Automatic Instrument and Environmental Sound Recognition for Media Annotation of TV Content

Sofia Cavaco[⋆] Frederico Malheiro[⋆] João Mateus[⋆] Rui Jesus[†] Nuno Correia[⋆]
[⋆] *CITI, Departamento de Informática, Faculdade de Ciências e Tecnologia*
Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
[†] *Multimedia and Machine Learning Group, Instituto Superior de Engenharia de Lisboa*
Rua Conselheiro Emídio Navarro, 1, 1959-007 Lisboa, Portugal
scavaco@fct.unl.pt, frederico.malheiro@gmail.com, rjesus@deetc.isel.ipl.pt, nmc@di.fct.unl.pt

Abstract

Due to the lack of annotation of their large video archives, multimedia content provider companies and television channels do not use the data in their archives to their full extent. In order to contribute with a solution to this problem, we have developed a tool that combines audio and visual information to annotate video. In particular, this tool has been used by a video production company that has given us positive feedback. The main innovation of this tool is the use of environmental sound recognition to annotate video. Here we focus on the tool's audio information extraction method, which consists of a sound recognizer that learns a small set of spectral features from the data using non-negative matrix factorization. The recognizer can be used for different purposes such as to classify musical instruments, to identify the notes that are played and to distinguish environmental sounds like water, traffic, trains and people.

Key words*: sound classification, musical instruments recognition, environmental sound recognition, non-negative matrix factorization, video annotation*

1. Introduction

In order to optimize the management of the available manpower and reduce the overall costs of multimedia content provider companies and television channels, there is a need for more efficient workflows. While the overall process of obtaining media from the initial production concepts until the archiving phase can be time consuming, the capturing and editing stages correspond to the tasks that have a major impact on the workflow duration. Re-using material available in large video archives, allows sparing the time spent on capturing footage, which, consequently, speeds the workflow processes of those companies. This way, television networks and content provider companies can produce more and better content, in a fast and convenient way.

In order to reuse the material available in video archives, there is a need to annotate the existing material. In many video production companies, this task is still performed manually. This is a hard and tedious job, which, in addition, is prone to depend on human subjectivity. In order to escape from human subjectivity, an automatic approach would be desirable.

We have developed a tool that combines audio and visual information to annotate video automatically. The developed tool has been included in the workflow of a video production company, Duvideo. We use face detection, color descriptors (marginal HSV color moments) and texture descriptors (Gabor filters and SIFT) to gather visual information that is used to train semantic models for video annotation [1, 2]. While the visual information obtained can be used per se to annotate video, we combine it with audio information. The goal of using both audio and video information is to give the system a better understanding of the content and also to enable better browse and search functionalities. Since the focus of this paper is the audio information extraction method, we will not discuss the visual information extraction method any further. (For examples and details of work in digital video, please refer to TREC video retrieval evaluation, TRECVID [3].)

There has been some previous work on video annotation that combines audio and visual information. The majority of these proposals used information extracted from speech to identify keywords that may help to annotate video. This information can be from, for example, speech recognition of voice annotations [4, 5], speech

 DOI: 10.1109/ICALIP.2012.6376785

recognition of the speech from the video [6, 7], or using speech annotation to create audio-visual stories [8, 9]. Notwithstanding, the main contribution of this paper is the use of the environmental sound to annotate video, which is a much less explored problem. We combine features extracted from the video's environmental sound (such as traffic noise or sounds from crowds) with features extracted from its visual content. Jiang et al. also combined features extracted from the video's environmental sound with visual features [10]. They build discriminative audio-visual codebooks by using the multiple instance learning technique. They extract local color and texture from the video segments, along with audio features extracted from the sound track. Then, they combine the visual and audio information into audio-visual atoms and train several concepts using this information. On the other hand, we train several concepts separately using only image features and only audio information, and we give the user the option of using only one modality or both.

