DRUM TRANSCRIPTION USING SEMI-SUPERVISED NON-NEGATIVE MATRIX FACTORIZATION

First author
Affiliation1
author1@ismir.edu

Second author
Retain these fake authors in
submission to preserve the formatting

Third author
Affiliation3
author3@ismir.edu

ABSTRACT

In this paper, a drum transcription algorithm using semi-supervised non-negative matrix factorization has been presented. This method allows users to separate percussive activities from harmonic activities with pre-trained drum templates, and detect drum events from the extracted percussive activity matrix. A polyphonic subset from ENST drum data set has been used for training and testing of the algorithm. The system has been tested using different rank settings, and a cross-performer validation process has been performed to evaluate the reliability of the system. The results show that the system can achieve 61 to 77% recognition rate on multiple drums in polyphonic music. In the future, more efforts will be put on differentiating different playing styles and more drum parts, leading toward a complete drum transcription system.

1. INTRODUCTION

Automatic music transcription is one of the most intensively researched areas in Music Information Retrieval (MIR). The reliable extraction of a score (or a score-related representation) from the audio signal will be the core technology of a large number of applications in fields such as music education, systematic musicology, and music visualization. Furthermore, a reliable transcription would enable high-level representations of music signals with the potential of improving virtually any MIR task.

A complete transcription system comprises many related sub-tasks such as multi-pitch detection, onset detection, instrument recognition, and rhythm extraction [2]. While the main focus is mostly on pitched instruments, a considerable amount of publications deal with the transcription of percussive sounds in polyphonic music. The drum track in popular music conveys information about tempo, rhythm, style, and possibly the structure of a song. A drum transcription system alone enables applications in music production [19], music education, and interactive music performance [17].

This study explores the application of the popular transcription method of Non-Negative Matrix Factorization (NMF) for drum transcription in polyphonic music. The paper is structured as follows: Section 2 provides an overview of the research in this area. In Sect. Section 3 we present our approach; evaluation results are being presented and discussed in Sect. 4. Section 5 provides a summary, conclusion, and directions of future work.

2. RELATED WORK

WITHOUT HAVING THE PAPERS I CANNOT REVIEW THIS SECTION — SKIPPED FOR NOW!

The early attempts to transcribe percussive sounds main focused on classifying monophonic signals [11] [10]. With standard approaches such as feature extraction and classification, fairly high accuracy were reported in the previous studies. However, in the real use case, a drum transcription system is expected to work in polyphonic signal instead of monophonic. Therefore, solving this problem in the context of polyphonic music had become another criterion, and different methods could be found in previous research [5] [20] [4] [3] [16] [7]. According to [7], recent studies on the drum transcription in polyphonic music could be categorized into three types: segment and classify, separate and detect, match and adapt. For the first type of approaches, the common procedure starts by applying onset detection to the audio signal in order to segment the music event. Once the event has been detected, various features from time or spectral domain of the signal will be extracted, and a classifier will be trained to classify the event based on the extracted features. This type of approaches seem to perform well when the data and features are well chosen [16] [1]. However, to get good results, sufficient amount of data, careful preprocessing and training steps are required in this type of systems. Additionally, to handle the situation of simultaneous sounds, more classes need to be trained.

The second type of approaches is based on the assumption that music signal is a superposition of different sound sources. By decomposing the signal and into different source templates and corresponding activities, the music content could be transcribed by detecting onsets of these activities. Different methods such as Independent Subspace Analysis (ISA) [6], Prior Subspace Analysis (PSA) [5], and Non-negative Matrix Factorization (NMF) [1] [13] [12] are the examples in this category. This type of approaches is usually easier to interpret, since most of the decompositions

have been done on the spectrogram of the signal. Also, the separate nature allows simultaneous events to be handled easily in this case. However, to be able to transcribe different kinds of music, a large template might be required prior to the decomposition. Moreover, the number of rank during the decomposition process could be difficult to determine in some cases.

