrast (OSC) [Jiang et al. 2002], and low-level spectral features [McKay 2010] are the most widely used features based on spectral analysis. However, most approaches pertain to only the spectral characteristics of music.

On the other hand, spectrograms provide effective representations for time-frequency analysis, because they describe the temporal change of energy distribution over frequency bins. Different genres of music exhibit different temporal structures [Grosche et al. 2012]. For instance, Pop songs typically feature a verse and chorus framework, whereas Folk songs exhibit a strophic form [Grosche et al. 2012]. Furthermore, spectrograms provide unique visual texture patterns for various musical instruments [Alm and Walker 2002], and music genres are also closely associated with types of music instrument [Pachet and Cazaly 2000]. Thus, spectrograms can reflect the distinct visual

Authors' addresses: M.-J. Wu, Department of Computer Science, National Tsing Hua University, Hsinchu 30013, Taiwan; email: hsbw@mirlab.org; J.-S. R. Jang, Department of Computer Science and Information Engineering, National Taiwan University, Taipei 10617, Taiwan; email: roger.jang@mirlab.org.

DOI: http://dx.doi.org/10.1145/2801127

texture patterns of various genres. Although much effort has been spent on music genre classification, few studies have focused on extracting features from spectrograms [Costa et al. 2012; Deshpande et al. 2001; Wu et al. 2011].

This article proposes multilevel visual features (MLVFs) for extracting spectrogram textures and their temporal variations. The MLVFs include our previously developed song-level texture features [Wu et al. 2011], and novel beat-level texture and heterogeneity features. The proposed MLVFs based on the time-frequency perspective consider beat tracking, which distinguishes the proposed approach from conventional approaches [Fu et al. 2011]. Because acoustic features are based on spectral analysis and visual features are based on time-frequency analysis, combining both types of feature benefits music genre classification. However, combining acoustic and visual features has rarely been attempted (only the early fusion approach was applied [Wu et al. 2011]). Therefore, a confidence-based late fusion approach is proposed to combine the decisions made by two individual classifiers (based on acoustic and visual features, respectively) to achieve the final prediction.

The remainder of this article is organized as follows. Section 2 describes the related literature, and Section 3 introduces the proposed MLVFs, Section 4 explains the proposed confidence-based late fusion for combing acoustic and visual features. The experimental results and a conclusion to the study are presented in Sections 5 and 6, respectively.

2. RELATED LITERATURE

Feature extraction is the basis of music genre classification [Tzanetakis and Cook 2002], which can be divided into three categories according to the temporal resolution.

(1) *Frame-Level.* Frame-level features are typically obtained from analysis windows of 10 to 100 ms frames, which can capture local spectral characteristics. Some of the commonly used frame-level features are MFCCs [Tzanetakis and Cook 2002], OSC [Jiang et al. 2002], spectral centroid, spectral rolloff, spectral flux [Fu et al. 2011], etc.
(2) *Segment-Level.* Because a segment is long enough to capture the sound texture, it is also referred to as a texture window [Tzanetakis and Cook 2002]. These features are typically obtained from statistical measures of a segment composed of several frames. For example, the winners of MIREX 2010 genre classification contest, Seyerlehner et al. [2010], proposed a set of block-level features for the Cent-scale spectrum [Goto 2003] by using various statistical operations, such as percentile and variance.
(3) *Song-Level.* Song-level (or clip-level) features tend to capture global characteristics of music. Costa et al. [2012] proposed using local binary patterns [Ojala et al. 2002] as visual features for music genre classification. Features were extracted from three 10-s segments from the beginning, middle, and end of each original song. Independent of Costa et al. [2012], Wu et al. [2011] also proposed song-level texture features based on the Gabor filter bank.

Some approaches use frame or segment-level features to generate other effective features. For example, Cao and Li [2009] (MIREX 2009 genre classification contest winners) applied the Gaussian super vector (GSV) [Campbell et al. 2006] to music genre classification. A Gaussian mixture model (GMM) is applied to train a universal background model (UBM) [Reynolds et al. 2000] to capture the global timbre characteristics of the frame-level features. The GSV from each music clip is then derived from a song-specific GMM by using the maximum a posterior (MAP) adaptation [Gauvain

Table I. Comparison of Approaches to Music Genre Classification

Aspects of comparison	Previous approaches	Proposed method
Temporal resolution of features	Fixed[a]	Dynamic[d]
Analysis unit for temporal variation	All of the music[b]	Inter-beat intervals[e]
Method for combining acoustic and visual features	Early fusion[c]	Confidence-based late fusion

[a][Costa et al. 2012; Jiang et al. 2002; Seyerlehner 2010; Seyerlehner et al. 2010; Tzanetakis and Cook 2002]
[b][Lee et al. 2009]
[c][Wu et al. 2011]
[d]Based on beat-level texture features.
[e]Based on beat-level heterogeneity features.

and Lee 1994] from the UBM. Because of the effectiveness of the GSV, it was applied to represent the acoustic features in this study.[1]

However, the GSV is a bag-of-frames approach, which is inadequate for modeling temporal variation. Approaches have been proposed for describing the temporal evolutions of music. The multivariate autoregressive model was employed to estimate temporal dependencies between frames by using an affine prediction scheme [Meng et al. 2007]. Modulation spectral analysis can be applied to reveal music trends. For example, Lee et al. [2009] proposed the modulation spectral contrast (MSC) and modulation spectral valley (MSV) to analyze the temporal variation of music. However, relatively little research has investigated using temporal analysis to classify music genres.

Moreover, using one type of feature may be inadequate to achieve optimal results. Consequently, fusion approaches can be employed to combine multiple types of features. For early fusion, multiple types of features can be directly concatenated to form a new feature vector before classification. In late fusion, the fusion is performed after classification. The majority vote rule, kernel-based late fusion, and probability-based late fusion are widely used late fusion strategies. According to the majority vote rule, the class (genre) that receives the most votes from various classifiers is selected as the prediction. During kernel-based late fusion, multiple kernels can be integrated into one kernel. For example, a convolution kernel and product probability kernel can be applied for such purposes [Meng and Shawe-Taylor 2005]. The problem in probability-based late fusion is to design a global measure that combines the probability of each class from various classifiers. For example, two individual support vector machine (SVM) classifiers can be trained using two types of features, and the SVM can estimate the posterior probability of each class (genre) [Wu et al. 2004]. Several strategies, such as Max, Product, and Sum rules, can be employed to fuse the probabilities returned by all classifiers [Costa et al. 2012; Kittler et al. 1998]. The class with the maximal value is then selected as the prediction. However, designing optimal late fusion strategies remains challenging.

Table I shows a comparison of the proposed method with previous approaches used to classify music genres. Conventional approaches for music genre classification have generally involved frame-level, segment-level, and song-level features with fixed temporal resolutions. By contrast, in the proposed approach, the temporal resolution adopted by beat-level texture features is dynamic instead of fixed. More specifically, beat-level texture features are a type of segment-level feature, wherein the segment size is dynamically determined based on the music content. For example, a piece of music with a fast tempo leads to shorter inter-beat interval (IBI) segments. Because IBIs are likely

[1]For more details on the GSV, please refer to Chen et al. [2011].