In order to extract content information from the audio data available in large video archives, we propose a sound recognizer that has proven to be suitable to deal with different types of data, such as samples from musical instruments and environmental sounds. When dealing with these types of data, two main approaches are possible. The first focuses on developing methods of retrieving, directly from the audio signal, information used to identify the instruments or other sources present in the mix [11, 12]. The second uses statistical sound source separation algorithms, such as independent component analysis, matching pursuit and non-negative matrix factorization (NMF), to learn sound features that can characterize the instruments, sources or notes in the signal [10, 13–18].

The proposed recognizer uses this second approach. The advantage of this approach is that it does not use a set of pre-defined features, such as Mel frequency cepstral coefficients or other short time features, instead it learns the set of spectral features that can better describe the data. The recognizer uses NMF to learn a reduced set of spectral features from sounds and a *k* nearest neighbor (*k*-NN) classifier to classify the data. The recognizer achieves excellent recognition rates for some of the classes present in the environmental sounds extracted from a multimedia content production company's videos (Duvideo). These sounds have the particularity of having a noisy nature, which adds to the difficulty of the problem. Moreover, the recognizer is not limited to environmental sounds. It can be used to classify other types of sounds. In particular, it also achieves very high recognition rates with musical samples when determining the instrument of the sample as well as the note being played.

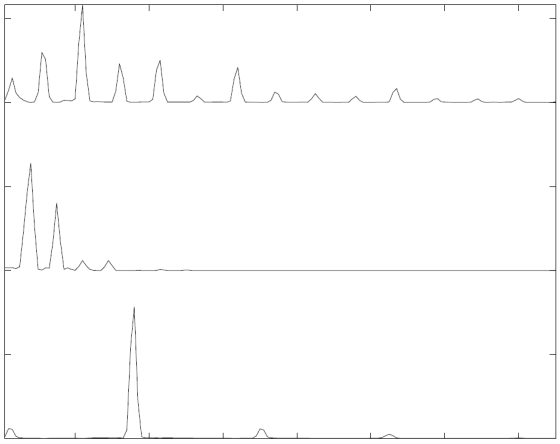

Figure 1. The spectral basis functions (or features) in Θ learned by NMF of the spectrograms of three different notes ($A5_{flute}$, $F3_{guitar}$ and $C4_{piano}$): $\theta_{C4_{piano}}$ (top row), $\theta_{F3_{guitar}}$ (middle row), and $\theta_{A5_{flute}}$ (bottom row).

2. The Recognizer

The proposed recognizer does not use a set of predefined features, instead, it learns them from the data using NMF[1]. In the training phase, the recognizer starts by normalizing the amplitude of the sounds, computing their magnitude spectrograms[2] (namely, $\mathbf{S}_1$ to $\mathbf{S}_\mathrm{N}$), and concatenating all the spectrograms to produce a single matrix $(\mathbf{S}_1, \ldots, \mathbf{S}_\mathrm{N})$ [3], where N is the number of sound samples.

The NMF of the concatenated magnitude spectrograms results into two matrices: $\mathbf{\Theta}$, the mixing matrix, whose columns consist of the spectra that characterize the sounds in the training data set, and $\mathbf{P}$, the source matrix, whose lines contain the temporal envelopes of the sounds. Using these matrices, the training data set $(\mathbf{S}_1, \ldots, \mathbf{S}_\mathrm{N})$ can be expressed as:

$$(\mathbf{S}_1, \ldots, \mathbf{S}_\mathrm{N}) = \mathbf{\Theta}\mathbf{P}\,. \tag{1}$$

The spectra in the columns of $\mathbf{\Theta}$ are the sound features that later will be used in the classification stage of the

[1] Our code was developed in MATLAB and we used an NMF software package by Virtanen [17].

[2] We do not use the spectrograms' phase information. For simplicity, in the remaining text we refer to the magnitude spectrogram as spectrogram.

[3] ($\mathbf{A}$, $\mathbf{B}$) represents two concatenated matrices.

