MIRAI: Multi-hierarchical, FS-tree based Music Information Retrieval System

Zbigniew W. Raś[1,2], Xin Zhang[1], and Rory Lewis[1]

[1] University of North Carolina, Dept. of Comp. Science, Charlotte, N.C. 28223, USA
[2] Polish-Japanese Institute of Information Technology, 02-008 Warsaw, Poland

Abstract. With the fast booming of online music repositories, there is a need for content-based automatic indexing which will help users to find their favorite music objects in real time. Recently, numerous successful approaches on musical data feature extraction and selection have been proposed for instrument recognition in monophonic sounds. Unfortunately, none of these methods can be successfully applied to polyphonic sounds. Identification of music instruments in polyphonic sounds is still difficult and challenging, especially when harmonic partials are overlapping with each other. This has stimulated the research on music sound separation and new features development for content-based automatic music information retrieval. Our goal is to build a cooperative query answering system (QAS), for a musical database, retrieving from it all objects satisfying queries like "find all musical pieces in pentatonic scale with a viola and piano where viola is playing for minimum 20 seconds and piano for minimum 10 seconds". We use the database of musical sounds, containing almost 4000 sounds taken from the MUMs (McGill University Master Samples), as a vehicle to construct several classifiers for automatic instrument recognition. Classifiers showing the best performance are adopted for automatic indexing of musical pieces by instruments. Our musical database has an FS-tree (Frame Segment Tree) structure representation. The cooperativeness of QAS is driven by several hierarchical structures used for classifying musical instruments.

1 Introduction

Broader research on automatic musical instrument sound classification goes back to last few years. So far, there is no standard parameterization used as a classification basis. The sound descriptors used are based on various methods of analysis of time and spectrum domain, with Fourier Transform amplitude spectra being most common. Also, wavelet analysis gains increasing interest for sound and especially for musical sound analysis and representation [21], [9]. Diversity of sound timbres is also used to facilitate data visualization via sonification, in order to make complex data easier to perceive [1].

Many parameterization and recognition methods, including pitch extraction techniques, applied in musical research come from speech and speaker recognition domain [5], [22]. Sound parameters applied in research performed in musical instrument classification include cepstral coefficients, constant-Q coefficients,

spectral centroid, autocorrelation coefficients, and moments of the time wave [3], wavelet analysis [23], [13], root mean square (RMS), amplitude envelope and multidimensional scaling analysis trajectories [12], and various spectral and temporal features [14], [17], [23]. The sound sets used differ from experiment to experiment, with McGill University Master Samples (MUMS) CDs being most common [19], yet not always used [3], making comparison of results more difficult. Some experiments operate on a very limited set of data, like 4 instruments, or singular samples for each instrument. Even if the investigations are performed on MUMS data, every researcher selects different group of instruments, number of classes, and testing method is also different. Therefore, data sets used in experiments and the obtained results are not comparable. Additionally, each researcher follows different parameterization technique(s), which makes comparison yet more difficult. Audio features in our system [26], [15] are first categorized as MPEG7 descriptors and other/non-MPEG7 descriptors in the acoustical perspective of view, where both spectrum features and temporal features are included. We have built a derivative database of those features with single valued data for KD-based classification. The spectrum features have two different frequency domains: Hz frequency and Mel frequency. Frame size was carefully designed to be 120ms, so that the 0th octave G (the lowest pitch in our audio database) can be detected. The hop size is 40ms with a overlapping of 80ms. A hamming window was applied to all STFTs (Short Time Fourier Transforms) to avoid jittering in the spectrum. By the results from the experiments, it was shown that the non-MPEG features significantly improve the performance of the classifiers [28].

The classifiers, applied in research on musical instrument sound classification, represent practically all known methods. The most popular classifier is k-Nearest Neighbor (k-NN), see for example [12]. This classifier is relatively easy to implement and quite successful. Other reported results include Bayes decision rules, Gaussian mixture model [3], artificial neural networks [13], decision trees and rough set based algorithms [24], discriminant analysis [17] hidden Markov Models (HMM), support vector machines (SVM) and other. The obtained results vary depending on the size of the data set, with accuracy reaching even 100% for 4 classes. However, the results for more than 10 instruments, explored in full musical scale range, generally are below 80%. Extensive review of parameterization and classification methods applied in research on this topic, with obtained results, is given in [10]. The classifiers investigated in our project include k-NN, Bayesian Networks, and Decision Tree J-48. We also consider use of neural networks, especially time-delayed neural networks (TDNN), since they perform well in speech recognition applications [18].