The third type of approaches uses pre-trained templates to detect drum events [21]. An iterative process has been taken to search for the closest matches to these templates and adapt them. The proposed system has been evaluated and performed well in MIREX 2005 drum detection competition. However, to coverage a wider range of sounds, multiple seed templates need to be prepared prior to the process.

In this paper, a method based on the NMF approach has been presented. Although NMF has been proven to be effective in music transcription ([15]), it is still difficult to determine the number of sources in the target signal for a correct rank parameter setting. Therefore, the goal of this paper is to develop an algorithm that provides a more general use case, which only requires users to input a few drum samples prior to the decomposition process. This method aims to serve as an alternative way to transcribe drum tracks in polyphonic music, leading toward an end user application for general drum transcription tasks.

3. METHOD

3.1 Algorithm Description

The basic concept of NMF is to approximate a non-negative matrix V with two non-negative matrices W and H as shown in Eq. (1).

$$V \approx WH \tag{1}$$

That is, given a $m \times n$ matrix V, the NMF will decompose the matrix into the product of a $m \times r$ matrix W and a $r \times n$ matrix H, with r being the rank of the NMF decomposition. W is the dictionary matrix, and H is the activation matrix. In most audio applications, V is the spectrogram to be decomposed, W contains the magnitude spectra of the salient components, and H indicates the activation of these components with respect to time. The matrices W and H are obtained through an iterative process that minimizes a distance measure between the target spectrogram V and its approximation. [14].

The application of this method to music transcription, however, offers some challenges. First, the number of sound sources and notes within a music recording is usually unknown. It is therefore difficult to estimate a correct rank number and obtain a clear representation of the decomposed components. Second, it is hard to identify the corresponding instrument of every component in the dictionary matrix W, especially when the rank is very high or very low. Third, when multiple similar entries exist in the dictionary matrix, the corresponding activation matrix could be activated at these entries simultaneously and might cause some confusions. Different methods have been proposed

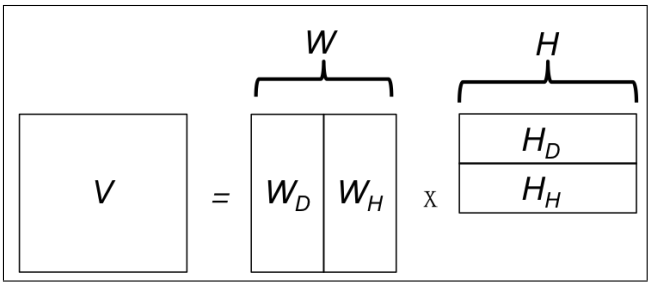

Figure 1. Illustration of the factorization process. The pre-trained drum basis matrix W_D will not be updated over the iterations.

in the previous studies to address these issues. Virtanen trained an SVM to separate drum components from the harmonic components; the rank number was derived empirically during the factorization process ([9]). The identified drum components and their corresponding activities could later be used to reconstruct the drum signal, resulting in a system for drum source separation. Virtanen's approach requires a significant amount of training data for the classifier and, more importantly, the results can be expected to be very susceptible to choice of rank. Yoo proposed a co-factorization algorithm [18] to simultaneously factorize a drum track a target signal containing this track and use the basis matrix from the drum track to identify the drum components in the target signal. This method ensures that the drum components in both basis matrices remain the same over the iterations, and thus proper isolation of the harmonic components from the drum components. The need to obtain a drum track prior to the factorization process, however, impacts the usefulness this approach for many application scenarios.