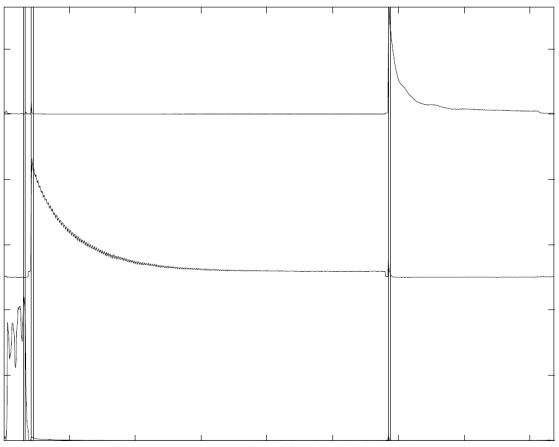

Figure 2. The temporal envelopes in $\mathbf{P}$ obtained by NMF of the spectrograms of three different notes ($A5_{flute}$, $F3_{guitar}$ and $C4_{piano}$). The three feature vectors found are indicated by the three grey rectangles.

recognizer. The temporal envelopes in $\mathbf{P}$ contain the values of these features. In other words, the columns of $\mathbf{\Theta}$ are spectral basis functions that define a new space where the data is now represented. The envelopes in $\mathbf{P}$ are vectors of coefficients, which are the coordinates of the data in this space, that is, each time frame[4] in $(\mathbf{S}_1, \ldots, \mathbf{S}_\mathrm{N})$ is now represented in this new space of spectral basis functions by a new vector of coefficients.

To illustrate these concepts, let us consider a simple scenario in which the training set contains only three samples: the notes *A5* from flute, *F3* from guitar and *C4* from piano. NMF of the matrix of concatenated spectrograms $(\mathbf{S}_{A5_{flute}}, \mathbf{S}_{F3_{guitar}}, \mathbf{S}_{C4_{piano}})$ produces a matrix $\mathbf{\Theta}$, represented in Figure 1 (for simplicity, we have plotted the transpose of $\mathbf{\Theta}$ in the figure). Each row in this figure consists of one of the columns from $\mathbf{\Theta}$ and defines one spectral feature, which here we call $\boldsymbol{\theta}_{C4_{piano}}$, $\boldsymbol{\theta}_{F3_{guitar}}$, and $\boldsymbol{\theta}_{A5_{flute}}$.

Along with $\mathbf{\Theta}$, NMF also produces a matrix $\mathbf{P}$, which, for the example used above is represented in Figure 2. There is a correspondence between the envelopes in this figure and the spectra in Figure 1. The envelopes in the ith row are associated to the spectrum in the ith row of Figure 1. Each line in Figure 2 consists of one of the lines from $\mathbf{P}$, where the ith line contains the feature values of feature $\boldsymbol{\theta}_i$. This way, a time frame from the spectrograms in the data set is represented by one column of $\mathbf{P}$ (that is, a vector with one coefficient for each of the spectral features in $\mathbf{\Theta}$).

If we consider all the coefficients related to the frames of one spectrogram and one basis function, we have a vector of coefficients, which is a temporal envelope. Now we can define matrix $\mathbf{P}_n$ which contains all the vectors of coefficients (or temporal envelopes) that are associated to spectrogram $\mathbf{S}_n$. In this way, the data set can be expressed as:

$$(\mathbf{S}_1, \ldots, \mathbf{S}_\mathrm{N}) = \mathbf{\Theta}\,(\mathbf{P}_1, \ldots, \mathbf{P}_\mathrm{N})\ . \qquad (2)$$

The ith row of every matrix $\mathbf{P}_n$ is associated to the ith basis function (the ith column) in $\mathbf{\Theta}$.