Musical instrument sounds can be classified in various ways, depending on the instrument or articulation classification. In [25], we review a number of possible generalizations of musical instruments sounds classification which can be used to construct different hierarchical decision attributes. Each decision attribute leads to a new classifier and the same to a different system for automatic indexing of music by instrument sounds and their generalizations. Values of any decision attribute and their generalizations can be seen as atomic queries of a query

language built for retrieving musical objects from musical database. When query fails, the cooperative strategy tries to find its lowest generalization which does not fail, taking into consideration all available hierarchical attributes. Paper [25] evaluates two hierarchical attributes (Hornbostel-Sachs classification and classification by articulation) upon the same dataset which contains 2628 distinct musical samples of 102 instruments. By cross checking the resulting schemes for both attributes, it was observed that the timbre estimation of instruments had higher accuracy than that of instruments from other families by the classification by articulation. Also, among the musical objects played by different articulations, the sounds played by lip-vibration tended to be less correctly recognized by Hornbostel-Sachs classification. This justifies the construction of atomic queries from values of more than one decision attribute.

2 Sound Data

This paper deals with recordings where for each channel there is only access to one-dimensional data, i.e. to single sample representing amplitude of the sound. Any basic information like pitch (or pitches, if there are more sounds), timbre, beginning and end of the sound must be extracted via digital signal processing. The audio database consists of stereo musical pieces from the MUMS samples. These audio data files are treated as mono-channel, where only left channel was taken into consideration, since successful methods for the left channel will also be successfully applied to the right channel. In the view of classification, these audio data can be categorized into two different types: one is monophonic sound note to generate training feature set; the other is polyphonic sound sequence for testing.

Our research is driven by the desire to identify the individual instrument types or instrument family categories of the predominant instruments in a music object. Timbre is a quality of sound that distinguishes one music instrument from another, while there are a wide variety of instrument families and individual categories. It is rather subjective quality, defined by ANSI as the attribute of auditory sensation, in terms of which a listener can judge that two sounds, similarly presented and having the same loudness and pitch, are different. Such definition is subjective and not of much use for automatic sound timbre classification. Therefore, musical sounds must be very carefully parameterized to allow automatic timbre recognition. The real use of timbre-based grouping of music is discussed in [2]. Evolution of sound features in time is essential for humans, therefore it should be reflected in sound description as well. The discovered temporal patterns may better express sound features than static features, especially that classic features can be very similar for sounds representing the same family or pitch, whereas changeability of features with pitch for the same instrument makes sounds of one instrument dissimilar.

Based on recent research performed in MIR area, MPEG proposed an MPEG-7 standard, in which it described a set of low-level sound temporal and spectral

features. The low-level descriptors in MPEG-7 are intended to describe the time-variant information within an entire audio segment, where most of them are, like other STFT related acoustic features, in a form of either vector or matrix of large size, where an audio segment was divided into a set of frames and each row represents a power spectrum in the frequency domain within each analysis window. Therefore, these features are not suitable for traditional classifiers, which require single-value cell of input datasets. Researchers have been explored different statistical summations in a form of single value to describe signatures of music instruments within vectors or matrices in those features, such as Tristimulus parameters [20] or Brightness [6]. However, current features fail to sufficiently describe the audio signatures which vary in time within a whole sound segment, esp. where multiple audio signatures are overlapping with each other. It was widely observed that a sound segment of a note, which is played by a music instrument, has at least three states: onset (transient), quasi-steady state and offset (transient). Vibration pattern in a transient state is known to significantly differ from the one in a quasi-steady state. Consequently, the harmonic features in the transient states behavior significantly different from those in the quasi-steady state. Also, it has been observed that a human needs to know the beginning of the music sound in order to discern the type of an instrument. Identifying the boundary of the transient state enables accurate timbre recognition.

3 Feature Database Construction

Our research involves the construction of two main databases, one is a monophonic sound feature database, which is used for classifiers construction; the other is a polyphonic audio database, which is used for testing. The latter will have FS-tree structure driven by automatic indexing of audio files by music instruments and their classes. The monophonic sound feature database contains over 1022 attributes, where 1018 of them were computed from the digital monophonic sound files and four decision hierarchical attributes were manually labelled. There are many ways to categorize the audio features. In our research, computational audio features are first categorized as MPEG7 based descriptors and other/non-MPEG7 descriptors in the acoustical perspective of view, where both spectrum features and temporal features are included. Then, a derivative database of those features with single valued data features, for the purpose of learning classifiers, is constructed. The manually labelled decision attributes will be discussed in latter section. Spectrum features have different frequency domains: Hz frequency and Mel frequency. Frame size is chosen as 0.12 second, so that the 0th octave G (the lowest pitch in our audio database) can be detected, which is also within the range of estimates for temporal acuity of human ear. The hop size is 0.04 second with a overlapping of 0.08 second. Since the sampling frequency of all the music objects is 44,100Hz, there are 5292 sample data per frame in the waveform.

