INSTRUMENTATION ANALYSIS AND IDENTIFICATION OF POLYPHONIC MUSIC USING BEAT-SYNCHRONOUS FEATURE INTEGRATION AND FUZZY CLUSTERING

Soo-Chang Pei and Nien-Teh Hsu

Graduate Institute of Communication Engineering, National Taiwan University, Taiwan, R.O.C.
Email: pei@cc.ee.ntu.edu.tw, r96942047@ntu.edu.tw

ABSTRACT

In this paper, a music instrumentation analysis and identification method is proposed. In contrast to existing systems, it tries to identify the whole instrument set in polyphonic music and also decide whether each instrument actually dominates at a particular moment or not, but without calculating the exact pitch, onset timing, or the volume of each note. Moreover, it does not require the music source separation in advance. We address this problem by incorporating the beat-synchronous scheme with fuzzy clustering to analyze the instrument components. Experiments show that the instrument identification process results in an 85.19% averaging recognition rate, which is comparable with other existing systems. In addition, it generates the extra time-varying instrumentation information. This information can be considered as a new mid-level feature in music information retrieval systems.

Index Terms— instrumentation analysis, fuzzy clustering, music information retrieval

1. INTRODUCTION

Automatic instrument identification of music signal is an interesting topic in signal processing and can have many potential applications. For example, the identification results can be used to determine the music genre, or be utilized as an aspect in music recommendation systems. Researchers have begun with the easiest task, which deals with the monophonic music in the last decade [1]. Several techniques have been applied to identify the isolated notes or phrases of the instrument, and the techniques had gradually come to maturity.

Identification the instrument set of polyphonic music is more complex and challenging. Eggink *et al.* first developed a system that can identify a predominant solo instrument in the presence of an accompanying keyboard or orchestra using harmonic features and GMM classifiers [2]. They make an assumption that the harmonic structure of the predominant melody should stick out than that of other instruments. Kitahara *et al.* tried to decompose the polyphonic problem into three sub-problems [3]. However, in their system the note data information should be already known. Essid *et al.* exploited a taxonomy of music ensembles by applying the hierarchical clustering technique [4].

The above works aim to identify the instruments appearing in an audio file. Nevertheless, it seems like they cannot reveal the instrumentation cue written by composers. In music, the term *instrumentation* refers to the particular combination of musical instruments employed in a composition. Some instruments tend to appear in only a few segments instead of the whole song. Instruments are also expected to change their roles of representing solo and accompaniment during the progression. Generally, the dominance of an instrument should be examined and analyzed by its melody line, volume, and even the note component. Here we only use the volume heard by the listeners to approximate the dominance. The aim of this work is to roughly manifest this time-varying instrumentation information.

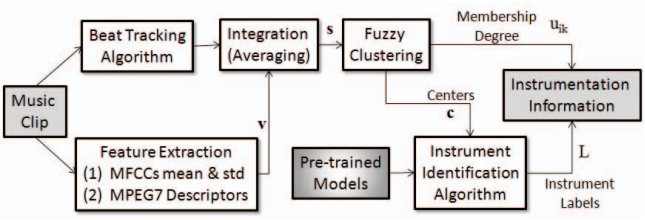


Fig. 1. Block diagram of the proposed instrumentation analysis system.

2. SYSTEM DESCRIPTION

2.1. Overview

The entire block diagram is shown in Fig. 1. To begin with, the feature vectors and beat data of the input polyphonic music clip are extracted. After that an integration process is formed by averaging frames inside the same beat intervals. A fuzzy clustering algorithm is then applied to the integrated feature. The number of clusters should equal to the number of instruments. Finally, the cluster center and a few corresponding integrated features are used in the instrument identification process to determine the final instrument set.

Table 1. Detail of the feature vector used in this system.

#	**MFCC Features**
01 - 13	Mean of the first 13 MFCCs
14 - 26	Standard deviation of the first 13 MFCCs
#	**MPEG-7 Timbre Descriptors**
27	Harmonic Centroid Descriptor
28	Harmonic Deviation Descriptor
29	Harmonic Spread Descriptor
30	Harmonic Variation Descriptor
31	Spectral Centroid Descriptor
32	Temporal Centroid Descriptor
33	Log-Attack-Time Descriptor

2.2. Feature Extraction

In order to simplify and unify the system, we select two low-level feature sets recommended in [5]. They are the MPEG-7 timbre descriptors and MFCC features (listed in Table 1). Musical instrument timbre descriptors in MPEG-7 standard aim at describing perceptual features of instrument sounds [6]. Collectively, they form a 33-dimensional feature vector of each frame. The input music clip is first converted to mono if needed, and then downsample to 16000 Hz to enhance the processing speed. After that, a hamming window with overlapped frame is applied to extract each feature vector.

