Independent Component Analysis for Music Similarity Computation

Tim Pohle[1], **Peter Knees**[1], **Markus Schedl**[1], **Gerhard Widmer**[1,2]
[1]Johannes Kepler University Linz, Austria
[2]Austrian Research Institute for Artificial Intelligence (OFAI), Vienna, Austria
music@jku.at

Abstract

In the recent years, a number of publications have appeared that deal with automatically calculating the similarity of music tracks. Most of them are based on features that are not intuitively understandable to humans, as they do not have a musically meaningful counterpart, but are merely measures of basic physical properties of the audio signal. Furthermore, most of these algorithms do not take into account the temporal development of the audio signal, which certainly is an important aspect of music. All of them consider the musical signal as a whole, not trying to reconstruct the listening process of dividing the signal into a number of sources.

In this work, we present a novel approach to fill this gap by combining a number of existing ideas. At the heart of our approach, Independent Component Analysis (ICA) decomposes an audio signal into individual parts that appear maximally independent from each other. We present one basic algorithm to use these components for similarity computations, and evaluate a number of modifications to it with respect to genre classification accuracy. Our results indicate that this approach is at least of similar quality as many existing feature extraction routines.

Keywords: audio feature extraction, music similarity computation

1. Introduction

Although “music similarity” is an ill-defined concept, most people have an intuitive idea of which music is similar, and which is not. For example, most people would consider two heavy metal pieces as being more similar to each other than to a classical choir piece. It is obvious that such commonsense judgements are somehow reflected in the musical signal. In recent years, some research has been performed to extract these aspects algorithmically, and to automatically compute the music similarity. Reviews of a number of these approaches can be found in [1, 2]. Among the algorithms that seem to perform best are those that are based on the well-known Mel Frequency Cepstral Coefficients (MFCCS) which were invented for speech recognition [3, 4, 5, 6]. MFCCs only describe the coarse shape of the spectrum, and discard the harmonic structure. Also, in these algorithms the temporal structure is discarded. Approaches to model the development of MFCCs in time have not significantly improved the performance (cf. [7]). Most of the other algorithms that aim to describe certain aspects (“features”) of the audio signal (cf. also [8]) also only operate on very short segments of the audio signal (“frames”) and do not consider their temporal order.

There are only few audio features that take into account the temporal structure of the signal, which certainly is of great importance when regarding music (e.g. beat histograms [9], meter and tempo descriptors [10], and fluctuation patterns [11, 12]).

One thing that is common to these audio similarity algorithms is that they are based on features that only look at the mixed audio signal as a whole, although splitting the audio signal into a number of subbands gives some separation as usually in each band only few instruments are dominant. One approach that has not been tried yet is to simulate the hearing process to a certain extent, and separate the audio signal into various sources before doing a similarity analysis. The purpose of this paper is to take a first step into this direction.

2. State of the Art

In this section, we describe the basic technique we use, Independent Component Analysis (ICA), and give an overview of how it has already been applied to the field of music information retrieval. Also, we motivate why we think ICA is useful for the purpose of this work.

2.1. Independent Component Analysis (ICA)

Independent Component Analysis (ICA) is an approach to solve the following problem: given a number of observed signals that are a linear mixture of a number of unknown signals, find these unknown signals *and* the amount each of them contributes to the observed signals. An example for this scenario is two people speaking simultaneously in a room, which is recorded with two microphones.

2.2. ICA in the field of Acoustics and Music Information Retrieval

In [13], the authors apply ICA to short frames of time-domain audio signals, so that each time-domain frame can be represented by a mixture of independent components. This work was motivated by prior work in image analysis that found independent components to be related to low-level aspects of human vision. The authors find some indication that for certain kinds of audio signals, the resulting independent components may be similar to low-level processing steps in the human auditory system.

In [14], ICA was applied to separate the sources of musical signals, including various drum sounds and vocals. The step of isolating vocals from the other instruments in a song was improved and automated in [15]. The promising results from these works indicate that ICA might be useful to separate the various musical instruments similar to human perception. Thus, ICA seems to be valuable for our experiments presented here. For the field of pitch detection, methods related to ICA were used in [16].

3. Approach

The starting point of our experiments is the combination of a number of existing ideas. We adapt the basic approach for image similarity computation by independent components from [17] to the domain of audio processing. The approach and its adaptation is described in the following sections.