The list of MPEG7 features includes: Harmonic Upper Limit, Harmonic Ratio, Basis Functions, Log Attack Time, Temporal Centroid, Spectral Centroid,

Spectrum Centroid/Spread I, Harmonic Parameters, Flatness. The list of extended MPEG7 features and other features includes: Tristimulus Parameters, Spectrum Centriod/Spread II, Flux, Roll Off, Zero Crossing, MFCC, Spectrum Centroid/Spread I, Harmonic Parameters, Flatness, Durations. Intermediate features include Harmonic Upper Limit and Projection.

4 Sound Separation

Our system consists of five modules: a quasi-steady state detector, a $STFT$ converter with hamming window, a pre-dominant fundamental frequency estimator, a sequential pattern matching engine (it will be replaced by a classifier) with connection to a feature database, a FFT subtraction device [27].

The quasi-steady state detector computes overall fundamental frequency in each frame by a cross-correlation function, and outputs the beginning and end positions of the quasi-steady state of the input sound.

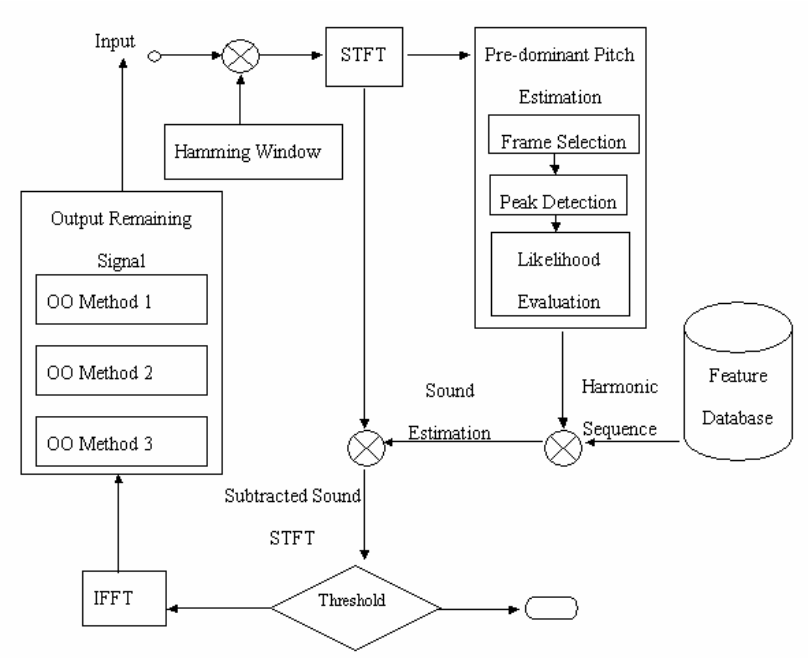

Fig. 1. Sound Separation System

The $STFT$ converter divides a digital audio object into a sequence of frames, applies $STFT$ transform to the mixed sample data of integers from time domain

to frequency domain with a hamming window, and outputs $NFFT$ discrete points.

The pre-dominant fundamental frequency estimator identifies all the possible harmonic peaks, computes the likelihood value for each candidate peak, elects the frequency with the maximum likelihood value as the fundamental frequency, and stores its normalized correspondence harmonic sequence.

The sequential-pattern matching engine computes the distance of each pair wise sequence of first N harmonic peaks, where N is set empirically, then outputs the sound with the minimum distance value for each frame, and finally estimates the sound object by the most frequent sound object among all the frames.

The FFT subtraction device subtracts the detected sound source from the spectrum, computes the imaginary and real part of the FFT point by the power and phase information, performs $IFFT$ for each frame, and outputs resultant remaining signals into a new audio data file.