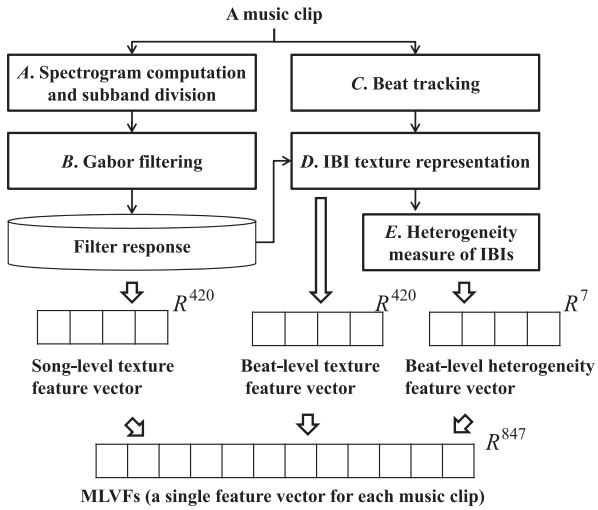

Fig. 1. MLVF flowchart.

to be the unit perceived by listeners, it is natural to use IBI-based features for genre classification. Thus far, only a few studies have addressed beat-level features. Ellis and Poliner [2007] used the beat-level MFCC and beat-level chroma features to assess music similarity, whereas Pei and Hsu [2009] identified musical instruments in polyphonic music using the beat-level MFCC and beat-level MPEG-7 features. However, no studies have investigated the visual features at the beat level. In addition, the proposed beat-level heterogeneity features measure the temporal variation of IBIs contained in music rather than measuring all aspects of music. The proposed confidence-based late fusion is also a novel attempt for combining acoustic and visual features.

3. PROPOSED MULTILEVEL VISUAL FEATURES

This section describes the proposed MLVFs, which comprise texture and heterogeneity features, as shown in Figure 1. The texture features are used to represent the texture of a spectrogram from a global and local perspective (i.e., song-level and beat-level texture features). The beat-level heterogeneity features are used to represent the texture variation of IBIs, which can also be considered a measure of temporal variation. Finally, the texture and heterogeneity feature vectors are concatenated to form the MLVFs.

3.1. Song-Level Texture Features

A spectrogram for each music clip is computed using a short-time Fourier transform (STFT) with a window size of 1024 samples (or 46.4 ms for a sampling rate of 22 050 Hz) and a half overlap. Each point in the spectrogram represents the log-scale energy at a particular time and frequency. To make the spectrogram easier to observe, the log-scale energy is quantized to an intensity of 256 gray levels based on linear mapping.

Human perceptions of music are based on a logarithmic frequency scale, and notes separated by an octave are perceived as harmonically equivalent [Muller et al. 2011]. Hence, octave-based sub-bands [Jiang et al. 2002] should be considered. Thus, the spectrogram is divided into seven sub-bands, $\mathbf{S}_i$, according to the following octave-based sub-bands: 0 to 200 Hz, 200 to 400 Hz, 400 to 800 Hz, 800 to 1600 Hz, 1600 to 3200 Hz, 3200 to 8000 Hz, and 8000 to 11025 Hz.

A two-dimensional Gabor filter in the spatial domain exhibits the following general form:

$$\psi_{\lambda,\theta}(x, y) = \exp\left(-\frac{x'^2 + y'^2}{2\sigma^2}\right)\exp\left(\frac{j2\pi x'}{\lambda}\right), \tag{1}$$

where

$$\begin{cases} x' = x\cos\theta + y\sin\theta \\ y' = -x\cos\theta + y\sin\theta \end{cases}. \tag{2}$$

In Eq. (1), λ is the wavelength, which is inversely related to the frequency. Thus, a higher wavelength corresponds to a lower frequency, causing the Gabor filter to generate a stronger response to the slowly varying components of an image. The variable θ is the rotation degree that controls the orientation selectivity of the filter. σ is the standard deviation of the Gaussian function, which is set to 0.5λ in this study. The same settings for the Gabor filter bank are applied that were used in Wu et al. [2011], with $\lambda \in \{2.5, 5, 7.5, 10, 12.5\}$ and $\theta \in \{0^\circ, 30^\circ, 60^\circ, 90^\circ, 120^\circ, 150^\circ\}$. The magnitude of the filter response is then obtained by convolving $\mathbf{S}_i$ with a Gabor filter,

$$\mathbf{R}_{i,\lambda,\theta} = \left|\mathbf{S}_i * \psi_{\lambda,\theta}\right|, \tag{3}$$

where $\mathbf{R}_{i,\lambda,\theta}$ represents the magnitude of the filter response for a particular $\mathbf{S}_i$ with specific λ and θ. The average and standard deviation of all elements in $\mathbf{R}_{i,\lambda,\theta}$ can be used to represent the global texture features of a spectrogram. The song-level texture feature vector is represented as follows:

$$\mathbf{f}_{song} = \left[\mu_{\mathbf{R}_{1,1,1}}, \ldots, \mu_{\mathbf{R}_{i,\lambda,\theta}}, \sigma_{\mathbf{R}_{1,1,1}}, \ldots, \sigma_{\mathbf{R}_{i,\lambda,\theta}}\right]. \tag{4}$$

3.2. Beat-Level Texture Features

Each IBI texture can be represented by a set of local texture descriptors. Let $\mathbf{B}_{i,\lambda,\theta,k}$ represents the segment between $\mathbf{t}_k$ and $\mathbf{t}_{k+1}$ in $\mathbf{R}_{i,\lambda,\theta}$, where $\mathbf{t}_k$ is a beat instance determined by the beat tracker [Ellis 2007]. Let $\mu_{\mathbf{B}_{i,\lambda,\theta,k}}$ denote a local texture descriptor, where $\mu_{\mathbf{B}_{i,\lambda,\theta,k}}$ is the average of $\mathbf{B}_{i,\lambda,\theta,k}$. The kth IBI texture in $\mathbf{S}_i$, $\mathbf{I}_{i,k}$, can be expressed with all combinations of λ and θ.

$$\mathbf{I}_{i,k} = \left[\mu_{\mathbf{B}_{i,1,1,k}}, \mu_{\mathbf{B}_{i,1,2,k}}, \ldots, \mu_{\mathbf{B}_{i,2,1,k}}, \mu_{\mathbf{B}_{i,2,2,k}}, \ldots, \mu_{\mathbf{B}_{i,\lambda,\theta,k}}\right]^T. \tag{5}$$

To visualize variation of local texture descriptors, i, λ, and θ are fixed and k varied.

$$\mathbf{L}_{i,\lambda,\theta} = \left[\mu_{\mathbf{B}_{i,\lambda,\theta,1}}, \mu_{\mathbf{B}_{i,\lambda,\theta,2}}, \ldots, \mu_{\mathbf{B}_{i,\lambda,\theta,k}}\right]. \tag{6}$$