Considering the example again, since matrix $\mathbf{P}$ contains the temporal envelopes associated with three sounds, it can be represented as $\mathbf{P} = \left(\mathbf{P}_{A5_{flute}}, \mathbf{P}_{F3_{guitar}}, \mathbf{P}_{C4_{piano}}\right)$, where $\mathbf{P}_x$ contains the temporal envelopes of note x. If we inspect Figure 2 more carefully, we can see that each row contains 3 temporal envelopes: one for each of the 3 sounds used in the example. For instance, the first row contains the envelopes $\mathbf{p}_{1,\,A5_{flute}}$, $\mathbf{p}_{1,\,F3_{guitar}}$, and $\mathbf{p}_{1,\,C4_{piano}}$, which are the first rows of $\mathbf{P}_{A5_{flute}}$ $\mathbf{P}_{F3_{guitar}}$ and $\mathbf{P}_{C4_{piano}}$ respectively. In particular, we can observe that $\mathbf{p}_{1,\,C4_{piano}}$ has more energy than the other two envelopes, which suggests that $\boldsymbol{\theta}_1$ (which we have named $\boldsymbol{\theta}_{C4_{piano}}$) describes a property of the $C4$ piano sample. The envelopes are time varying functions that describe how the sounds evolve with time. For instance, $\mathbf{p}_{1,\,C4_{piano}}$ indicates that at the attack the $C4$ piano sample's spectrum can be obtained by multiplying $\boldsymbol{\theta}_{C4_{piano}}$ by a strong coefficient, but as time passes lower coefficients are used to obtain the spectra, eventually reaching zero, at the end of the decay portion of the sound. Notice that this way we obtain a sequence of spectra that when put together actually form a spectrogram ($\mathbf{S}_{C4_{piano}}$).

By inspecting the rest of Figure 2, we can see that the temporal envelope that is active in the middle row corresponds to the second note in the sequence, that is, $F3_{guitar}$, which means that the middle spectra in Figure 1 describes this note. Finally, the temporal envelope that is active in the bottom row in Figure 2 corresponds to the first note in the sequence, that is, $A5_{flute}$, which indicates that the bottom spectra in Figure 1 accounts for this sound.

The feature vectors used for training the classifier are built using the values in the vectors $\mathbf{P}_n$. A feature vector $\boldsymbol{v}$ is an M dimensional vector, where M is the number of basis functions in $\mathbf{\Theta}$ (that is, the number of features). The definition of $\boldsymbol{v}$ depends on the type of sounds used. For instance, for musical sounds, most envelopes in the matrices $\mathbf{P}_n$ have an attack portion followed by a sustain and decay portions. We can use the peaks of these envelopes (which are located at

[4] A time frame is a column of the spectrogram.

the end of the attack and right before the sustain) to build $\boldsymbol{v}$. In more detail, when a peak is found in $\mathbf{P}_n$, the algorithm looks for a peak in the other lines of $\mathbf{P}_n$, such that the peaks are in close proximity, that is, the algorithm analyses only a limited number of columns (determined by Δi) such that the peaks correspond to the same event (such as the attack of a note). Thus, $\boldsymbol{v}_{i,n} = (v_{1,i,n}, \ldots, v_{M,i,n})$ is a vector of coefficients associated to a small neighborhood of the ith frame of spectrogram $\mathbf{S}_n$: $v_{m,i,n} \in \{p_{m,i-\Delta i,n}, \ldots, p_{m,i+\Delta i,n}\}$, where $p_{r,c,n}$ is the coefficient in the rth row and cth column of $\mathbf{P}_n$. When the envelopes in $\mathbf{P}_n$ are not well defined, such as the envelopes of noisy sounds (like from cars, crowds, flowing water, etc.), instead of using directly the coefficients in $\mathbf{P}_n$ to build $\boldsymbol{v}$, we use the average value, the median and the energy of the envelopes.

This way, N training feature vectors are built using the values in $\mathbf{P}_1, \ldots \mathbf{P}_N$ (one for each sound in the training set). The training feature vectors are used to train a *k*-NN classifier that uses the Euclidean distance metric. While we could extract several training feature vectors from each sound (for example, some from the attack, some from the sustain and decay parts of the sound), here we use only one such vector per sound.

