International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:7, No:8, 2013

Orchestra/Percussion Classification Algorithm for Unified Speech Audio Coding System

Yueming Wang, Rendong Ying, Sumxin Jiang, and Peilin Liu

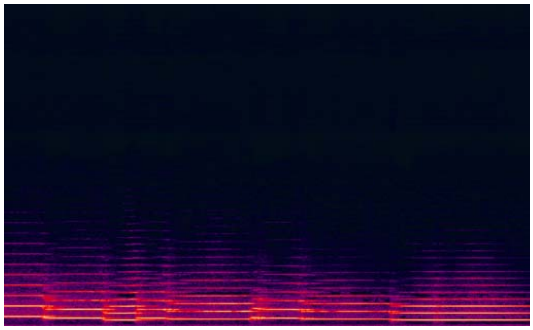
***Abstract*—**Unified Speech Audio Coding (USAC), the latest MPEG standardization for unified speech and audio coding, usesa speech/audio classificationalgorithmto distinguish speech and audio segments of the input signal. The quality of the recovered audio can be increased by well-designed orchestra/percussion classification and subsequent processing.However, owing to the shortcoming of the system, introducing an orchestra/percussion classification and modifying subsequent processing can enormously increase the quality of the recovered audio. This paper proposes an orchestra/percussion classification algorithm for the USAC system whichonly extracts 3 scales of Mel-Frequency Cepstral Coefficients (MFCCs) rather than traditional 13scales of MFCCs and use Iterative Dichotomiser 3 (ID3) Decision Treerather than other complex learning method, thusthe proposed algorithm has lower computing complexity than most existing algorithms. Considering that frequent changing of attributes may lead to quality loss of the recovered audio signal, this paper also design a modified subsequent process to help the whole classification system reach an accurate rate as high as 97% which is comparable to classical 99%.

***Keywords*—**ID3 Decision Tree, MFCC, Orchestra/Percussion Classification, USAC.

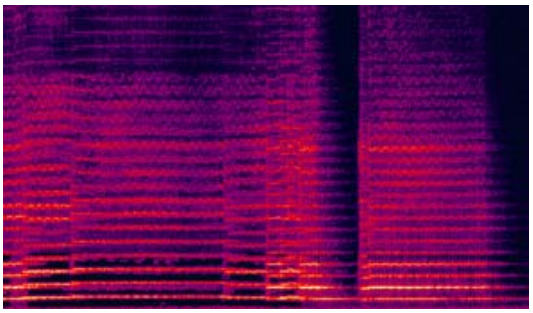
I. Introduction

UNIFIED Speech and Audio Coding [1] (USAC) is an emerging coding standard which is aimed at efficiently coding both speech and music signals. As we all know, the coding methods are different for speech and audio in order to maintain the specific features of them. The USAC allows dynamic switching between different coding modes of speech and audio. The system distinguishes between audio and speech signals and applies different core code and Spectral Bandwidth Replication (SBR) algorithms to them.

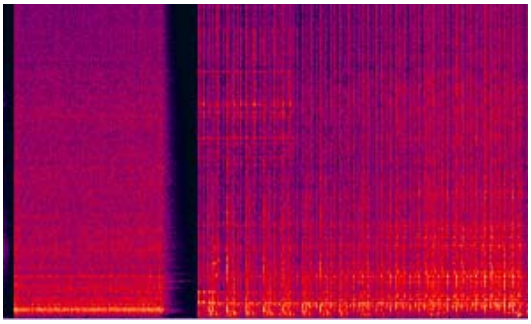
There are two different SBR algorithms in the encoder: one is CT-SBR [3] and the other Harmonic Spectral Bandwidth Replication [4] (H-SBR). A speech/audio classifier sends speech segments to the CT-SBR module and audio segments to H-SBR. However, percussion, which belongs to audio signals, is not fit for the H-SBR module because that H-SBR stretches the harmonic part in low frequency region and copies it to the high frequency region. Compared to orchestra, percussion has a spectral similar to noise and has little harmonic part. Consequently, CT-SBR is more appropriate for percussion than H-SBR.



(a)



(b)



(c)

Fig. 1 Spectrum of two typical music pieces: (a) Wind Music; (b) String Music; (c) Percussion

This work was supported by the National Natural Science Foundation of China under Grant No. 61171171.

Yueming Wang is with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, CO 200240 China (phone: +86-188-1751-9397; e-mail: wymatcn@sjtu.edu.cn).