3.1. Original Approach from [17]

For convenience and a better understanding, here first the outline of the original approach is repeated.

3.1.1. Features and Feature Extraction

In the original work [17], feature extraction is done in the following way: after a preprocessing step, a number of randomly chosen excerpts of 12×12 pixels are taken from each image. The authors call these excerpts *patches* [18]. ICA is run on a great number of patches. Each individual patch can be thought of as a linear combination of independent components. To describe a whole image, this image is divided into patches, and each patch is described by the degree of activation of each independent component. The features for the whole image are the activation histograms for each independent component over all patches from the image. Only those histograms are retained whose average activation per picture has the highest variance over a large number of pictures. Two images are compared by comparing their (independent components' activation) histograms, as described in the next section.

3.1.2. Determining the Similarity Between Images

As the activations of the independent components are sparse and highly uncorrelated, interdependencies between the various activation histograms of the independent components can be disregarded. Thus, the comparison of two images reduces to a comparison of the histograms belonging to the same component. The individual similarities then can be combined to obtain one overall similarity value. The authors propose a number of approaches to compare the histograms.

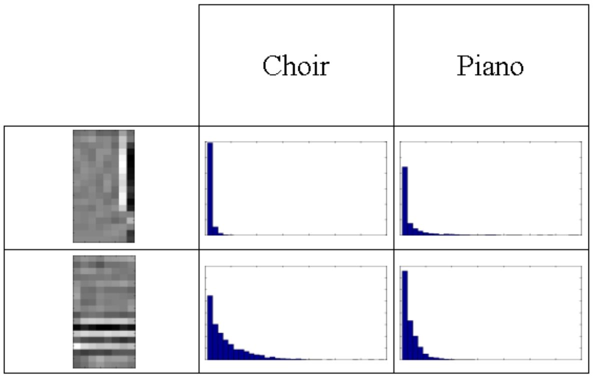

Figure 1. Example of activation histograms for a choir piece and a piece for prepared piano. The two independent components whose histograms are shown are given in the left column. It can be seen that the upper component is more often close to zero in the choir piece than in the piano piece, while the component depicted below is more frequently activated in the in the choir piece than in the piano piece.

- The simplest is to take the mean of all activations of a component over all patches of the image. Thus, the feature data of an image consists of one vector of length n, where n is the number of independent components. The similarity of two images is determined by the Eucledian distance between these vectors.

- A better description of the histogram shapes is achieved by modelling them as a Gaussian distribution, which results in two values per histogram (mean and variance). These values are compared by applying the KL Distance. As the activation histograms of the independent components are known to have a high kurtosis, the authors also introduce an additional flavour of this Gauss model: the activation histogram is mirrored at $x = 0$, and the Gauss model is calculated over the resulting distribution. Obviously, in this case the mean is zero.

- The most exact comparison between histograms was done by modelling each histogram with a B spline, and calculating the common area under the spline models.

The similarity measure was evaluated on a set of 540 images from four categories. Lacking a reliable pairwise similarity judgement, evaluation of the similarity measure was done by nearest neighbour classification, and the classification accuracy was taken as an indicator of the similarity measure's quality. The authors report classification accuracies of up to 87 percent.

3.2. Adoption to Musical Signals

When adopting this approach to music signals, the crucial step is to find an analogue for the image patches (i.e. the 12×12 pixel samples from the images), including a suitable data representation for their extraction. Some authors have chosen a single time domain or frequency domain frame as one "patch". Although we also investigate this definition of a patch in the course of our experiments, we found it more interesting to also take into account the temporal changes of the signal, because the development in time is an important aspect of music, and furthermore already a great number of audio descriptors exist that operate on a per-frame basis. Thus, in our initial experimental setup, we opted to regard a number of N consecutive (frequency-domain) frames as one patch, with the exact value of N being one parameter in our experiments. Each patch covers all frequencies.In [19], this definition of a "patch" was already proposed. Also, it is mentioned there that the great number of frequency bins in a usual spectrum produces problems, as it causes the input vector to the ICA algorithm to be very large (ie. number of frequency bins × number of frames in patch). Because of this, and due to the more perceptually motivated representation, we transformed the spectrum to a mel-sone-representation, which has only 18 frequency bands instead of e.g. 256. These 18 frequency bands roughly correspond to critical bands in the human auditory system.