2.3. Beat Tracking and Feature Integration

So far, the feature vectors obtained from frames can only reveal the spectral information in a considerably short period, which is believed to be stationary. The integration of these fragmentary vectors needs to be done according to the note data messages inside a music clip. However, since that calculating the exact onset time and duration of the note is still fairly challenging today, we adopt the beat-synchronous integration scheme instead [7]. BeatRoot is used to perform the beat-tracking algorithm to the input signal [8]. It estimates the beginning and the ending time of each beat and gives an acceptable result for further processing. Generally, a music clip tends to have only one or two notes appearing inside a beat time. Minor exceptions can be treated as outliers and be removed by a smoothing filter in the final stage.

Let $\mathbf{v}_{k,1}$, $\mathbf{v}_{k,N_k}$ denote the feature vectors calculated from the first and the last frame in the kth beat, respectively. Then the integration process can be done by averaging N_k vectors,

$$\mathbf{s}_k = \sum_{t=1}^{N_k} \frac{1}{N_k} \mathbf{v}_{k,t}, \tag{1}$$

where $\mathbf{s}_k$ denotes the beat-synchronous integrated vector in the kth beat.

2.4. Fuzzy Clustering

One of the major difficulty in instrument identification of polyphonic music is the timbre mismatch between the training and testing instruments. For instance, the violin used in the training process could not resemble the one in the testing music. Due to this reason, directly applying the beat-synchronous feature to supervised classifiers in a frame-by-frame manner would not perform well. On the other hand, we exploit the temporal continuity property in the instrumentation to solve this problem. That is, in most cases the instrument tends to be arranged consistently and the timbre of each specific instrument should not have a large change. We thus apply the fuzzy clustering technique to the integrated vectors of the entire music clip.

The fuzzy c-means clustering (FCM) algorithm attempts to partition a finite collection of elements $\mathbf{s}$ into a collection of c fuzzy clusters with respect to minimizing the following objective function Q:

$$Q = \sum_{i=1}^{c} \sum_{k} u_{ik}^{m} ||\mathbf{s}_k - \mathbf{c}_i||^2, \tag{2}$$

where m is a fuzzification coefficient ($m = 2$ in this paper). We use $\mathbf{c}$ and u_{ik} to denote the resulting cluster centers and the membership function. They can be obtained by iteratively repeating the following equations:

$$\mathbf{c}_i(t) = \frac{\sum_k u_{ik}^m(t)\mathbf{s}_k}{\sum_k u_{ik}^m(t)}, \tag{3}$$

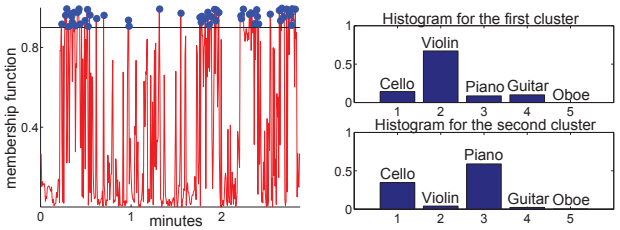
and

$$u_{ik}(t+1) = \frac{1}{\sum_{j=1}^{c} (\frac{||\mathbf{s}_k - \mathbf{c}_i(t)||}{||\mathbf{s}_k - \mathbf{c}_j(t)||})^{2/(m-1)}} \tag{4}$$

Details of the algorithm can be found in [9]. Unlike hard k-means clustering, FCM gives the membership function as soft labeling, which can be regarded as the degree of dominance for a particular instrument. Experimental result shows that the volume of the instrument would directly make influence on the membership function output. Since estimating the number of instruments is beyond the scope of our work, we manually fed the correct number c into the system.