Figure 2(a) shows a spectrogram in $\mathbf{S}_4$ of a hip-hop music clip. In Figure 2(b), the black vertical lines represent beats. The bottom annotations represent the IBI indices, and the top annotations represent the IBI durations (in frames). The figure shows that $\mathbf{L}_{4,5,0^\circ}$ correlates with the vertical components, and $\mathbf{L}_{4,5,90^\circ}$ correlates with the horizontal components. This shows the effectiveness of $\mathbf{L}_{i,\lambda,\theta}$ at describing textures. Then beat-level texture features are the average and standard deviations of $\mathbf{L}_{i,\lambda,\theta}$.

$$\mathbf{f}_{beat} = \left[\mu_{\mathbf{L}_{1,1,1}}, \ldots, \mu_{\mathbf{L}_{i,\lambda,\theta}}, \sigma_{\mathbf{L}_{1,1,1}}, \ldots, \sigma_{\mathbf{L}_{i,\lambda,\theta}}\right]. \tag{7}$$

3.3. Beat-Level Heterogeneity Features

Music from different genres is likely to exhibit different degrees of temporal variation. To take advantage of this characteristic, a self-distance matrix (SDM) [Paulus et al. 2010] is defined to reflect the heterogeneity measure of IBIs. Each element (x, y) in

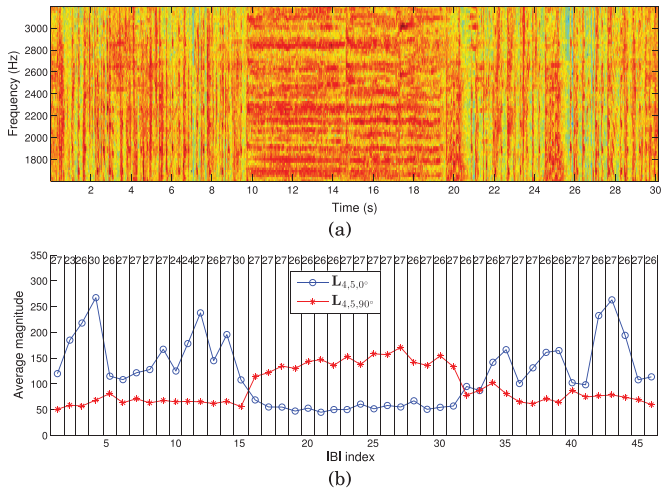

Fig. 2. (a) The spectrogram in $\mathbf{S}_4$ of a hip-hop clip. (b) Each IBI is divided by beat locations (black vertical lines). The bottom annotations indicate the IBI indices and the top annotations indicate the IBI durations (in frames). $\mathbf{L}_{4,5,0^\circ}$ correlates with the vertical components in the spectrogram, and $\mathbf{L}_{4,5,90^\circ}$ correlates with the horizontal components.

$\mathbf{SDM}_i$ represents the correlation-based distance between $\mathbf{I}_{i,x}$ and $\mathbf{I}_{i,y}$.

$$\mathbf{SDM}_i(x,y) = 1 - \frac{(\mathbf{I}_{i,x} - \bar{\mathbf{I}}_{i,x})^T(\mathbf{I}_{i,y} - \bar{\mathbf{I}}_{i,y})}{|\mathbf{I}_{i,x} - \bar{\mathbf{I}}_{i,x}||\mathbf{I}_{i,y} - \bar{\mathbf{I}}_{i,y}|}. \tag{8}$$

A greater $\mathbf{SDM}_i(x,y)$ reflects greater dissimilarity between $\mathbf{I}_{i,x}$ and $\mathbf{I}_{i,y}$. A substantial difference between two adjacent elements in the SDM indicates a sharp temporal variation between two IBIs.

Because the gradient is a vector pointing in the direction of greatest change for each element in the SDM, it can be used to indicate temporal variation. Specifically, the magnitude of the gradient $\mathbf{M}_i$ represents the changing rate, and the angle of the gradient $\mathbf{A}_i$ is the direction of that change.

$$\nabla\mathbf{SDM}_i = \left[\frac{\partial\mathbf{SDM}_i}{\partial x}, \frac{\partial\mathbf{SDM}_i}{\partial y}\right] \tag{9}$$

$$\mathbf{M}_i = \|\nabla\mathbf{SDM}_i\| \tag{10}$$

$$\mathbf{A}_i = \tan^{-1}\left(\frac{\partial\mathbf{SDM}_i/\partial y}{\partial\mathbf{SDM}_i/\partial x}\right). \tag{11}$$

To identify neighbors with high absolute gradients in $\mathbf{SDM}_i$, the following two steps are performed for each element (x,y) in $\mathbf{SDM}_i$:

Step (1). If $\mathbf{M}_i(x,y) \geq \rho$, go to Step (2).
Step (2). Select a neighbor (x_g, y_g) according to the direction of $\mathbf{A}_i(x,y)$. Set $\mathbf{V}_i(x_g, y_g)$ to 1,

where ρ denotes the threshold of $\mathbf{M}_i$ and $\mathbf{V}_i$ denotes a Boolean matrix with all zero elements. When $\mathbf{M}_i(x,y) \geq \rho$, this indicates that the distance rapidly changes from (x,y) to (x_g, y_g), hence (x_g, y_g) is recorded by setting $\mathbf{V}_i(x_g, y_g)$ to 1.

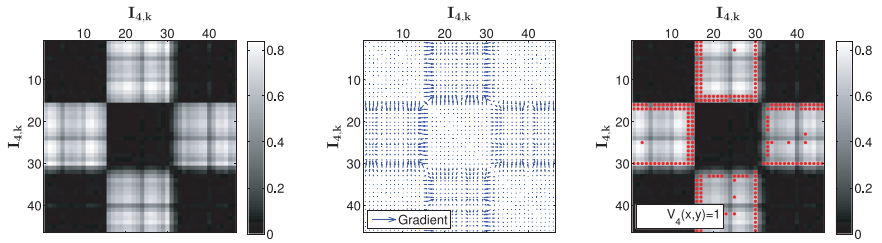

Fig. 3. (a) Example of an SDM obtained from $\mathbf{S}_4$ of the same music clip used in Figure 2. (b) Gradient computed for the SDM. (c) $\mathbf{V}_4$ indicate the distances that rapidly change, which correlate to the variation of IBIs.

Figure 3(a) shows an example of SDM obtained from $\mathbf{S}_4$ of the music clip used in Figure 2. The darker block in the SDM represents a homogenous segment [Paulus et al. 2010] with similar music content. The corner of the darker block along the main diagonal indicates the novelty point, which corresponds to the transition of two music segments [Paulus et al. 2010]. Figure 3(b) shows the gradient of the SDM. Figure 3(c) shows $\mathbf{V}_4$ with $\rho = 0.17$. The red points tend to be located on the boundaries of blocks, because these points indicate a rapid change in distance (which is associated with IBI variation). Consequently, more red points in an SDM reflect a greater heterogeneity measure.