Going back to our example again, the next step consists of evaluating $\mathbf{P}$ to determine the largest peak in each of the submatrices ($\mathbf{P}_{A5_{flute}}, \mathbf{P}_{F3_{guitar}}$, and $\mathbf{P}_{C4_{piano}}$) to create three training feature vectors (one for each sound in the training set) as was described above. The training feature vectors created by the algorithm are marked by the grey rectangles in Figure 2. Essentially, each feature vector represents a single note, and these are the vectors used to train the *k*-NN. Note that while we mentioned that for most musical sounds we use the peaks from the attack portion of the sound, the flute sound shows different characteristics from the other two sounds in the example and therefore its feature vector does not correspond to the attack portion of the sound: the active temporal envelope of the flute sound (first envelope in bottom row of Figure 2) looks very different from the active envelopes of the other two sounds, and it can also be observed that the spectral feature that characterizes this sound ($\boldsymbol{\theta}_{A5_{flute}}$) is almost sinusoidal.

A test feature vector consists also of an M dimensional vector of coefficients extracted from M temporal envelopes, but while the envelopes in the training set are learned by NMF of the spectrograms, the envelopes in the test set are determined by the following equation:

$$\mathbf{P}_{\text{test sound}} = \boldsymbol{\Theta}^{-1}\mathbf{S}_{\text{test sound}} \ , \qquad (3)$$

where $\boldsymbol{\Theta}^{-1}$ is the pseudoinverse of $\boldsymbol{\Theta}$.

So, in order to classify a test sound, the algorithm starts by representing it in the same space as the training feature vectors: it normalizes the amplitude, computes the spectrogram, and through equation 3 obtains a matrix $\mathbf{P}_{\text{test sound}}$, with the coefficients of the test sound. The algorithm then obtains a test feature vector by computing the maximum peaks (for musical sounds) or by computing the average, median and energy in the envelopes (for environmental sounds).

3. Result Analysis

We tested the classifier with two types of data: sounds from musical instruments and from the environment. The environmental samples were extracted from video footage from Duvideo's archive and consist of 0.2 seconds samples. We used a total of 96 segments of flowing water sounds (from rivers, fountains, etc.), 125 segments of moving car sounds, 1400 segments of train sounds (which consist of sounds recorded inside a moving train) and 615 segments of sounds from crowds (in a stadium, etc.).

The musical instrument samples consist of single notes from flute, guitar and piano. The recordings were made with a microphone (AKG C1000S) connected to an USB Audio Interface (Edirol UA-25) and all recordings were digitized using a sampling frequency of 44100 Hz. In all, 48 notes (spanning four octaves) were recorded, of which 12 are guitar (*C3* to *B3*), 24 are piano (*F#3* to *F5*) and 12 are flute (*C5* to *B5*). Six different recordings were made for each note using different playing techniques, giving a total of 288 sound samples: 72 guitar samples, 72 flute samples and 144 piano samples.

To test the recognizer with the musical instrument samples, we performed n-fold cross validation experiments. The training sets were created with all the samples except one from each note and instrument (that is, all training sets had 240 sounds). The remaining 48 sounds were used as test data (all the sounds from the same instrument in the test data used the same technique, such as staccato or sustain). Therefore all sounds were used for training at five of the experiments and for testing in one of the experiments. Since the number of samples from the same instrument and note in the training sets is five, we used this number as the value for *k* in the *k*-NN classifier.

The recognition results we obtained were very positive. When dealing with musical sounds, the recognizer can not only identify the instrument but also the note played. Table 1 shows that the accuracy of the system is consistently very high. The columns *Instrument* and *Note* show the number of correctly classified samples in terms of instrument and note, respectively. The last column shows the total number of tested samples. All

Table 1. Classification rates for the musical instrument samples.