2.5. Instrument Identification

The remaining work is to identify the correct instrument represented by each cluster. Before the identification process, we use the support vector machine (SVM) to build pre-trained models [10]. The training data is collected from different solo recordings, in order to consider the timbre variation between different music clips. Since that directly classifying the cluster centers using SVMs sometimes gives unfavorable results,



(a) (b)

Fig. 2. Instrument identification illustration. (a) Selection of integrated vectors with their membership function exceeding the threshold. (b) Calculating the instrument labeling histogram of selected integrated vectors using pre-trained SVM models for each cluster.

we use an alternative method. A membership degree threshold T (0.9 in this paper) is set. For every cluster, integrated vectors with their membership function values higher than the threshold are grouped together and individually applied to SVMs to obtain the classification labeling result l_i of those integrated vectors.

$$l_i = \text{SVM}\{\mathbf{x}_k | u_{ik} > T\} \tag{5}$$

The result is then used to calculate the instrument labeling histogram. An example is shown in Fig. 2.

Let $H_{i,j}$ denote the jth instrument labeling count in the ith cluster derived from l_i, the labeling probability $P_{i,j}$ is generated by averaging the count $H_{i,j}$ within the same cluster as follows,

$$P_{i,j} = \frac{H_{i,j}}{\sum_{j=1}^{N_j} H_{i,j}} \tag{6}$$

The identification process is done by selecting the largest probability in $P_{i,j}$,

$$L_i = \arg_j \max_{i,j} P_{i,j}, \tag{7}$$

where L_i represents the identification result of the ith cluster. After the first step, the jth instrument is marked as already used in the histogram table. The system will continue to find the largest probability that exists in unused instruments and so on, until all clusters are labeled. For example, in Fig. 2(b) cluster 1 will first be labeled as violin, and then cluster 2 will be labeled as piano. Finally, the labeled instrument set L_i and the membership function u_{ik} are treated as the instrumentation information output. This method is designed to solve the timbre mismatch problem. Since that the fuzzy clustering step is unsupervised, integrated vectors will automatically be clustered together with respect to different instruments, due to its temporal continuity property. Classifiers are applied to the clustering result in the last stage.

Table 2. Recognition rates of different instrument combinations in Western classical music. Note that string is regarded as a combination of violin and cello here.

	Violin Sonata		Cello Sonata
Violin	82.93%	Cello	85.71%
Piano	85.37%	Piano	90.48%
	Piano Trio		Oboe Concerto
Piano	96.67%	Oboe	83.33%
Violin	81.11%	String	66.67%
Cello	94.44%	**Average**	**85.19%**

3. EXPERIMENTS AND EVALUATION

3.1. Experiment Setup

The evaluation of this work can be divided into two parts. First, the instrument identification process gives an estimation of the instrument set. We can calculate the averaging recognition rate by testing a set of duo and trio songs. For another, the instrumentation analysis results output a time-varying distribution related to each instrument. In this part we select two famous classical music clips to demonstrate our simulation results.

3.2. Instrument Identification Result

In this experiment, five common instrument models are trained by the SVM using clean solo recordings beforehand. They are cello, violin, piano, guitar and oboe. The average length of training data for each model is about 50 minutes. To evaluate the identification performance, instead of using the MIDI-based synthesized files, we select four regular musical forms in real-world Western classical music as the testing data. Each of them is composed of different instrument sets. The database consists of 200 music clips, and the overall duration of the music clips is about 10 hours. The recognition accuracy of instrument i is defined by

$$\text{Accuracy}_i = \frac{\#\text{ of clips correctly identified as } i}{\#\text{ of testing music clips}} \tag{8}$$

The result is listed in Table 2. It gives an 85.19% recognition rate in average, which is essentially comparable to other relative works in terms of the training model size [4], [11].

3.3. Instrumentation Analysis Result

Membership function output from the fuzzy clustering algorithm combined with the instrumental labels are considered as the dominance of the instrument played in a music clip. We select two well-known Western classical music pieces to demonstrate the result. They are a violin sonata composed by Beethoven, and a piano trio composed by Brahms. Fig. 3 and

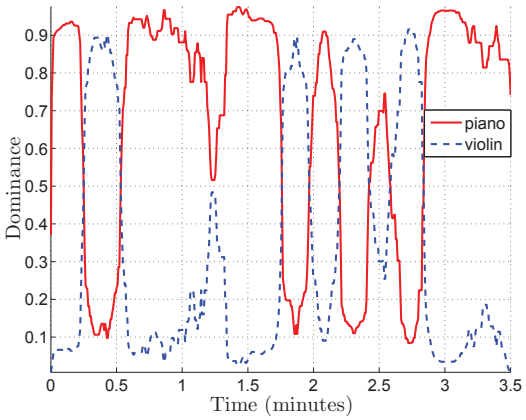


Fig. 3. Simulation results of *Violin Sonata "Spring" mov.4 by Beethoven*. Only the 3.5 minutes in beginning is selected.