The heterogeneity measure, h_i, for $\mathbf{SDM}_i$ is defined by

$$h_i = \sum_{x=1}^{n} \sum_{y=x+1}^{n} \frac{\mathbf{V}_i(x,y)\mathbf{SDM}_i(x,y)}{|x-y|}, \tag{12}$$

where h_i is the weighted summation of distances that rapidly change, and the weighting is divided by $|x-y|$ to reduce the measure if the two IBIs ($\mathbf{I}_{i,x}$ and $\mathbf{I}_{i,y}$) are far apart. Only the upper triangle of the SDM should be considered because it is symmetric.

Beat-level heterogeneity features exhibit the following form;

$$\mathbf{f}_{heterogeneity} = [h_1, \ldots, h_i], \tag{13}$$

where h_i denotes the heterogeneity measure of $\mathbf{S}_i$, as in Eq. (12). A higher h_i value indicates that stronger temporal variation occurred at the ith sub-band in a spectrogram.

4. PROPOSED CONFIDENCE-BASED LATE FUSION

To perform the proposed confidence-based late fusion, two quantities from SVMs are measured. Figure 4 shows the flowchart of this process. The prediction of the multiclass SVM is based on the one-against-one approach. If the predicted classes of the two multiclass SVMs, ω_{GSV} and ω_{MLVF}, are different, then the confidence measures of the pair of the binary-class SVMs (corresponding to classes $\{\omega_{GSV}, \omega_{MLVF}\}$), c_{GSV} and c_{MLVF}, are computed and compared to complete the final prediction. In other words, the final prediction is taken from the binary classifier with a higher confidence measure. Because different types of feature may exhibit different discriminative powers for a given music clip, confidence-based late fusion selects a presumably more accurate prediction. The following section describes the basic concept of the SVM and how to compute its confidence measure from two confidence factors.

The goal of a binary-class SVM is to identify the hyperplane (i.e., decision boundary) with the widest separation between two classes of training data, which can be

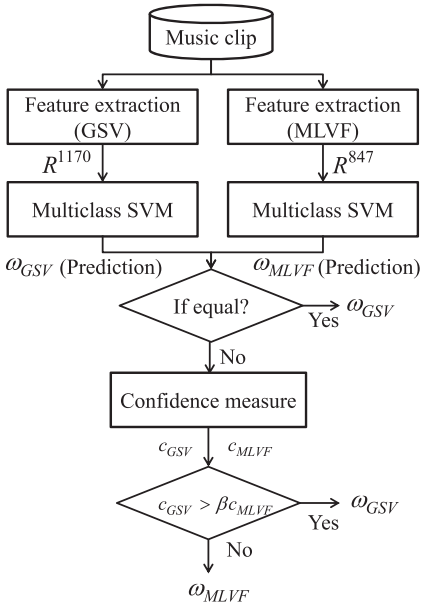

Fig. 4. Flowchart of the proposed confidence-based late fusion.

expressed as

$$\mathrm{g}(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b = \sum_{i=1}^{l} \lambda_i y_i \mathbf{x}_i^T \mathbf{x} + b, \tag{14}$$

where $\mathbf{x}$ is the feature vector of the test instance; $\mathbf{w}$ is a normal vector; b is the bias term in the hyperplane; $\mathbf{x}_i$ is a d-dimensional feature vector of training instances; y_i is the label (ground truth) of $\mathbf{x}_i$, which is set at either 1 or -1 to distinguish between the two classes; l is the number of music clips in the training set; λ_i is the Lagrange multiplier, which can be either zero or positive. Specifically, the optimal hyperplane is the linear combination of $\mathbf{x}_i$ with $\lambda_i > 0$. These $\mathbf{x}_i$ are support vectors, which support the maximum-margin and create the optimal hyperplane. Predicted class ω of test instance $\mathbf{x}$ is either 1 or -1, depending on whether the sign of $\mathrm{g}(\mathbf{x})$ is positive or negative.

To facilitate data separation, a linear mapping ϕ is applied to transform feature vector $\mathbf{x}_i$ into a new space with high dimensionality. According to the kernel trick, the inner product in the high-dimensional space can be expressed as kernel function K in the original space. The optimal hyperplane can then be expressed as

$$\mathrm{g}(\mathbf{x}) = \sum_{i=1}^{l} \lambda_i y_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) + b = \sum_{i=1}^{l} \lambda_i y_i \, \mathrm{K}(\mathbf{x}_i, \mathbf{x}) + b. \tag{15}$$

In this study, the widely used radial basis function (RBF) kernel is applied.

$$\mathrm{K}(\mathbf{x}_i, \mathbf{x}) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}\|^2}{2\sigma^2}\right). \tag{16}$$

Because the corresponding linear mapping ϕ transforms data to the Hilbert space (i.e., a vector space with infinite dimensions) for classification, two confidence factors in the Hilbert space are proposed.

(1) *Confidence Factor 1. The Distance Between the Test Instance and the Hyperplane in the Hilbert Space*. The goal of an SVM is to identify the hyperplane with the maximal margin between two classes of training data. Consequently, the prediction of the test instance is likely to be correct if the instance is far from the hyperplane. The distance between the test instance and the hyperplane can be expressed as

$$\frac{\left|g(\mathbf{x})\right|}{\|\mathbf{w}\|}. \tag{17}$$

To allow this distance to be directly comparable, Eq. (17) is normalized by dividing it by the half margin (the distance between support vectors and the hyperplane in the Hilbert space). This normalized distance cf_1 is then used as the first confidence factor:

$$cf_1 = \frac{\frac{\left|g(\mathbf{x})\right|}{\|\mathbf{w}\|}}{\frac{1}{\|\mathbf{w}\|}} = |g(\mathbf{x})|. \tag{18}$$

When $cf_1 < 1$, the test instance is inside the margin. When $cf_1 = 1$, the test instance is on the margin. When $cf_1 > 1$, the test instance is outside the margin. Consequently, a greater cf_1 tends to reflect higher confidence.

(2) *Confidence Factor 2. The Distance between the Test Instance and Its Nearest Neighbor in the Hilbert Space.* As demonstrated in Eq. (15), a linear mapping ϕ transforms data to a new space with high dimensions. The relationship between training data $\mathbf{x}_i$ and test instance $\mathbf{x}$ in the new space should also be considered. The distance between $\phi(\mathbf{x})$ and $\phi(\mathbf{x}_i)$ in the Hilbert space can be computed in the original space by using the kernel trick.

$$\begin{aligned} \|\phi(\mathbf{x}_i) - \phi(\mathbf{x})\|^2 &= (\phi(\mathbf{x}_i) - \phi(\mathbf{x}))^T(\phi(\mathbf{x}_i) - \phi(\mathbf{x})) \\ &= \langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_i)\rangle - 2\langle\phi(\mathbf{x}_i), \phi(\mathbf{x})\rangle + \langle\phi(\mathbf{x}), \phi(\mathbf{x})\rangle \\ &= \mathrm{K}(\mathbf{x}_i, \mathbf{x}_i) - 2\,\mathrm{K}(\mathbf{x}_i, \mathbf{x}) + \mathrm{K}(\mathbf{x}, \mathbf{x}). \end{aligned} \tag{19}$$