3.3. Filters as templates for event detection

Of course it would be desirable to obtain independent components ("filters") that represent meaningful aspects of hearing. Examples of such dedicated filters include "percussive sound", "high-frequency noise", "sustained sound in the lower frequencies". Such filters then might serve as templates to scan a given piece of music for these kinds of events. The only publication we are aware of that uses a template-based process for music analysis in the acoustic domain is [20], where it was used for drum sound detection.

4. Experimental Setup

We carried out our experiments with the tracks from the ISMIR'04 Genre Classification Contest training set. This set consists of 724 tracks [1] from the six genres classical (43.7%), electronic (15.9%), jazz/blues (3.6%), metal/punk (6.0%), rock/pop (14.0%), world (16.9%). [2] Feature extraction was done on 30 seconds from the middle of each file. The overall algorithm was as described above: after transforming each track into the mel/sone representation using the MA Toolbox [11], we calculated the independent components on a subset of the collection. In the next step, the components were used to extract the features from each song by determining how strong each component is activated during the song. The final similarity computation depends on the particular histogram comparison method.

4.1. Obtaining the Independent Components

We used the FastICA [3] algorithm to compute the independent components. The data on which we computed ICA were small fragments from the mel/sone representation of the audio tracks. We started the evaluation with three different fragment lengths: 0.15 sec, 0.3 sec, and 0.6 sec. Note that 0.6 sec is the duration of one quarter note when playing a $4/4th$ metre at 100 bpm, and similarly 0.3 sec and 0.15 sec correspond to the durations of $1/8$ and $1/16$ note. Based on the outcome of the experiments, we then also tried other lengths. As the resulting patches contained up to 576 values, calculating ICA on them also produced up to 576 components. To reduce this large number of components, we also applied a dimensionality reduction of the input data by PCA before calculating ICA (the PCA compression is then inverted on the – fewer – individual components afterwards to expand them again to e.g. 576 values per component). This way, we reduced the number of components by 0% (i.e. no PCA), 50% and 75%. We evaluated three ways to obtain ICA components:

1. As a first approach, we created a representative subset of 100 songs from our collection. On these 100 songs, we calculated ICA on randomly chosen fragments. In Figure 2, an example of the resulting components is given.

2. The second approach was like the first approach, with the difference that the patches were not extracted at fully randomly chosen points, but rather starting on those frames that are likely to contain beat onsets.

3. Finally, as an interesting try, we defined the components manually, according to what we subjectively thought to be meaningful entities such as high / mid / low frequency content, beat onsets in various frequencies, periodicities at various levels. An example of these component is given in Figure 3.

The quality of these approaches was evaluated as described in the next section.

4.2. Evaluation

Lacking human judgements about the similarity of each pair of tracks in the collection, we evaluated the algorithm's performance in a similar manner as in the original paper. We assume that tracks that belong to the same genre are more similar to each other than tracks that are labelled to belong to different genres. After calculating the similarity of each pair of tracks, we do a leave-one-out 1-Nearest-Neighbour classification, and take the classification accuracy as an indicator of the algorithm's capability to calculate the perceived similarity of music tracks.

[1] We left out five tracks due to file naming inconsistencies.

[2] The full set is available for download at http://ismir2004.ismir.net/genre_contest/index.htm.

[3] http://www.cis.hut.fi/projects/ica/fastica/

Mel/Sone Patch Length	0.075 sec			0.15 sec			0.3 sec		
PCA compression	1	0.5	0.25	1	0.5	0.25	1	0.5	0.25
Mean	63.8%	62.4%	61.8%	64.2%	63.5%	61.7%	64.2%	62.8%	61.5%
KL	68.5%	66.7%	65.7%	67.4%	68.2%	65.4%	64.5%	64.7%	65.6%
KL_zero	67.4%	66.0%	65.7%	67.2%	66.3%	65.4%	63.6%	64.9%	64.3%

Table 1. ***A patch defined as a short excerpt of the mel/sone representation of the song.*** **Average classification accuracy for a number of patch sizes, for various histogram comparison methods and with varying PCA compression factor. Histogram comparison methods were the Euclidean distance between the means of each histogram (*Mean*), Kullback-Leiber (KL) divergence based on mean and standard deviation of the histograms (*KL*), and KL divergence based on the standard deviation of the histograms, mirrored at the zero-point, which produces a mean of zero (*KL_zero*). The overall maximum accuracy found with this setup was 68.5% at a patch length of 0.075 sec, below which patch length all accuracies tended to decrease.**