Rendong Ying, Sumxin Jiang, and Peilin Liu are with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, CO 200240 China (e-mail: RDying@sjtu.edu.cn, microsum2005@sjtu.edu.cn, liupeilin@sjtu.edu.cn).

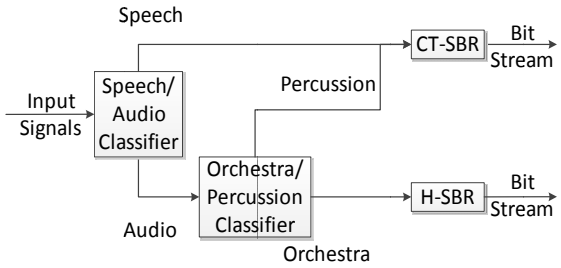


Fig. 2 An overview of the classifier designed for USAC

The spectral of wind music, string music and percussion music is shown in Fig. 1. As shown in Fig. 1, typical orchestra spectrum has distinct harmonics while typical percussion spectrum is more like the spectrum of noise which has almost all the frequencies. (Though the triangle is a type of percussion instrument, it has a spectrum consists of harmonics. Thus, in this paper, it is classified as the orchestra audio.) As a result, an orchestra/percussion classification algorithm is a desideratum, and the classifier of USAC can be improved to be what is shown in Fig. 2. The orchestra/percussion classifier is applied only to audio signals.

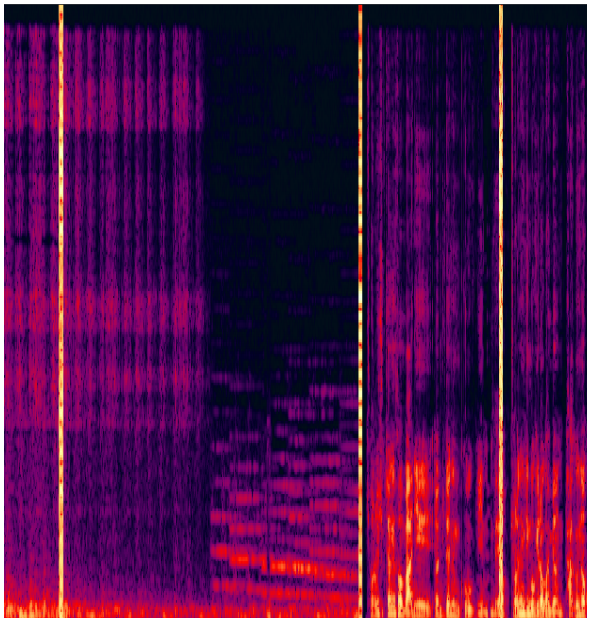


Fig. 3 An example of switching noise

In General, the classification algorithm consists of two parts: the feature extracting part and the classifying part. As with audio features,they can be classified into temporal, spectral and perceptual features. However, ordinary temporal and spectral features are not able to coincide with human auditory system, thus some scholars propose features in Bark domain [8] or Mel domain [13]. Though Bark and Mel domain are simply extensions of ordinary frequency domain, they are more appropriate to fit the human auditory systems. There are large numbers of feature extracting methods, specifically, Mel-frequency cepstral coefficients [5] (MFCCs), MPEG-7 [6] multimedia description, feature extraction based on wavelet [7], Fuzzy list based on Bark domain [8] and so on. To increase the accurate rate of the classification, we choose the MFCCs as our feature.

MFCCs are universality applied in speech processing, tone type classification and instrument classification. Hyoung-Gook Kim and Thomas Sikora [9] classify the test audio documents using Maximum likelihood hidden markov model and reaching an accurate rate of 93.24%. Compared to another widely-used feature, which is MPEG-7, the vectors of MFCCs features have a dimension smaller than that of MPEG-7. They are easier to compute and have a higher accuracy.

As for the classifying part, a lot of learning methods have been proposed. For example, K nearest neighbor (KNN) [2], Support Vector Machine (SVM) [10] and so on. However, they have large computation complexity thus cannot be applied to real-time systems. To limit the computation complexity, decision tree method is a good choice. C4.5 [11] and ID3 [12] decision tree are widely used. They have low computation complexity and is easy to use in real-time systems.