According to Eq. (16) and Eq. (19), the second confidence factor can be expressed as

$$cf_2 = \min_{i,\text{with } \mathbf{x}_i \text{ in class } \omega} \left\{2 - 2\exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}\|^2}{2\sigma^2}\right)\right\}. \tag{20}$$

That is, cf_2 computes the minimal distance between $\phi(\mathbf{x})$ and $\phi(\mathbf{x}_i)$, where $\mathbf{x}_i$ are the training instances of the same class as predicted class ω. A lower cf_2 indicates a higher similarity between $\phi(\mathbf{x})$ and $\phi(\mathbf{x}_i)$, and thus ω should be more convincing. The confidence measures c_{GSV} and c_{MLVF} are then defined as follows:

$$\begin{cases} c_{GSV} = \frac{cf_{1\,(GSV)}}{cf_{2\,(GSV)}} \\ c_{MLVF} = \frac{cf_{1\,(MLVF)}}{cf_{2\,(MLVF)}} \end{cases}. \tag{21}$$

Therefore, a greater cf_1 and a smaller cf_2 lead to a higher confidence. The final decision can be determined according to

$$\text{Apply } \omega_{GSV} \ (\omega_{MLVF}) \text{ if } c_{GSV} > (\leq)\ \beta c_{MLVF}, \tag{22}$$

where β represents the weighting for adjusting the importance of c_{GSV} and c_{MLVF}. When c_{GSV} is larger than βc_{MLVF}, ω_{GSV} is applied as the final decision. Otherwise, ω_{MLVF} is applied.

5. EXPERIMENTAL RESULTS

This section describes the datasets, experimental settings, and experimental results.

5.1. Datasets

Three datasets are used in this study.

(1) *Universal Background Model (UBM) Music Dataset.* This dataset is used for training a robust UBM. Because the UBM dataset should be as diverse as possible [Chen et al. 2011], 2000 music clips (previews) were randomly selected from 7digital, a database of more than 25 000 000 songs from various genres, artists, and music styles.[2]
(2) *GTZAN Dataset [Tzanetakis and Cook 2002].* This dataset is the public benchmark dataset most used in the literature. It contains 1000 clips equally distributed over 10 genre classes, namely Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Pop, Reggae, and Rock. In this dataset, leave-one-out cross validation is adopted because it can provide an unbiased accuracy estimate.
(3) *Unique Dataset [Seyerlehner 2010].* This dataset contains 3115 clips from 3115 unique artists spanning 14 genres: Blues (41), Country (58), Dance (766), Electronic (187), Hip-hop (229), Jazz (310), Classical (744), Reggae (74), Rock (398), Pop (59), Soul or Rhythm and Blues (39), Folk (38), World (146) and Spoken Word (26). The duration of each clip is approximately 30 s. Here leave-one-out cross validation is also applied as the performance index.
(4) *MSD Allmusic Style Dataset (MASD) [Schindler et al. 2012].* This dataset contains 273 936 clips encompassing 25 genres, and is a benchmark dataset of the MSD (Million Song Dataset) [Bertin-Mahieux et al. 2011]. Schindler et al. [2012] constructed this dataset to provide a large-scale comprehensive music genre dataset that afforded researchers a realistic environment in which to test their systems. The dataset genres include Big Band (3115), Contemporary Blues (6874), Traditional Country (11 164), Dance (15 114), Electronica (10 987), Experimental (12 139), Folk International (9849), Gospel (6974), Emo Grunge (6256), Hip Hop Rap (16 100), Classic Jazz (10 024), Alternative Metal (14 009), Death Metal (9851), Heavy Metal (10 784), Contemporary Pop (13 624), Indie Pop (18 138), Latin Pop (7699), Punk (9610), Reggae (5232), RnB Soul (6238), Alternative Rock (12 717), College Rock (16 575), Contemporary Rock (16 530), Hard Rock (13 276), and Neo-Psychedelia Rock (11 057). The duration of the clips is typically 30 or 60 s.

In the current study, we apply the same stratified split used in Schindler et al. [2012]; 2/3 of the data are used for training and 1/3 is employed for testing; artist, album, and time filters are applied for both training and test sets. In other words, the split prevents the same artist and album from appearing in both the training and test sets. In addition, all music clips in the training set are released earlier than the music clips in the test set. Because copyright laws prevent redistributing the music clips, the dataset cannot provide audio files. Consequently, we downloaded the audio files from 7digital according to the provided track IDs. Because some files from 7digital were corrupted or unavailable, we obtained only 98.65% of the dataset. In particular, 1.36% and 1.35% of the music clips were unavailable in the training and test sets, respectively.[3]

[2]http://www.7digital.com/.
[3]The missing tracks are listed at http://mirlab.org/users/brian.wu/genreClassification/missing_tracks.txt.

Table II. Feature Comparison

Features	Feature dimension	GTZAN	Unique	MASD
Beat-level heterogeneity features	7	33.50%	42.41%	11.68%
Song-level texture features	420	84.30%	75.31%	40.00%
Beat-level texture features	420	82.80%	74.64%	38.08%
MLVFs	847	85.70%	75.67%	40.28%

5.2. Experimental Settings

The music clips were converted to a sampling rate of 22 050 Hz with 16-bit resolution in all the datasets. The well-known SVM tool, LIBSVM [Chang and Lin 2010], with a RBF kernel is applied as a classifier. The cost value, C, of the SVM is set to 3. For normalization, the MLVFs are normalized to a zero mean and unit variance. For the parameters, ρ is empirically set to 0.05. The GSV parameters are established according to the settings used in Wu et al. [2011], in which the MFCC dimension is 39 and the number of mixture components is 30. Therefore, the GSV dimension is 1170 ($39 \times 30 = 1170$). When the early fusion for the MLVFs and GSV is performed, the new feature vector totals 2017 dimensions ($1170 + 847 = 2017$).

5.3. Visual Feature Comparison

To examine the performance of the MLVFs, Table II shows a comparison of the MLVFs with various visual features. Features are directly input into the SVM for classification. For the beat-level heterogeneity features, the maximal accuracy is approximately 42.41%, despite its low dimensionality of 7. This implies that temporal variation is important to music genre classification. The accuracies of the beat-level texture features are different from, but comparable to, the accuracies of the song-level texture features. This indicates that the local textures are also informative. Moreover, the MLVFs achieve the most favorable performance of all the visual feature combinations.

In order to demonstrate the statistical significance of difference among visual features, we used the Friedman test [Demšar 2006] with $\alpha = 0.05$ to evaluate the differences among beat-level features, song-level texture features, and MLVFs, where the Friedman test is a statistical test for the comparison of multiple methods over multiple datasets. The p value of our experiment is 0.0498, which is less than 0.05, indicating the differences among these three sets of visual features are statistically significant.

Admittedly, the improvement of MLVFs over song-level texture features is not impressive in terms of the recognition rate. However, if we look at the error reduction rate (which is commonly used in speech recognition), the improvement becomes much more significant. More specifically, the error reduction rate is 8.92% for the GTZAN dataset, 1.46% for the Unique dataset, and 0.47% for MASD.