5 Multi-way Hierarchic Classification

Classification of musical instrument sounds can be performed in various ways [11]. Paper [25] reviews several hierarchical classifications of musical instrument sounds but concentrates only on two of them: Hornbostel-Sachs classification of musical instruments and classification of musical instruments by articulation with 15 different articulation methods (seen as attribute values): blown, bowed, bowed vibrato, concussive, hammered, lip-vibrated, martele, muted, muted vibrato, percussive, picked, pizzicato, rubbed, scraped and shaken. Each hierarchical classification represents a unique decision attribute which leads us to a discovery of a new classifier and the same to a different system for automatic indexing of music by instruments and their certain generalizations.

The goal of each classification is to find descriptions of musical instruments or their classes (values of attribute d) in terms of values of attributes from A. Each classification results in a classifier which can be evaluated using standard methods like bootstrap or cross-validation.

In [25] authors concentrate on classifiers built by rule-based methods (for instance: *LERS, RSES, PNC2*) and next on classifiers built by tree-based methods (for instance: *See5, J48 Tree, Assistant, CART, Orange*).

Let us assume that $S = (X, A \cup \{d\}, V)$ is a decision system, where d is a hierarchical attribute. We also assume that $d_{[i_1,...,i_k]}$ (where $1 \leq i_j \leq m_j$, $j = 1, 2..., k$) is a child of $d_{[i_1,...,i_{k-1}]}$ for any $1 \leq i_k \leq m_k$. Clearly, attribute d has $\Sigma\{m_1 \cdot m_2 \cdot ... \cdot m_j : 1 \leq j \leq k\}$ values, where $m_1 \cdot m_2 \cdot ... \cdot m_j$ shows the upper bound for the number of values at the level j of d. By $p([i_1, ..., i_k])$ we denote a path (d, $d_{[i_1]}$, $d_{[i_1,i_2]}$, $d_{[i_1,i_2,i_3]}$,..., $d_{[i_1,...,i_{k-1}]}$, $d_{[i_1,...,i_k]}$) leading from the root of the hierarchical attribute d to its descendant $d_{[i_1,...,i_k]}$.

Let us assume that R_j is a set of classification rules extracted from S, representing a part of a rule-based classifier $R = \bigcup\{R_j : 1 \leq j \leq k\}$, and describing

all values of d at level j. The quality of a classifier at level j of attribute d can be checked by calculating $Q(R_j) = \frac{\sum\{sup(r)\cdot conf(r): r \in R_j\}}{\sum\{sup(r: r \in R_j\}}$, where $sup(r)$ is the support of the rule r in S and $conf(r)$ is its confidence. Then, the quality of the rule-based classifier R can be checked by calculating $Q(\bigcup\{R_j : 1 \leq j \leq k\}) = \frac{\sum\{Q(R_j): 1 \leq j \leq k\}}{k}$.

The quality of a tree-based classifier can be given by calculating its quality for every node of a hierarchical decision attribute d. Let us take a node $d_{[i_1,...,i_k]}$ and the path $p([i_1, ..., i_k])$ leading to that node from the root of d. There is a set of classification rules $R_{[i_1,...,i_m]}$, uniquely defined by the tree-based classifier, assigned to a node $d_{[i_1,...,i_m]}$ of a path $p([i_1, ..., i_k])$, for every $1 \leq m \leq k$. Now, we define $Q(R_{[i_1,...,i_m]})$ as $\frac{\sum\{sup(r)\cdot conf(r): r \in R_{[i_1,...,i_m]}\}}{\sum\{sup(r): r \in R_{[i_1,...,i_m]}\}}$. Then, the quality of a tree-based classifier for a node $d_{[i_1,...,i_m]}$ of the decision attribute d can be checked by calculating $Q(d_{[i_1,...,i_m]}) = \prod\{Q(R_{[i_1,...,i_j]}) : 1 \leq j \leq m\}$. In our experiments, presented in Section 4 of this paper, we use *J48 Tree* as the tool to build tree-based classifiers. Also, their performance on level m of the attribute d is checked by calculating $Q(d_{[i_1,...,i_m]})$ for every node $d_{[i_1,...,i_m]}$ at the level m. Finally, the performance of both classifiers is checked by calculating $Q(\bigcup\{R_j : 1 \leq j \leq k\})$ (the first method we proposed).

Learning values of a decision attribute at different generalization levels is extremely important not only for designing and developing an automatic indexing system of possibly highest confidence but also for handling failing queries. Values of a decision attribute and their generalizations are used to construct atomic queries of a query language built for retrieving musical objects from MIR Database (see http://www.mir.uncc.edu). When query fails, the cooperative strategy [7], [8] may try to find its lowest generalization which does not fail. Clearly, by having a variety of different hierarchical structures available for d we have better chance not only to succeed but succeed with a possibly smallest generalization of an instrument class.