Nevertheless, Yoo's work on co-factorization inspired our work to present a similar system without the requirement the prior drum track. Figure 1 visualizes the basic concept: the matrices W and H are split into the matrices W_D and W_H, and H_D and H_H, respectively. The matrix W_D is initialized with the drum templates to be detected and will not be updated. Matrices W_H, H_H, and H_D can be initialized with random numbers. The cost function as shown in Eq. (2) is minimized by applying gradient decent and multiplicative update rules, the matrices W_H, H_H, and H_D will be updated according to Eqs. (4)–(5).

$$C = \frac{1}{2}||V - W_D H_D - W_H H_H||^2 \tag{2}$$

$$H_D \leftarrow H_D \frac{W_D^T V}{W_D^T W_D H_D + W_D W_H H_H} \tag{3}$$

$$W_H \leftarrow W_H \frac{V H_H^T}{W_H H_H H_H^T + W_D H_D H_H^T} \tag{4}$$

$$H_H \leftarrow H_H \frac{W_H^T V}{W_H^T W_H H_H + W_H W_D H_D} \tag{5}$$

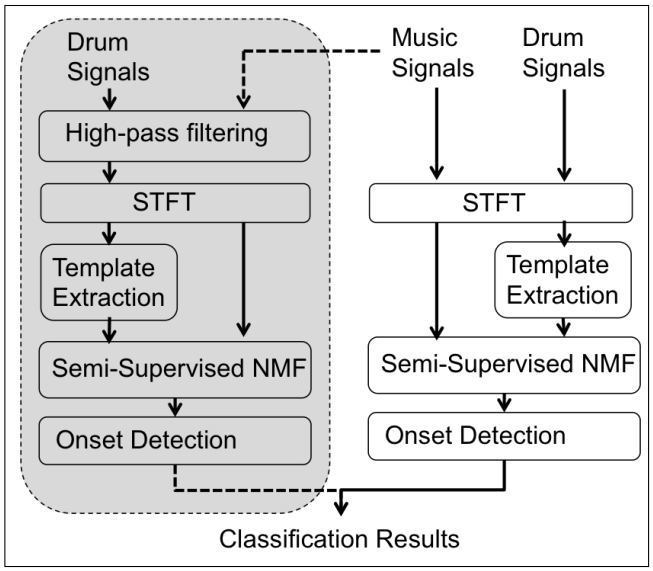

Figure 2. Flowchart of the drum transcription system

To summarize, the method described above consists of the following steps:

1. Construct a $m \times r_D$ dictionary matrix W_D, r_D is the number of drum components to be detected.
2. Given a pre-defined rank r_H, initialize a $m \times r_H$ matrix W_H, a $r_D \times n$ matrix H_D and a $r_H \times n$ matrix H_H.
3. Normalize W_D and W_H.
4. Update H_D, W_H, and H_H using Eqs. (4)–(5).
5. Calculate the cost of current iteration using Eq. (2).
6. Repeat step 3 to step 5 until convergence.

The time positions of the drum events can then be extracted with peak-picking the matrix H_D.

3.2 Implementation

Figure 2 shows the flow chart of the implemented system. The basic setup is shown on the right hand side. The STFT of the signals will be calculated using a window size and a hop size of 2048 and 512, respectively. A pre-trained dictionary matrix will be constructed from the training set, consisting of isolated drum sounds (see Sect. 3.2.1). Next, the semi-supervised NMF will be performed with rank r as described above. Finally, the activation Matrix H_D is evaluated to determine the onset positions and their corresponding classes (see Sect. 3.2.1).

The gray area in Figure 2 is an optional addition to the system. Preliminary tests showed improved Hi-Hat (HH) detection accuracy when applying a high-pass filter to both the template and the music signal. We used a 2nd order Butterworth high-pass filter with a heuristically determined cutoff frequency of 8000 Hz.

The following subsections provide more details on the template extraction and activity detection.

3.2.1 Template Extraction

As mentioned above, the dictionary matrix W_D is created by extracting a template spectrum from isolated training drum samples. The template magnitude spectrum is median spectrum of all individual events of one drum class in the training set. The length of each event is approximately 80 ms. The templates are extracted for the three classes Hi-Hat (HH), Bass Drum (BD) and Snare Drum (SD).

3.2.2 Activity Detection

High activity values in the activation matrix H_D indicate the presence of a drum event. More specifically, the activity difference of each row of the dictionary matrix could be considered as the onset novelty function of each individual drum. We use a median filter as a standard approach to create an adaptive threshold for pick peaking result.