We can also evidence the advantage of MLVFs over song-level texture features using visualization by projecting features onto 2D plane via linear discriminant analysis (LDA). As shown in Figure 5, the projection of MLVFs are more separable than song-level texture features. Consequently, it makes sense to have MLVFs as the final visual features.

5.4. Performance Evaluation

Figure 6 shows the proposed confidence-based late fusion using various β for three datasets. As can be seen, we tend to obtain superior results when $\beta \geq 1$, indicating visual features are more critical than acoustic features. For the GTZAN dataset, the accuracy is 88.40% when $\beta = 1$, and the best accuracy is 88.60% when $\beta = 1.35$. For the Unique dataset, the best accuracy is 77.66% when $\beta = 1$. For the MASD, the accuracy is 41.45% when $\beta = 1$, and the best accuracy is 41.76% when $\beta = 1.9$. Consequently,

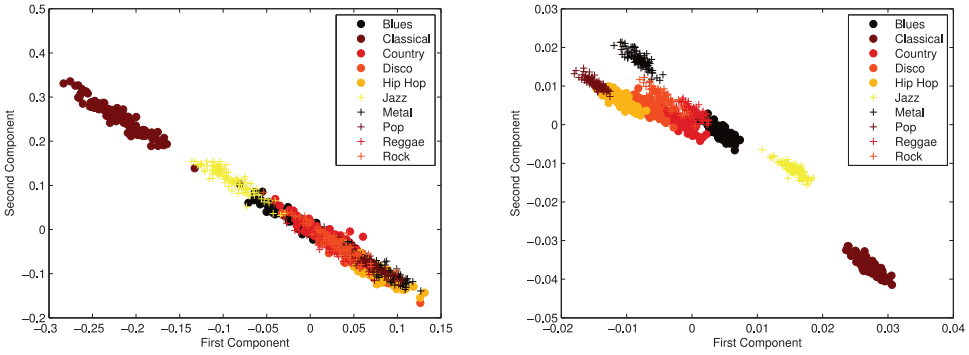

Fig. 5. Visualization via LDA projection to 2D space for the GTZAN dataset using (a) song-level visual features and (b) MLVFs.

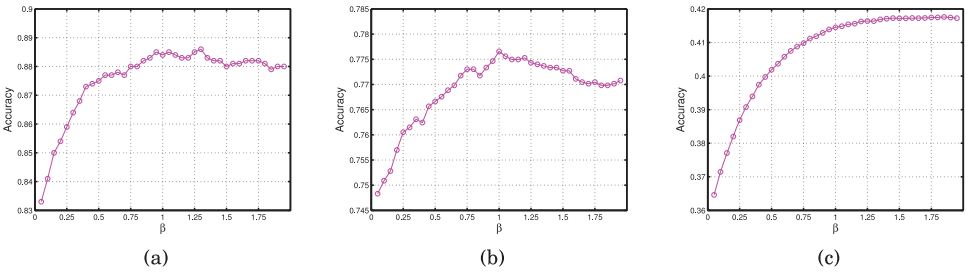

Fig. 6. The proposed confidence-based late fusion using various β. (a) GTZAN dataset. (b) Unique dataset. (c) MASD.

Table III. Comparison of the Fusion (Two Feature Types) and Nonfusion (One Feature Type) Methods

Method	GTZAN	Unique	MASD
MLVF+GSV (confidence-based late fusion)	**88.60%**	**77.66%**	41.76%
MLVF+GSV (probability-based late fusion by the *max* rule)	87.70%	76.98%	40.60%
MLVF+GSV (probability-based late fusion by the *sum* rule)	88.10%	77.14%	41.49%
MLVF+GSV (probability-based late fusion by the *prod* rule)	88.20%	77.43%	41.53%
MLVF+GSV (early fusion)	87.00%	77.50%	**41.90%**
MLVF	85.70%	75.67%	40.28%
GSV	80.10%	73.58%	34.73%

$\beta = 1$ can usually achieve comparable results when compared with the optimal value of β. As a result, we can set $\beta = 1$ as the default value for unknown datasets.

To validate whether using both acoustic and visual features outperforms the recognition rates when only one type of feature is used, the fusion (two types of feature) and nonfusion methods (one type of feature) are compared. Table III illustrates the comparison, in which the fusion methods outperform the nonfusion method. Because MLVFs and GSVs are used, both spectral and time-frequency aspects are utilized, considerably increasing the discriminating power of the features. This is vital to the success of music genre classification.

Table III shows that the proposed confidence-based late fusion is superior to the probability-based late fusion for three datasets. To further validate the statistical significance, we applied the Friedman test [Demšar 2006] with $\alpha = 0.05$ for all late fusion

approaches (optimal β is applied for the confidence-based late fusion) to obtain the p value of 0.0293 which is smaller than 0.05. Although the proposed confidence-based late fusion can achieve results that are only comparable to those of early fusion, early fusion increases the dimensionality of feature space, whereas confidence-based late fusion does not. Because the memory requirement is a crucial concern when training a classifier using a large-scale dataset, the proposed confidence-based late fusion is more applicable for large-scale datasets than is early fusion.

5.5. Comparison with Other Approaches

Table IV shows a comparison of various approaches on various datasets. Bergstra et al. [2010] proposed a set of spectral features. Seyerlehner et al. [2010] also used block-level features to capture spectral characteristics. Tsunoo et al. [2011] developed an approach to identify rhythmic and bass-line patterns. Ren and Jang [2012] applied time-constrained sequential pattern mining to discover genre-specific patterns. Panagakis et al. [2010] proposed dimensionality reduction methods for auditory temporal modulations, and Panagakis et al. [2014] proposed a joint sparse low-rank representation. Yeh et al. [2013] proposed a dual-layer bag-of-frames feature representation.

For the GTZAN dataset, the proposed method is superior to other approaches except for the approach of Panagakis et al. [2014]. However, the proposed method outperforms Panagakis et al. [2014] for the Unique dataset. This indicates our approach is comparable to Panagakis et al. [2014]. For the MASD, the proposed method achieves an accuracy level of 41.76%. Notably, there are 1246 (1.35%) unavailable music clips in the test set. To ensure a fair comparison with other approaches, the proposed system can be assumed to be unable to recognize all 1246 music clips, in which case, the accuracy would decrease to 41.20%.[4] Nevertheless, the proposed method achieves an approximately 14% improvement. The superior performance indicates that combining GSVs and MLVFs can be used to obtain more discriminating power than that achieve when using conventional features.

5.6. MIREX Contest

To further demonstrate the feasibility of the proposed method, we participated in the MIREX genre classification contest. The competition is rigorous because it evaluates each submission based on threefold cross validation (with artist filtering) using a private dataset that contains 7000 music clips from 10 genres. Table V shows a comparison of the recognition rates of the winning MIREX submissions over the past seven years.[5] The results are directly comparable because the same dataset (which is not available to the public) has been used in evaluations since 2007. Our team has won the competition for three consecutive years since 2011.[6] Our submissions from 2011 to 2013 used the same early fusion approach, but different features were used, as shown in Table V. In particular, because we used more visual features at various levels, the performance improved.