	Instrument		Note		Total number
	#	%	#	%	of samples
Flute	70	97%	62	86%	72
Guitar	72	100%	72	100%	72
Piano	137	95%	143	99%	144

Table 2. Confusion matrix for the environmental samples.

	Water	Car	Train	People	Total
Water	75	20	0	1	78%
Car	7	61	6	51	49%
Train	0	342	1053	5	75%
People	0	35	0	580	94%

guitar samples were correctly classified both in terms of note and instrument. The instrument classification was also very high for flute and piano (97% and 95%, respectively) and there was only one piano sample that was classified as the wrong note. The worse results were the note classification of flute, but even so this gave quite an acceptable recognition rate (86%).

The value of k for the tests performed with the environmental sounds was set in a different manner: $k = ceiling(\sqrt{mean(|C_1|, \ldots, |C_4|)}$, where $|C_i|$ is the number of samples from class C_i. The results were excellent for the class people with 94% recognition rate and also good for the classes water and train (see Table 2). Even though we had more car segments than water, the recognition rate of car sounds was the lowest. This may be due to some existing similarities between car sounds and large crowds' sounds (note how there are also some people's sounds misclassified as car sounds), which suggests that more car samples would be necessary to improve the recognition capability of these sounds.

4. Conclusions

The main contribution of this paper is the use of non speech sounds to annotate video. For that end, we proposed a sound recognizer that instead of using a set of predefined features, it learns a set of spectral features from the data using NMF. The recognizer has been included in the workflow of a multimedia content provider company to annotate video.

We have tested the recognizer with environmental sounds extracted from a real video archive and obtained very high recognition rates for some of the classes tested. It is worth to note that the environmental sounds used have a noisy nature which adds to the difficulty of finding features that successfully separate them.

The recognizer is not restricted to environmental sounds; it can also deal with other types of data, like music samples. When tested with musical instrument samples, this approach proved very successful: very high recognition rates (from 95% to 100%) were obtained for all tested instruments. In addition, the recognizer is also able to identify the note being played: again, very high recognition rates (from 86% to 100%) were obtained for samples spanning four octaves.

As future work we plan to use more samples from the real video archive that include other classes of sounds. We also plan to extend the recognizer with a pre-processing module which decides if the signal being analyzed is a musical sound or an environmental sound, such that the recognizer can decide what type of feature vector to use, that is, how to process the activation values in the envelopes learned by NMF (a vector with the peak activation values near the note onset for musical sounds, or a vector with the average value, median and energy of the activations for environmental sounds). This distinction can be done by analyzing the rhythmic content in the data. The rhythmic content has been used on music genre classification and contains information such as the beat, the tempo, the regularity of the rhythm and time signature [19–22]. Another possible extension of this work is to combine it with a speech recognizer so that both information from the environment and speech are used to annotate the data.

Acknowledgments

This work was part of the Videoflow project (3096) funded by *Quadro de Referência Estratégica Nacional* (QREN) and *Fundo Europeu para o Desenvolvimento Regional* (FEDER) through the *Programa POR Lisboa*.

We thank the professionals at Duvideo for their help and cooperation on all the life cycles of the project. We would also like to thank Luís Gomes for lending us the equipment necessary to record the musical instrument samples, and Ricardo Andrade for giving us access to a piano in *Faculdade de Ciências Sociais e Humanas, Universidade Nova de Lisboa* and helping recording the piano samples.

References

[1] R. Jesus, A. Abrantes, and N. Correia, "Methods for automatic and assisted image annotation," *Multimedia Tools and Applications*, vol. 1380:7501, pp. 1–20, 2010.

[2] J. Mateus, F. Malheiro, S. Cavaco, N. Correia, and R. Jesus, "Video annotation of TV content using audiovisual information," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS)*, 2012.