6 Flexible Query Answering System

Now, we discuss how a Flexible Query Answering System (see Figure 1) associated with a database D of music files works for a sample query which consists of two parts: a digital musical file F and an instrument T. The query should be read as: *Find all musical pieces, in the database D, which are played by the same instruments as the instruments used in F.* Also the duration time of all these instruments has to be the same (threshold value can be provided).

The digital musical file is divided into segments of equal length. Automatic indexing system operates on each segment piece and outputs a vector of features describing its content. Then a classifier estimates what instruments are present in each segment and what is their time duration and then searches the FS-tree to identify the musical pieces in database D satisfying the query. If query

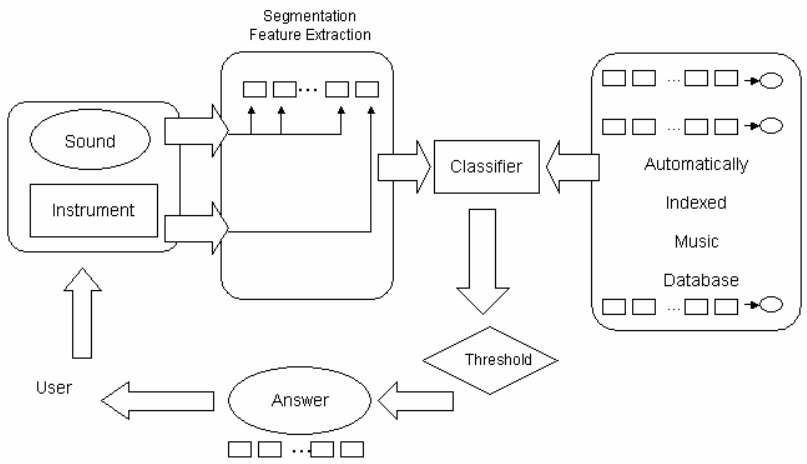

Fig. 2. Flexible Query Answering System based on MIR

fails, then an instrument used in F which has the most similar timbre to the instrument T is identified and it is replaced by T assuming that its time duration is the same as the time duration of the replaced instrument. Finally, the closest musical file to the file requested by user is returned as the result of the query. Alternatively, the classifier of a higher level in the instrument family tree is assigned for timbre classification on its own level, and repeats the steps until a desire result is achieved or the root of the instrument family tree is reached. This approach especially benefits non-musician users who have limited information on music instrument classification schema.

7 Conclusion and Acknowledgement

The ultimate goal of this research is to build a cooperative system for automatic indexing of music by instruments or classes of instruments, use this system to build FS-tree type music database for storing automatically indexed musical files, and finally design and implement a Cooperative Query Answering System to handle user requests submitted to music database.

This research was supported by the National Science Foundation under grant IIS-0414815.

References

1. Ben-Tal, O., Berger, J., Cook, B., Daniels, M., Scavone, G., Cook, P., "SONART: The Sonification Application Research Toolbox", Proceedings of the 2002 International Conference on Auditory Display, Kyoto, Japan, July 2002