The existing audio classification algorithms call for high computation complexity, thus are hard to be applied in real-time systems. In addition to the shortage of long time delay, their sensitivity to percussion or orchestra mode changing is another severe problem. Orchestra/percussion mode changing leads to the changing of high-frequency reconstruction module, suggesting the changing of CT-SBR and H-SBR module, which may lead to the formation of switching noise. Switching noise may lead to a significant quality loss in the decoded audio file. As is shown in Fig. 3, the long and bright lines are the switching noise, which have large energy and almost all frequencies. In conclusion, a low-complexity, efficient orchestra/percussion classification algorithm which is not too sensitive to sudden changing of reconstruction mode is appropriate for the USAC system.

In this paper, we propose an orchestra/percussion classification algorithm which extract a few scales of MFCCs and apply an ID3 decision tree to decide whether the frame is orchestra or percussion. The algorithm has a fine accurate rate and the computation complexity is small.

II. Orchestra/Percussion Classification Algorithm

Just as most other algorithms, the orchestra/percussion classification algorithm consists of two dominating parts: feature extracting part and classifying part.

A. Feature Extracting

Considered that MFCC [5] is widely used in speech recognition and pattern recognition, we use different scales of MFCCs as the audio feature. The algorithm for extracting MFCCs is shown below,

Algorithm 1: Mel-Frequency Cepstral Coefficients

1: Obtain a frame of audio signal and apply hamming window to the frame. The windowed frame is denoted as $x(n)$

2: Obtain the Fourier Transform of $x(n)$ and denote the result by $X(n)$, i.e. $X(n) = FFT[x(n)]$.

3: Calculate the power spectrum of $X(n)$, i.e. $P(n) = |X(n)|^2$.

4: Filter $P(n)$ with a series of triangle filters. The frequency response of the filers are defined as follows:

$$H_m(K) = \begin{cases} 0 & ,k < f(m-1) \\ \dfrac{2(k - f(m-1))}{(f(m+1) - f(m-1))(f(m) - f(m-1))} & ,f(m-1) \le k < f(m) \\ \dfrac{2(f(m+1) - k)}{(f(m+1) - f(m-1))(f(m+1) - f(m))} & ,f(m) \le k \le f(m+1) \\ 0 & ,k \ge f(m+1) \end{cases}$$

where $\sum_{m=0}^{M-1} H_m(k) = 1$

5: Take the logarithm of the filtered results, i.e. $S(m) = \ln\left(\sum_{k=0}^{N-1} |X(k)|^2 H_m(k)\right), 0 \le m < M$.

6: Calculate the MFCC, i.e. $C(n) = \sum_{m=0}^{N-1} S(m) \cos\left(\frac{\pi n(m-0.5)}{M}\right), 0 \le n < M$.

where n indicates the scale of the MFCC

MFCC is widely used as a kind of audio feature, however, most algorithms use 13 scales of MFCCs, which can lead to relatively high computational complexity. To reduce the complexity, in this paper, we use only 3 scales of MFCCs.

B. Classification

To reduce the computation complexity of the algorithm, we choose ID3 decision tree. ID3 is a kind of learning method to generate a decision tree. It has long been studied. The algorithm can be studied from [17], thus in this paper, we won't introduce the algorithm in details.

In the proposed orchestra/percussion classification algorithm, we extract the MFCCs from training set where the audio signals have been classified manually (thus the MFCCs are classified as well). Afterwards, ID3 algorithm obtains a set of classified MFCCs and generates a learned ID3 decision tree.

C. The Orchestra/Percussion Algorithm

The orchestra/percussion algorithm is the combination of MFCC extraction and ID3 tree decision. For ID3 tree is off-line, training files are needed to generate the decision tree. The training files are framed, windowed and MFCCs extracted. In the training, there are two attributes: orchestra and percussion. The MFCCs of the training files are labeled with the two attributes. We apply the ID3 algorithm to the MFCCs and get a trained tree. The algorithm is described in Algorithm 2.

Algorithm 2: Orchestra/Percussion Classification Algorithm

Training:

1: Initialization:
Obtain audio signals and denote them by *sig _ data*
Initialize the MFCCs set, i.e. $M = \{MFCC_{frame1}, ..., MFCC_{frameN}\} = 0$
Initialize the attributes set, i.e. $A = \{a_1 = orchestra, a_2 = percussion\}$
Initialize an empty decision tree, i.e. $T = NULL$

2: Extract the MFCCs of *sig _ data* using and store the MFCCs in to set M

3: Label the MFCCs in M with attributes in A and generate the decision tree T

Testing:

1: Initialization:
Obtain a frame of audio signal, and denote it by *x*
Obtain trained tree into *Trained _ T*
Initialize MFCC variable, i.e. $MFCC = 0$

2: Apply hamming window to *x*

3: Extract the MFCCs of *x* and store it into *MFCC*

4: Apply the trained tree *Trained _ T* to *MFCC* and get the decision results, i.e. $Trained\_R = a_1 = orchestra$

5: Post-processing, which is described in Fig. 4

The algorithm has two parts including training part and testing part. The training part is mainly to train the ID3 decision tree, and the testing part is to apply the trained decision tree to extracted MFCC thus get the attribute (*orchestra* or *percussion*).