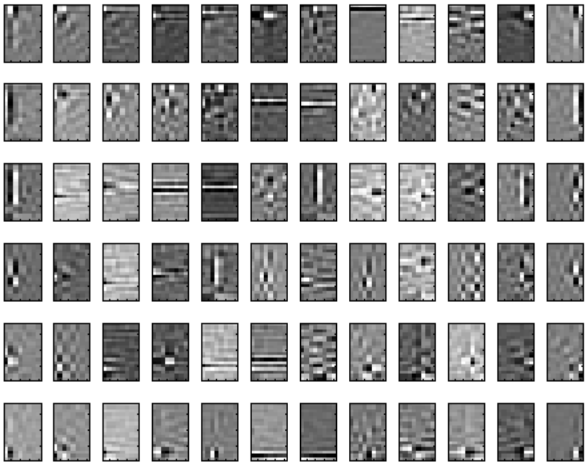

Figure 2. Independent components calculated on patches of length 0.15 sec with randomly chosen starting times, and 50% PCA compression.

5. Results and Discussion

In this section, we give the results of the experiments described above, and give a short interpretation of these results.

5.1. ICs calculated on randomly chosen patches

When calculating the independent components on patches that are randomly chosen, in many cases the components seem not to correspond to meaningful acoustical entities. But in some configurations interesting results appeared. For example, in Figure 2 the components for patches of length 0.15 sec with 50% PCA compression are given. It can be seen that some components are primarily located in time, which can be assumed to be activated on percussive music events. Other components are mainly located in frequency, with a horizontal shape. Components with a horizontal shape might be preferably activated on sustained sounds. We examined the calculated components for such properties by investigating tracks for which components are activated most frequently, and by visualizing the activation of the components over time. Unfortunately, we found no strong indication that these components are related to meaningful musical entities in such a way. But at least comparing certain activation histograms of clearly different pieces give a weak indication that there might be such relationships (cf. Figure 1).

The classification accuracies for $1NN$ leave-one-out evaluation are given in Table 1. The values shown are the result of considering all available components. The accuracies are constantly above 60%, with a maximum value of 68.5%. Although this value is still below the classification accuracy achieved with the algorithm and proposed parameter set from [5], which is at 72%, it is still above the values we achieved with other audio descriptors for simple nearest-neighbor classification [1], and within the range of the accuracies achieved with sophisticated machine learning algorithms applied on a set of many commonly used audio features [1].

Following the practice in [17], we also tried to apply a Hamming window to each patch before calculating the independent components or determining the activation strength of each component, respectively. This additional step did not lead to increased classification accuracy in our experiments.

Another observation we made was that only considering those n components whose average activation per song had the highest variance over a large number of songs produced lower accuracies than considering *all* components. For this, we tried $n = \{1, 2, 5, 10, 15, 20\}$. However, it should be noted that for $n = 20$, the achieved classification accuracy was only a few percentage points below the results obtained when using all components.

5.1.1. Patches Defined as One Frame

As the highest results were achieved with patches of short length ($\leq$ 8 Frames), we also investigated the aforementioned alternative way to define patches to use only one single frame, but with a higher frequency resolution. Except for the differences in the frequency representation, the parameters of the experiment stayed the same. The results for $numFreq = 129$ which are given in Table 2 indicate that this alternative way to define patches does not contribute to a higher classification accuracy.

	1	0.5	0.25	0.125	0.0625
Mean	50.0%	53.3%	53.5%	50.5%	49.1%
KL	58.4%	57.8%	62.8%	58.8%	55.2%
KL_zero	58.1%	58.4%	61.1%	59.6%	51.7%

Table 2. ***A patch defined as one FFT frame with 129 frequencies:*** **Average classification accuracy for the various methods with varying PCA compression factor. Same abbreviations as in Table 1.**

5.2. Patches at Likely Onsets

Instead of trying to model each possible patch with arbitrary start position by independent components, it might be beneficial to only consider patches that start at onset times. This might contribute to making various patches better comparable to each other, as in this case it is known that each patch starts at an onset. We evaluated this by finding possible onset times with a simple onset-detection algorithm. This algorithm was applied to select the patches for calculating the independent components, and also during feature extraction for each particular song.