[3] P. Over, G. Awad, M. Michel, J. Fiscus, W. Kraaij, and A. Smeaton, "Trecvid 2011 - an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TREC Video Retrieval Evaluation*, 2011.

[4] K. Rodden and K. Woods, "How do people manage their digital photographs?," in *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, 2003, pp. 409–416.

[5] T. Mills, D.Pye, D Sinclair, and K. Wood, "Shoebox: a digital photo management system," Tech. Rep., AT&T Research, 2000.

[6] V. Tseng, J.-H. Su, J.-H. Huang, and C.-J. Chen, "Integrated mining of visual features, speech features, and frequent patterns for semantic video annotation," *IEEE Transactions on Multimedia*, vol. 10, no. 2, pp. 260–267, 2008.

[7] L.Xie, L. Kennedy, S.-F. Chang, and C.-Y. Lin A. Divakaran, H. Sun, "Discovering meaningful multimedia patterns with audio-visual concepts and associated text," in *Proceedings of IEEE International Conference on Image Processing*, 2004, pp. 2383–2386.

[8] T. Jokela, J. Lehikoinen, and H. Korhonen, "Mobile multimedia presentation editor: enabling creation of audio-visual stories on mobile devices," in *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, 2008, pp. 63–72.

[9] D. Frohlich and M. Jones, "Audiophoto narratives for semi-literate communities," *ACM Interactions*, vol. 15, no. 6, pp. 61–64, 2008.

[10] W. Jiang, C. Cotton, S.-F. Chang, D. Ellis, and A. C. Loui, "Audio-visual atoms for generic video concept classification," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 6, no. 3, 2010.

[11] W. Jiang, A. Wieczorkowska, and Z. W. Ras, "Music instrument estimation in polyphonic sound based on short-term spectrum match," in *Foundations of Computational Intelligence*, A.-E. Hassanien, A. Abraham, and F. Herrera, Eds., vol. 2 of *Studies in Computational Intelligence Vol. 202*, pp. 259–273. Springer, 2009.

[12] X. Zhang and Z. W. Ras, "Sound isolation by harmonic peak partition for music instrument recognition," *Fundam. Inf.*, vol. 78, no. 4, pp. 613–628, 2007.

[13] S. Cavaco and M.S. Lewicki, "Statistical modeling of intrinsic structures in impact sounds," *Journal of the Acoustical Society of America*, vol. 121, no. 6, pp. 3558–3568, June 2007.

[14] E. Benetos, M. Kotti, and C. Kotropoulos, "Musical Instrument Classification Using Non-Negative Matrix Factorization Algorithms and Subset Feature Selection," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2006, vol. 5, pp. 221–224.

[15] P. S. Lampropoulou, A. S. Lampropoulos, and G. A. Tsihrintzis, "Musical instrument category discrimination using wavelet-based source separation," in *New Directions in Intelligent Interactive Multimedia*, pp. 127–136. Springer, 2008.

[16] L.Martins, J. Burred, G. Tzanetakis, and M. Lagrange, "Polyphonic instrument recognition using spectral clustering," in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, 2007.

[17] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 1066–1074, 2007.

[18] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence. with application to music analysis," *Neural Computation*, vol. 21, no. 3, pp. 793–830, 2009.

[19] L. Barreira, S. Cavaco, and J.Ferreira da Silva, "Unsupervised music genre classification with a model-based approach," in *Proceedings of the 15th Portuguese Conference on Artificial Intelligence (EPIA)*, L. Antunes and H. S. Pinto, Eds. 2011, vol. 7026 of *Lecture Notes in Computer Science*, pp. 268–281, Springer-Verlag.

[20] A.L. Koerich and C. Poitevin, "Combination of homogeneous classifiers for musical genre classification," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 2005, vol. 1, pp. 554–559.

[21] C. Lee, J. Shih, K. Yu, and H. Lin, "Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features," *IEEE Transactions on Multimedia*, vol. 11, no. 4, pp. 670–682, 2009.

[22] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.