Do to the short comings of the USAC system, frequent changes of attributes.The post-processing is toeliminate the frequent changes of attributes thus the post-processing is actually a smooth algorithm. The smooth algorithm isdescribed in Algorithm 3.

Algorithm 3: Smooth Algorithm

1: Initialization:
Obtain a frame of audio signal and denote it by x
Initialize $Energy = 0$
Initialize 2 empty FIFO arrays *buffer_te*and*buffer_mo*, i.e.
$buffer\_te = [NULL, NULL, NULL, NULL, NULL]$
$buffer\_mo = [NULL, NULL, NULL, NULL, NULL]$
Obtain the attribute of present frame and denote it by PA, i.e. $PA = a_2 = percussion$
Initialize attribute of previous frame $LA = a_1 = orchestra$
2: Store PA into the end of $buffer\_te$ and $buffer\_mo$ and calculate the energy of the frame and store it to $Energy$, i.e. $Energy = \|x\|_2$
3: If two FIFO arrays are not full
Do 2
Else if two FIFO arrays are full
Do 5
End if
4: If $Energy < ET$ (where ET is a threshold)
$PA = LA$
Modify the latest atoms of buffers into attribute in PA
End if
5: If the number of some attribute is larger than 3
$PA = LA$
Modify the latest atom in $buffer\_te$ into attribute in PA
End if
6: If the attributes in $buffer\_te$ change more than 2 times
$PA = LA$
Modify the latest atoms of buffers into attribute in PA
End if
7: If not EOF
Do 2
End if

In Algorithm 3, the sizes of *buffer_te*and*buffer_mo*, the threshold of the number of some attribute (in this paper, the threshold is 3) and the threshold of the change times of attributes in *buffer_te* (in this paper, the threshold is 2) are determined after many times of experiments to ensure the smooth algorithm can reach the best performance.

III. Experiments

In this section, we present a serious of experiments to show the effectiveness of the proposed strategies by 1) using different number of MFCC scales, 2) using different MFCC scales and 3) removing the post-processing module to test and evaluate the accuracy of the algorithm. Training audio files (16bit, 48kHz) and testing audio files (16bit, 48kHz) are provided by Huawei, embracing 8 training and 15 testing audio files, with 4 training orchestra files, 4 training percussion files and others testing files.

Fig. 4 is the test result of the accuracy rate comparison of three pieces of different .wav files. In this test, we choose one percussion file; one orchestra file and one speech file to test the effect of the classification effect without smoothing. We can see from Fig. 4 that the classification performs well on percussion files. To enhance the effect of the classification algorithm, we are focused on enhancing the classification effect on orchestra files.

To determine which scales of MFCC and how many decision tree levels are to choose, we make some tests. Fig. 5 shows the accurate rates of the classification using MFCCs with different numbers of scales. It is obvious that larger number of scales and tree levels lead to better performance of the classification.

However, Fig. 5 shows the lower scales of MFCCs. How about using higher scales? Fig. 6 shows the accurate rates of the classification using merely 3 scales of MFCC. The legend shows which three scales are used.