	0.075 s	0.15 s	0.3 s	0.6 s
Mean	63.2%	62.5%	62.2%	54.8%
KL	65.8%	65.1%	63.1%	56.8%
KL_zero	66.4%	62.9%	56.8%	48.0%

Table 3. ***Components extracted at likely onset positions:*** **Average classification accuracy for the various methods with varying component length. Same abbreviations as in Table 1.**

As can be seen in Table 3, this approach did not improve the classification accuracies. Alternative reasons for the failure might be that the chosen onset detection algorithm might not be the best for our purpose, or that too few patches are extracted from each song, resulting in ill-defined activation histograms. The latter point in particular might be a reason for the low accuracies for patches of length 0.6 sec.

5.3. Self-defined Components

As described in Section 5.1, the independent components computed in this experiment are not clearly related to intuitively understandable musical properties. We were also interested in finding out if it is beneficial to intentionally define the components so that they might describe such properties. Therefore, we created a number of components ourselves (cf. Fig. 3). These components included averaging filters for three frequency regions, high-pass like elements in four frequency bands, components aimed to detect sustained sounds at the individual frequencies, and components that should be triggered by the presence of certain periodic events at various frequencies. The latter resemble the fluctuation patterns [11].

With the self-defined components, the independent activation of the components can not be assumed any more.

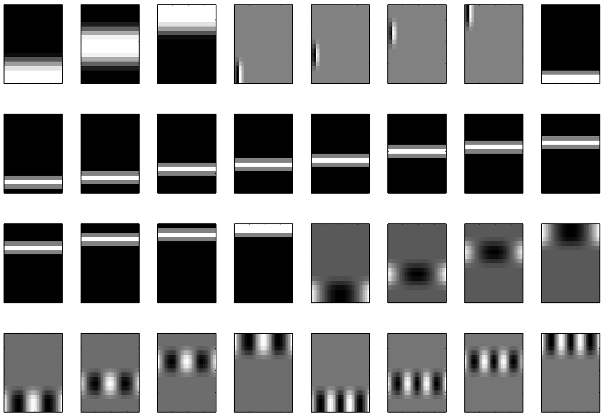

Figure 3. Example for self-defined components.

Thus, we expand the KL distance measure between the histograms to also take into account the covariance between the activations of different components during each song. Consequently, for this comparison method the feature data for a song consists of the mean value and the full covariance matrix. This additional comparison method is denoted as *KL_full* in Table 4.

	0.075 s	0.15 s	0.3 s	0.6 s
Mean	59.6%	58.1%	57.0%	55.7%
KL	64.0%	62.0%	61.0%	58.5%
KL_zero	64.4%	63.3%	61.0%	59.5%
KL_full	67.6%	68.7%	68.2%	67.0%

Table 4. ***Self-defined Components:*** **Average classification accuracy for the various methods with varying component length. Same abbreviations as in Table 1, with the additional method using the full covariance matrix (*KL_full*). Intentionally reducing the number of components did not lead to increased accuracies.**

From Table 4 it can be seen that *KL_full* produces the highest classification accuracies for the self-defined components. These accuracies are also slightly higher than those achieved with the other approaches we evaluated in this paper. However, we still were not able to find that the activations correspond to musical aspects as perceived by human listeners. Also, a linear combination of the distances produced by these algorithms with the distances produced by a MFCC-based algorithm (which produces an accuracy of 72% on this data) did not yield a relevant improvement.

6. Conclusion and Future Work

We evaluated a number of ways to apply what we consider an interesting approach to music similarity computation. The approach is based on Independent Component Analysis (ICA) and was originally developed as an image similarity measure. We chose the classification accuracy as a quality measure, which yields promising results. However, we also had hoped to extract musically meaningful information from the audio data with this approach; we have not succeeded in this so far.

Possible improvements of the algorithm include the use of other sparse coding techniques (e.g. Non-negative Matrix Factorization) instead of ICA, and the use of B-splines for histogram comparison. Using B-splines would not be feasible for self-defined components, as no interdependencies between the components can be modelled with this. Also it might be possible to model the temporal order of component activations, e.g. via Hidden Markov Models (HMMs).