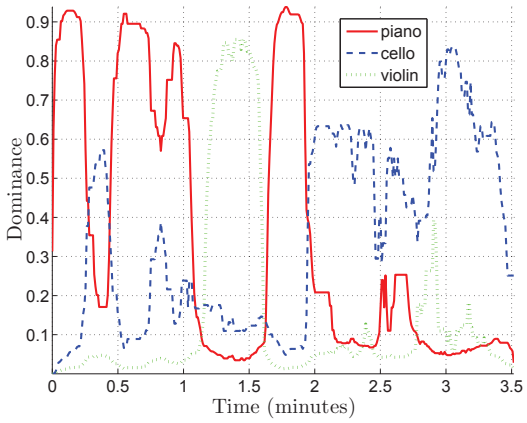


Fig. 4. Simulation results of *Piano Trio mov.3 by Brahms*. Only the 3.5 minutes in beginning is selected.

Fig. 4 show the results of these duo and trio pieces, respectively. By referring to the music scores and recordings, it can be shown that the system output a reasonable estimation.

4. CONCLUSIONS AND FUTURE WORK

In this paper, an instrumentation analysis algorithm for polyphonic music is proposed. The system can automatically identify the instrument set without knowing note data information such like the pitch and onset timing. As mentioned earlier, the system takes considerably less computation time. It only requires a beat-tracking algorithm with fuzzy clustering technique. Moreover, it can roughly sketch the dominance of each instrument during the music progression. We believe this time-varying result can be used as a mid-level feature to improve the performance of music recommendation systems. For instance, the systems may recommend a list of songs which is similar in their instrumentation information as a new option, instead of using the genre or artist metadata.

For future work, we would like to increase the number of training instruments. The system will be modified to further accommodate to drum and human voice, which are very common in recent popular music. The inharmonic nature of these signals needs to be handled by extra algorithms.

5. REFERENCES

[1] J. Marques and P. J. Moreno, "A study of musical instrument classification using gaussian mixture models and support vector machines," *Tech. Rep., Cambridge Research Laboratory*, vol. 4, 1999.

[2] J. Eggink and G. J. Brown, "Instrument recognition in accompanied sonatas and concertos," in *Proc. ICASSP*, 2004, vol. 4, pp. 217–220.

[3] T. Kitahara, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Instrument identification in polyphonic music: Feature weighting with mixed sounds, pitch-dependent timbre modeling, and use of musical context," in *Proc. ISMIR*, 2005, pp. 558–563.

[4] S. Essid, G. Richard, and B. David, "Instrument recognition in polyphonic music based on automatic taxonomies," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 1, pp. 68–80, 2006.

[5] J. D. Deng, C. Simmermacher, and S. Cranefield, "A study on feature analysis for musical instrument classification," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 38, no. 2, pp. 429–438, 2008.

[6] S. F. Chang, T. Sikora, and A. Purl, "Overview of the MPEG-7 standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 688–695, 2001.

[7] D. P. W. Ellis, C. V. Cotton, and M. I. Mandel, "Cross-correlation of beat-synchronous representations for music similarity," in *Proc. ICASSP*, 2008, pp. 57–60.

[8] S. Dixon, "Automatic extraction of tempo and beat from expressive performances," *Journal of New Music Research*, vol. 30, no. 1, pp. 39–58, 2001.

[9] W. Pedrycz and F. Gomide, *Fuzzy Systems Engineering: Toward Human-Centric Computing*, Wiley-IEEE Press, 2007.

[10] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," 2001, Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[11] T. Kitahara, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Instrogram: A new musical instrument recognition technique without using onset detection nor f0 estimation," in *Proc. ICASSP*, 2006, pp. 14–19.