6. CONCLUSION AND FUTURE WORK

This article proposes the MLVFs as a new feature set for music genre classification. The MLVFs are based on the time-frequency perspective, which includes song-level

[4]38107/(91253+1246)=41.20%.

[5]We have not listed the accuracy of Philippe Hamel [Pei and Hsu 2009] in 2011 because the author declared that the result was untrustworthy due to an unforeseen bug in his submission.

[6]Participation in the MIREX genre classification contest began in 2010. The 2010 submission achieved an accuracy level of 67.57%; however, the 2010 submission differed from the method used in this study, and therefore, is not introduced.

Table IV. Comparison between the Proposed Approach and Other Approaches

Method	Dataset	Accuracy
Panagakis et al. [2014]	GTZAN	89.40%
Proposed method[a]	GTZAN	88.60%
Seyerlehner et al. [2011]	GTZAN	87.03%
Yeh et al. [2013]	GTZAN	85.70%
Panagakis et al. [2010]	GTZAN	84.30%
Ren and Jang [2012]	GTZAN	81.70%
Bergstra et al. [2010]	GTZAN	81.00%
Tsunoo et al. [2011]	GTZAN	76.10%
Proposed method	Unique	77.66%
Seyerlehner et al. [2011]	Unique	75.86%
Panagakis et al. [2014]	Unique	75.05%
Proposed method	MASD	41.76%
Statistical spectrum descriptors [Lidy and Rauber 2005]	MASD	27.41% [Schindler et al. 2012]
MFCCs [Rabiner and Juang 1993]	MASD	24.13% [Schindler et al. 2012]
LPC [McKay 2010]	MASD	17.92% [Schindler et al. 2012]
Low-level spectral features [McKay 2010]	MASD	17.91% [Schindler et al. 2012]
Rhythm patterns [Lidy and Rauber 2005]	MASD	17.23% [Schindler et al. 2012]

[a] $\mathbf{f}_{GSV} + \mathbf{f}_{MLVF}$ with confidence-based late fusion.

Table V. Comparison between the Proposed Approach and Winning Submissions in the MIREX Genre Classification Contest (Mixed Popular Dataset)

Submission	Ranking (# of submissions)	Year	Accuracy
Our 2013 submission[b]	**1 (11)**	**2013**	**76.23%**
Our 2012 submission[c]	**1 (16)**	**2012**	**76.13%**
Our 2011 submission[d] **[Wu et al. 2011]**	**1 (15)**	**2011**	**75.57%**
Seyerlehner et al. [2010]	1 (24)	2010	73.64%
Cao and Li [2009]	1 (31)	2009	73.33%
MARSYAS [Tzanetakis 2007]	1 (13)	2008	66.41%
IMIRSEL M2K [Downie et al. 2005]	1 (7)	2007	68.29%

[a] $\mathbf{f}_{GSV} + \mathbf{f}_{MLVF}$ with confidence-based late fusion.
[b] $\mathbf{f}_{GSV} + \mathbf{f}_{MLVF}$ with early fusion.
[c] $\mathbf{f}_{GSV} + \mathbf{f}_{song} + \mathbf{f}_{beat}$ with early fusion.
[d] $\mathbf{f}_{GSV} + \mathbf{f}_{song}$ with early fusion.

and beat-level texture features, and beat-level heterogeneity features. The proposed confidence-based late fusion method successfully combines different types of feature.

The findings are summarized as follows:

(1) The experimental results show that the proposed MLVFs are more effective at describing spectrogram characteristics than song-level texture features, indicating the importance of using multiple temporal resolutions when design features.
(2) The experimental results indicate that the MLVFs are more critical than or equally critical to GSVs when the confidence-based late fusion is applied. This implies that time-frequency analysis may be more important than timbre analysis for music genre classification.
(3) Both confidence-based late fusion and early fusion approaches can effectively combine acoustic and visual features; however, the optimal fusion approach could be dataset (genre) dependent.

In addition to genre classification, MLVFs have been applied to music mood recognition and classical composer identification in MIREX contests. Future studies should apply MLVFs to other MIR tasks, including tag annotation and audio music similarity, to demonstrate the feasibility of MLVFs. We should also apply dimensionality reduction analysis to find the intrinsic structure embedded in MLVFs. Because the proposed confidence-based late fusion is a generic scheme for combining multiple decisions from SVM classifiers using different features, we should also explore the possibility of applying the proposed fusion method to other machine learning tasks. We will also develop a framework of probability-based late fusion based on the proposed confidence measures as a direction of our future work.

REFERENCES

Jeremy F. Alm and James S. Walker. 2002. Time-frequency analysis of musical instruments. *SIAM Review* 44, 3, 457–476.

James Bergstra, Michael I. Mandel, and Douglas Eck. 2010. Scalable genre and tag prediction with spectral covariance. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*. J. Stephen Downie and Remco C. Veltkamp (Eds.), International Society for Music Information Retrieval, 507–512. http://dblp.uni-trier.de/db/conf/ismir/ismir2010.html#BergstraME10.

Thierry Bertin-Mahieux, Daniel P. W. Ellis, Brian Whitman, and Paul Lamere. 2011. The million song dataset. In *Proceedings of the International Conference on Music Information Retrieval*. 591–596.

William M. Campbell, Douglas E. Sturim, and Douglas A. Reynolds. 2006. Support vector machines using GMM supervectors for speaker verification. *IEEE Sig. Process. Lett.* 13, 5, 308–311.

Chuan Cao and Ming Li. 2009. Thinkits submission for MIREX 2009 audio music classification and similarity tasks. http://www.music-ir.org/mirex/results/2009/abs/CL.pdf.

Chih-Chung Chang and Chih-Jen Lin. 2010. LIBSVM: A library for support vector machine. (2010). http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Zhi-Sheng Chen, Jyh-Shing Roger Jang, and Chin-Hui Lee. 2011. A kernel framework for content-based artist recommendation system in music. *IEEE Trans. Multimedi*. 13, 6, 1371–1380.

Y. M. G. Costa, L. S. Oliveira, A. L. Koerich, F. Gouyon, and J. G. Martins. 2012. Music genre classification using LBP textural features. *Sig. Process.* 92, 11, 2723–2737. DOI:http://dx.doi.org/10.1016/j.sigpro.2012.04.023

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* 7, 1–30. http://dl.acm.org/citation.cfm?id=1248547.1248548.

Hrishikesh Deshpande, Rohit Singh, and Unjung Nam. 2001. Classification of music signals in the visual domain. In *Proceedings of the COST-G6 Conference on Digital Audio Effects*. 1–4.

J. Stephen Downie, Andreas F. Ehmann, and Xiao Hu. 2005. Music-to-knowledge (M2K): A prototyping and evaluation environment for music digital library research. In *Proceedings of the 5th ACM/IEEE Joint Conference on Digital Libraries*. IEEE, 376–376.

Daniel P. W. Ellis. 2007. Beat tracking by dynamic programming. *J. New Music Res.* 36, 1, 51–60.