7. Acknowledgements

This research is supported by the Austrian Fonds zur Förderung der Wissenschaftlichen Forschung (FWF) under project number L112-N04, and by the EU 6th FP project S2S2 ("Sound to Sense, Sense to Sound", IST-2004-03773). The Austrian Research Institute for Artificial Intelligence also acknowledges financial support by the Austrian Federal Ministries BMBWK and BMVIT.

References

[1] Tim Pohle, "Extraction of Audio Descriptors and their Evaluation in Music Classification Tasks," M.S. thesis, TU Kaiserslautern, ÖFAI, DFKI, 2005, available at http://kluedo.ub.uni-kl.de/volltexte/2005/1881/.

[2] Elias Pampalk, *Computational Models of Music Similarity and their Application in Music Information Retrieval*, Ph.D. thesis, Technische Universitat Wien, 2006.

[3] Beth Logan, "Mel frequency cepstral coefficients for music modeling," Read at the first International Symposium on Music Information Retrieval, 2000.

[4] Beth Logan and Ariel Salomon, "A music similarity function based on signal analysis," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'01)*, Tokyo, Japan, August 22-25 2001.

[5] Jean-Julien Aucouturier and Francois Pachet, "Improving timbre similarity: How high is the sky?," *Journal of Negative Results in Speech and Audio Sciences*, vol. 1, no. 1, 2004.

[6] Michael Mandel and Dan Ellis, "Song-Level Features and Support Vector Machines for Music Classification," in *Proc. International Symposium on Music Information Retrieval (ISMIR'05)*, London, UK, 2005.

[7] Arthur Flexer, Elias Pampalk, and Gerhard Widmer, "Hidden markov models for spectral similarity of songs," in *Proceedings of the 8th International Conference on Digital Audio Effects (DAFx'05)*, 2005.

[8] Michael Casey, "Mpeg-7 sound-recognition tools," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 737–747, June 2001.

[9] 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.

[10] S. Dixon, E. Pampalk, and G. Widmer, "Classification of dance music by periodicity patterns," in *Proceedings of the Fourth International Conference on Music Information Retrieval (ISMIR'03)*, Baltimore, MD, USA, October 26-30 2003, pp. 159–166, John Hopkins University.

[11] Elias Pampalk, "A matlab toolbox to compute music similarity from audio," in *Proceedings of the Fifth International Conference on Music Information Retrieval (ISMIR'04)*, Barcelona, Spain, October 10-14 2004.

[12] Elias Pampalk, Arthur Flexer, and Gerhard Widmer, "Improvements of audio-based music similarity and genre classificaton," in *Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR'05)*, London, UK, October 10-14 2005.

[13] Samer A. Abdallah and Mark D. Plumbley, "If the independent components of natural images are edges, what are the independent components of natural sounds?," 2001.

[14] Paris Smaragdis, *Redundancy reduction for computational audition. A unifying approach.*, Ph.D. thesis, Massachusetts Institute of Technology, Media Laboratory, 2001.

[15] Shankar Vembu and Stephan Baumann, "Separation of vocals from polyphonic audio recordings," in *Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR'05)*, London, UK, September 11-15 2005, pp. 337–344.

[16] Samer Abdallah and Mark Plumbley, "Polyphonic music transcription by non-negative sparse coding of power spectra," in *Proceedings of the Fifth International Conference on Music Information Retrieval (ISMIR'04)*, Barcelona, Spain, October 10-14 2004.

[17] Hervé Le Borgne, Guérin-Dugué Anne, and Anestis Antoniadis, "Representation of images for classification with independent features," *Pattern Recognition Letters*, vol. 25, pp. 141–154, jan 2004.

[18] Anthony J. Bell and Terrence J. Sejnowski, "The 'independent components' of natural scenes are edge filters," *Vision Research*, vol. 37, no. 23, pp. 3327–3338, 1997.

[19] Samer A. Abdallah, *Towards music perception by redundancy reduction and unsupervised learning in probabilistic models*, Ph.D. thesis, Department of Electronic Engineering, King's College London, 2002.

[20] Kazuyoshi Yoshii, Masataka Goto, and Hiroshi G. Okuno, "Automatic drum sound description for real-world music using template adaption and matching methods," in *Proceedings of the Fifth International Conference on Music Information Retrieval (ISMIR'04)*, Barcelona, Spain, October 10-14 2004.