2. Bregman, A.S., "Auditory scene analysis, the perceptual organization of sound", MIT Press, 1990
3. Brown, J. C., Houix, O., McAdams, S., "Feature dependence in the automatic identification of musical woodwind instruments", in *J. Acoust. Soc. of America*, 109, 2001, 1064-1072
4. Cardoso, J. F., Comon, P., "Independent Component Analysis, a Survey of Some Algebraic methods", In Proc. ISCAS Conference, vol. 2, 93-96, Atlanta, May 1996
5. Flanagan, J. L., "Speech Analysis, Synthesis and Perception", Springer-Verlag, New York, 1972
6. Fujinaga, I., McMillan, K., "Real time Recognition of Orchestral Instruments", in *International Computer Music Conference*, 2000, 141-143
7. Gaasterland, T., "Cooperative answering through controlled query relaxation", in *IEEE Expert*, Vol. 12, No. 5, 1997, 48-59
8. Godfrey, P., "Minimization in cooperative response to failing database queries", in *International Journal of Cooperative Information Systems*, Vol. 6, No. 2, 1993, 95-149
9. Goodwin, M. M., "Adaptive Signal Models: Theory, Algorithms, and Audio Applications", Ph.D. dissertation, University of California, Berkeley, 1997
10. Herrera, P., Amatriain, X., Batlle, E., Serra X. "Towards instrument segmentation for music content description: a critical review of instrument classification techniques". In Proc. of International Symposium on Music Information Retrieval (ISMIR 2000), Plymouth, MA, 2000.
11. Hornbostel, E. M. V., Sachs, C., "Systematik der Musikinstrumente. Ein Versuch", in *Zeitschrift fur Ethnologie*, Vol. 46, No. 4-5, 1914, 553-90, available at http://www.uni-bamberg.de/ppp/ethnomusikologie/HS-Systematik/HS-Systematik
12. Kaminskyj, I, "Multi-feature Musical Instrument Classifier", MikroPolyphonie 6, 2000 (online journal at http://farben.latrobe.edu.au/)
13. Kostek, B., Czyzewski, A., "Representing Musical Instrument Sounds for Their Automatic Classification", in *J. Audio Eng. Soc.*, Vol. 49, No. 9, 2001, 768-785
14. Kostek, B. Wieczorkowska, A., "Parametric Representation of Musical Sounds", in *Archive of Acoustics*, Vol. 22, No. 1, 1997, 3-26
15. Lewis, R., Zhang, X., Ras, Z.W., "Blind Signal Separation of Similar Pitches and Instruments in a Noisy Polyphonic Domain", in *Foundations of Intelligent Systems*, Proceedings of ISMIS 2006, F. Esposito et al. (Eds.), Bari, Italy, LNAI, No. 4203, Springer, 2006, 228-237
16. Manjunath, B. S., Salembier, P., Sikora, T. (Eds.), "Introduction to MPEG-7. Multimedia Content Description Interface", J. Wiley and Sons, 2002
17. Martin, K. D. and Kim, Y. E., "Musical instrument identification: a pattern-recognition approach", in *Proceedings of 136th Meeting of the Acoustical Society of America*, Norfolk, VA, October, 1998
18. Meier, U., Stiefelhagen, R., Yang, J., Waibel, A., "Towards Unrestricted Lip Reading", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 14, No. 5, 2000, 571-586
19. Opolko, F. and Wapnick, J., "MUMS - McGill University Master Samples", CD's, 1987
20. Pollard, H.F. and Jansson, E.V., "A Tristimulus Method for the spectificaiton of Musical Timbre", in *Acustica*, No. 51, 1982, 162-171
21. Popovic, I., Coifman, R., Berger, J., "Aspects of Pitch-Tracking and Timbre Separation: Feature Detection in Digital Audio Using Adapted Local Trigonometric

Bases and Wavelet Packets" Center for Studies in Music Technology, Yale University, Research Abstract, June 1995
22. Rabiner, L., Schafer, R., "Digital Processing of Speech Signals, Prentice-Hall, Englewood Cliffs, New Jersey, 1978
23. Wieczorkowska, A, "Musical Sound Classification based on Wavelet Analysis", in *Fundamenta Informaticae Journal*, Vol. 47, No. 1/2, 2001, 175-188
24. Wieczorkowska, A, "The recognition efficiency of musical instrument sounds depending on parameterization and type of a classifier", PhD. thesis (in Polish), Technical University of Gdansk, Poland, 1999
25. Wieczorkowska, A., Raś, Z.W., Zhang, X., Lewis, R., "Multi-way Hierarchic Classification of Musical Instrument Sounds", in *Proceedings of the IEEE CS International Conference on Multimedia and Ubiquitous Engineering (MUE 2007)*, April 26-28, 2007, in Seoul, Korea, will appear
26. Zhang, X., Raś, Z.W., "Differentiated Harmonic Feature Analysis on Music Information Retrieval For Instrument Recognition", in *Proceedings of IEEE International Conference on Granular Computing (IEEE GrC 2006)*, May 10-12, 2006, Atlanta, Georgia, 578-581
27. Zhang, X., Marasek, K., Raś, Z.W., "Maximum Likelihood Study for Sound Pattern Separation and Recognition", in *Proceedings of the IEEE CS International Conference on Multimedia and Ubiquitous Engineering (MUE 2007)*, April 26-28, 2007, in Seoul, Korea, will appear
28. Zhang, X., Raś, Z.W., "Analysis of Sound Features for Music Timbre Recognition", in *Proceedings of the IEEE CS International Conference on Multimedia and Ubiquitous Engineering (MUE 2007)*, April 26-28, 2007, in Seoul, Korea, will appear