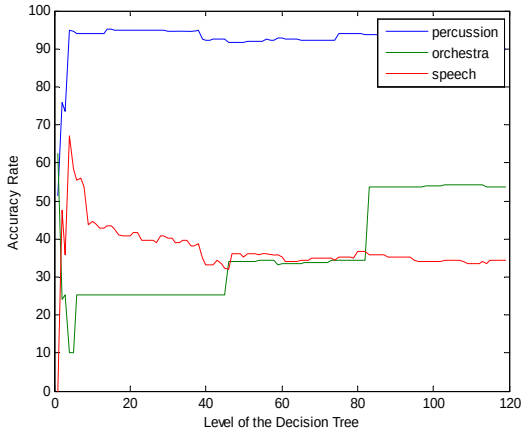


Fig. 4 Accurate rate comparison of percussion, orchestra and speech

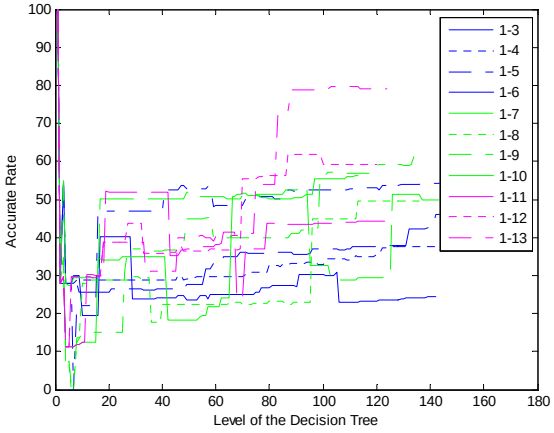


Fig. 5 Accurate rate using 13 scales of MFCC without smoothing

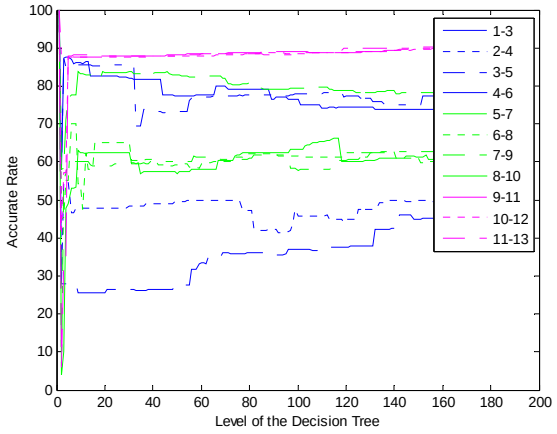


Fig. 6 Accurate rate using 3 scales of MFCC without smoothing

It is obvious in Fig. 6 that using higher scales of MFCCs can reach better performance than using lower ones. Compared to Fig. 5, the results of the decision become stable using remarkable fewer tree levels than those using more scales of MFCCs.

The experiments above indicate that fewer scales of MFCCs and decision tree levels can be used to classify percussion and orchestra. Based on the conclusion, we use merely 3 scales of MFCCs (from scale 11 to scale 13) and 15 levels of decision tree. The decision tree is described in Fig. 7, where MFCC_1, MFCC_2 and MFCC_3 are 3 different scales of MFCCs, T1~T5 are different thresholds.We use small numbers of MFCCs and tree levels so that the computation complexity is reduced and the delay of the classification can be ignored.

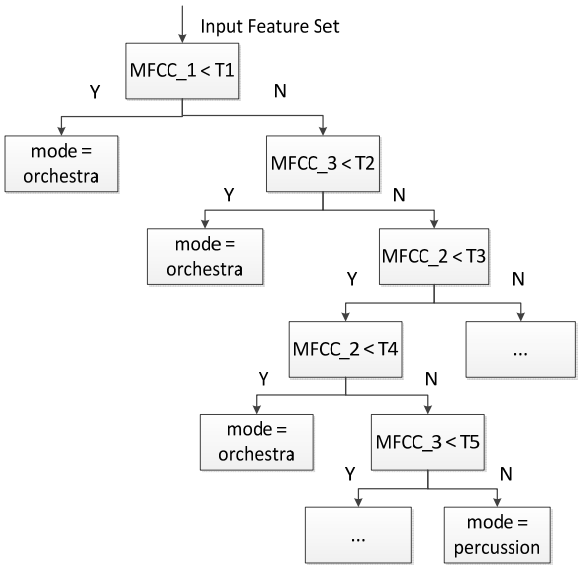


Fig. 7 Generated ID3 decision tree

The tests above optimize the classification algorithm but ignoring the effect of the smoothing part. For the reason that common audio files have little chance that the attributes of orchestra and percussion change frequently, smooth algorithm can not only reduce the appearance of switching noise but also improve the decision accuracy. Fig. 8 shows that the smoothing improves the accuracy by about 10%.