Daniel P. W. Ellis and Graham E. Poliner. 2007. Identifying cover songs' with chroma features and dynamic programming beat tracking. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. Vol. 4, IEEE, 1429–1432.

Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. 2011. A survey of audio-based music classification and annotation. *IEEE Trans. Multimed.* 13, 2, 303–319. DOI:http://dx.doi.org/10.1109/TMM.2010.2098858

Jean-Luc Gauvain and Chin-Hui Lee. 1994. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Trans. Speech Audio Process.* 2, 2, 291–298. DOI:http://dx.doi.org/10.1109/89.279278

Masataka Goto. 2003. SmartMusicKiosk: Music listening station with chorus-search function. In *Proceedings of the 16th ACM Conference on User Interface Software and Technology*. ACM, 31–40.

Peter Grosche, Joan Serrà, Meinard Müller, and Josep Ll. Arcos. 2012. Structure-based audio fingerprinting for music retrieval. In *Proceedings of the International Conference on Music Information Retrieval*. 55–60.

Dan-Ning Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai. 2002. Music type classification by spectral contrast feature. In *Proceedings of the IEEE International Conference on Multimedia and Expo.* Vol. 1, 113–116. DOI:http://dx.doi.org/10.1109/ICME.2002.1035731

Josef Kittler, Mohamad Hatef, Robert P. W. Duin, and Jiri Matas. 1998. On combining classifiers. *IEEE Trans. Patt. Anal. Mach. Intell.* 20, 3 (1998), 226–239.

Chang-Hsing Lee, Jau-Ling Shih, Kun-Ming Yu, and Hwai-San Lin. 2009. Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features. *IEEE Trans. Multimed.* 11, 4, 670–682. DOI:http://dx.doi.org/10.1109/TMM.2009.2017635

Thomas Lidy and Andreas Rauber. 2005. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In *Proceedings of the International Conference on Music Information Retrieval*. 34–41.

Cory McKay. 2010. Automatic music classification with jMIR. Ph.D. dissertation, McGill University, Canada.

Anders Meng, Peter Ahrendt, Jan Larsen, and Lars Kai Hansen. 2007. Temporal feature integration for music genre classification. *IEEE Trans. Audio, Speech, Lang. Process.* 15, 5 (July 2007), 1654–1664. DOI: http://dx.doi.org/10.1109/TASL.2007.899293

Anders Meng and John Shawe-Taylor. 2005. An investigation of feature models for music genre classification using the support vector classifier. In *Proceedings of the International Conference on Music Information Retrieval*. 604–609.

Meinard Muller, Daniel P. W. Ellis, Anssi Klapuri, and Gaël Richard. 2011. Signal processing for music analysis. *IEEE J. Select. Topics Sig. Process.* 5, 6, 1088–1110.

Timo Ojala, Matti Pietikainen, and Topi Maenpaa. 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Patt. Anal. Machine Intell.* 24, 7, 971–987.

François Pachet and Daniel Cazaly. 2000. A taxonomy of musical genres. In *Proceedings of the RIAO Conference*. 1238–1245.

Y. Panagakis, C. L. Kotropoulos, and G. R. Arce. 2014. Music genre classification via joint sparse low-rank representation of audio features. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* 22, 12, 1905–1917. DOI: http://dx.doi.org/10.1109/TASLP.2014.2355774

Yannis Panagakis, Constantine Kotropoulos, and Gonzalo R. Arce. 2010. Non-negative multilinear principal component analysis of auditory temporal modulations for music genre classification. *IEEE Trans. Audio, Speech, and Lang. Process.* 18, 3, 576–588. DOI: http://dx.doi.org/10.1109/TASL.2009.2036813

Jouni Paulus, Meinard Müller, and Anssi Klapuri. 2010. State of the art report: Audio-based music structure analysis. In *Proceedings of the International Conference on Music Information Retrieval*. 625–636.

Soo-Chang Pei and Nien-Teh Hsu. 2009. Instrumentation analysis and identification of polyphonic music using beat-synchronous feature integration and fuzzy clustering. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 169–172.

Lawrence Rabiner and Biing-Hwang Juang. 1993. *Fundamentals of Speech Recognition*. Vol. 14, Prentice Hall PTR.

Jia-Min Ren and J. R. Jang. 2012. Discovering time-constrained sequential patterns for music genre classification. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* 20, 4, 1134–1144. DOI: http://dx.doi.org/10.1109/TASL.2011.2172426

Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. 2000. Speaker verification using adapted gaussian mixture models. *Digital Signal Process.* 10, 13, 19–41. DOI:http://dx.doi.org/10.1006/dspr.1999.0361

Alexander Schindler, Rudolf Mayer, and Andreas Rauber. 2012. Facilitating comprehensive benchmarking experiments on the million song dataset. In *Proceedings of the International Conference on Music Information Retrieval*. 469–474.

Klaus Seyerlehner. 2010. Content-based music recommender systems: Beyond simple frame-level audio similarity. Ph.D. dissertation, Johannes Kepler University, Linz, Austria.

Klaus Seyerlehner, Markus Schedl, Peter Knees, and Reinhard Sonnleitner. 2011. Draft: A refined block-level feature set for classification, similarity and tag prediction. http://www.music-ir.org/mirex/abstracts/2011/SSKS1.pdf.

Klaus Seyerlehner, Markus Schedl, Tim Pohle, and Peter Knees. 2010. Using block-level features for genre classification, tag classification and music similarity estimation. http://www.music-ir.org/mirex/abstracts/2010/SSPK1.pdf.

E. Tsunoo, G. Tzanetakis, N. Ono, and S. Sagayama. 2011. Beyond timbral statistics: Improving music classification using percussive patterns and bass lines. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* 19, 4 (May 2011), 1003–1014. DOI: http://dx.doi.org/10.1109/TASL.2010.2073706

George Tzanetakis. 2007. MARSYAS submissions to MIREX 2007. http://www.music-ir.org/mirex/abstracts/2007/AI_CC_GC_MC_AS_tzanetakis.pdf.

George Tzanetakis and Perry Cook. 2002. Musical genre classification of audio signals. *IEEE Trans. Speech Audio Process.* 10, 5.

Ming-Ju Wu, Zhi-Sheng Chen, Jyh-Shing Jang, Jia-Min Ren, Yi-Hsung Li, and Chun-Hung Lu. 2011. Combining visual and acoustic features for music genre classification. In *Proceedings of the International Conference on Machine Learning and Applications (ICMLA)*. Vol. 2, IEEE, 124–129.

Ting-Fan Wu, Chih-Jen Lin, and Ruby C. Weng. 2004. Probability estimates for multi-class classification by pairwise coupling. *J. Mach. Learn. Res.* 5, 975–1005.

C.-C. M. Yeh, Li Su, and Yi-Hsuan Yang. 2013. Dual-layer bag-of-frames model for music genre classification. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 246–250. DOI: http://dx.doi.org/10.1109/ICASSP.2013.6637646

Received September 2014; revised January 2015 and April 2015; accepted April 2015