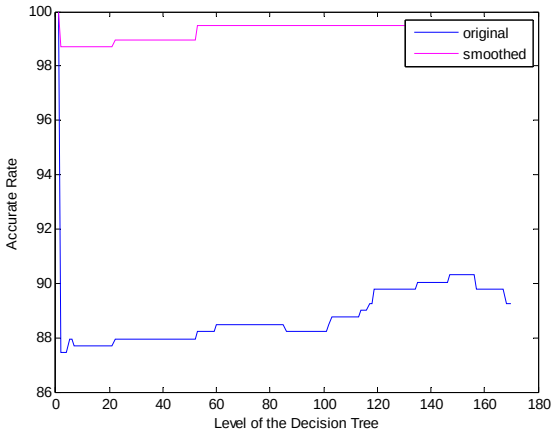


Fig. 8 Accurate rates of orchestra decision with and without smoothing

TABLE I
ACCURATE RATE OF CLASSIFICATION ALGORITHM

".wav" file names	Accurate rate	Description
RefM_tel5	97.75%	percussion
RefM_twinkle_ff51	97.83%	percussion & orchestra
RefM_SpeechOverMusic_4	97.60%	percussion
RefM_phi7	99.74%	orchestra

At last, results of several test files are given in Table I. These files are provided by HUAWEI and are widely used in the cooperation as the test audio files. Table I indicates that the classification algorithm is well performed and can reach an accurate rate of more than 95%.

IV. Conclusion

In this paper, we propose an orchestra/percussion classification algorithm which uses MFCCs and ID3 decision tree to classify audio signals, and a post-processing procedure to further adjust the results. Compared to existing audio classification algorithms, this orchestra/percussion classification algorithm reduces the amount of MFCCs from 13 or more to 3, and retains merely 15 levels of decision tree, thus can reduce the computation complexity by at least 70%.The fast and simple classification algorithm can reach an accurate of more than 97%.Even when the results of classification are unsatisfying, the post-processing module adjusts the results to enhance the accurate rate and at the same time reduce the sensitivity of sudden changing of attributes.

Future works may include further improving of the accuracy rate of classification module by making the decision tree better and faster and may include using more efficient features because FFT is only fit for linear and stationary signals while audio signal are in fact nonlinear and non-stationary signals.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No. 61171171.

References

[1] Jeongook Song, Hyen-o Oh, Hong-Goo Kong, "Enhanced long-term predictor for Unified Speech and Audio Coding", *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp: 505-508, 22-27 May 2011.

[2] Extended Adaptive Multi-Rate-Wideband (AMR-WB+) codec; Transcoding functions (Release 9), 3GPP TS 26.304 V6.2.0, 2005-03.

[3] Martin Dietz, Lars Liljeryd, Kristofer Kjörling and Oliver Kunz, "Spectral Band Replication, a novel approach in audio coding", In 112th *AES Convention*, Munich, May, 2002.

[4] Nagel, F.; Disch, S., "A harmonic bandwidth extension method for audio codecs",*Acoustics, Speech and Signal Processing, ICASSP. IEEE International Conference on*, vol., no., pp.145-148, 19-24 April 2009.

[5] E. Aylon. Automatic detection and classification ofdrum kit sounds. Master's thesis, Universitat PompeuFabra, 2006.

[6] S. Z. Li, "Content-based audio classification and retrieval using thenearest feature line method," *IEEE Trans. Speech Audio Process.*, vol.8, no. 5, pp. 619–625, Sep. 2000.

[7] ISO/IEC Working Group: MPEG-7 overview. URLhttp:// www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm (2004) Accessed8.2.2006.

[8] Lin, C.-C.; Chen, S.-H.;Truong, T.-K.; Chang, Y., "Audio Classification and Categorization Based on Wavelets and Support Vector Machine", *Speech and Audio Processing, IEEE Transactions on*,Volume: 13, Issue: 5,pp. 644-651, Sept. 2005.

[9] Eigenfeldt, A., Pasquier, P. 2009. "Realtime Selection of Percussion Samples Through Timbral Similarity in Max/MSP", *in Proceedings of ICMC.*

[10] Hyoung-Gook Kim, Commun. Syst. Group, Technische Univ. Berlin, Germany Sikora, T."Comparison of MPEG-7 audio spectrum projection features and MFCC applied to speaker recognition, sound classification and audio segmentation", *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, Volume5 pp- 925-8 vol.5, 17-21 May 2004.

[11] J. R. Quinlan, "Learning efficient classification procedures and the irapplication to chess end games", *Machin eLearning: An Artificia lIntelligence Approach*,Vol.1,pp.463-482, Toiga, Palo Alto, CA, 1983.

[12] E. Aylon. "Automatic detection and classification ofdrum kit sounds.", Master's thesis, Universitat PompeuFabra, 2006.

[13] Stevens, Stanley Smith; Volkman; John; Newman, Edwin B. "A scale for the measurement of the psychological magnitude pitch". Journal of the Acoustical Society of America 8 (3): 185–190. 1937.