4. EVALUATION

4.1 Dataset Description

The experiments have been conducted on the *minus one* subset from the ENST public drum data set [8]. This data set consists of recordings from three different drummers performing on their own drum kits. The set for each drummer contains individual hits, short phrases of drum beats, drum solos, and short excerpts played with the accompaniments. The minus one subset has 64 tracks of polyphonic music. Each track in this subset has a length of approximately 70 s with varying style. More specifically, the subset features many drum playing techniques such as ghost notes, flam, and drag; these techniques are considered difficult to identify with existing drum transcription systems. Since we are only interested in the three classes HH, BD, and SD, tracks missing one of these instruments or featuring specific playing techniques have been discarded, leaving a subset of 53 out of 64 tracks.

The accompaniments are mixed with the drum tracks in the data set without any modification (e.g., no level adjustment). The distribution of onset counts per class per drummer is shown in Table 1.

The drums template (see Sect. 3.2.1) have been generated from the tracks with single hits. Each track contains 5 to 6 single hits on different drums. The onset position of these single hits were determined using the annotated ground truth.

	Drum. 1	**Drum. 2**	**Drum. 3**	**Total**
HH	1942	2145	1813	5900
BD	2140	1488	1378	5006
SD	2165	2079	1994	6238
Total	6247	5712	5185	17144

Table 1. Onset counts in selected data set

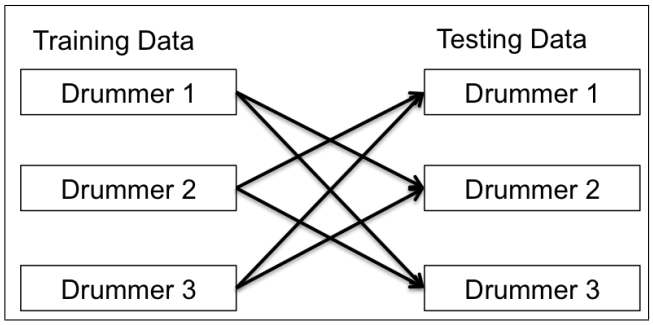

Figure 3. Cross-performer validation process.

4.2 Evaluation Procedure

We evaluate two different combinations of training and test data.

First, we use training samples from all three drummers to train the drum basis matrix, and test the system on all 53 tracks. In this test run, we also evaluate the results in case of filtered training and testing signals.

Second, we investigate cross-performer validation as shown in Figure 3. The training samples are selected only from one drummer, and the test samples will be only the other drummers' recordings. This scenario should be similar to a real-world use case for which the trained drum sounds are not necessarily similar to the drum sounds in the target signals.

The evaluation metrics follow by the standard calculation of the precision (P), recall (R), and F-measure (F). An onset is considered to be a match with the ground truth if the time deviation between the annotated and detected onset is less or equal to 50 ms.

4.3 Evaluation Results

4.3.1 Pre-Evaluation Test

In an initial test to determine the rank r_H of the algorithm, $r_H = 5, 10, 20, 40, 80, 160, 320$ have been tested. The resulting individual F-measures are shown in Figure 4. A general trend of decreasing performance with increasing r_H can be observed, especially for lower frequency sounds such as SD and BD. Based on this observation, a rank number $r_H = 10$ is chosen in current setup.

4.3.2 Evaluation I

Table 2 shows the results (precision, recall, and F-measure) when using the complete training and test set. The results for the standard setup (the white section of Figure 2) are given in the column named *Standard*. The middle column (*HPF+LPF+BPF*) presents the results for signals pre-processed with filters. More specifically, we use three filters corresponding to the three instruments we intend to detect; the filter frequencies are derived from the median magnitude spectra:

- HH: $f_{hp} = 8000\,\text{Hz}$
- SD: $f_{bp} = 200 - -500\,\text{Hz}$

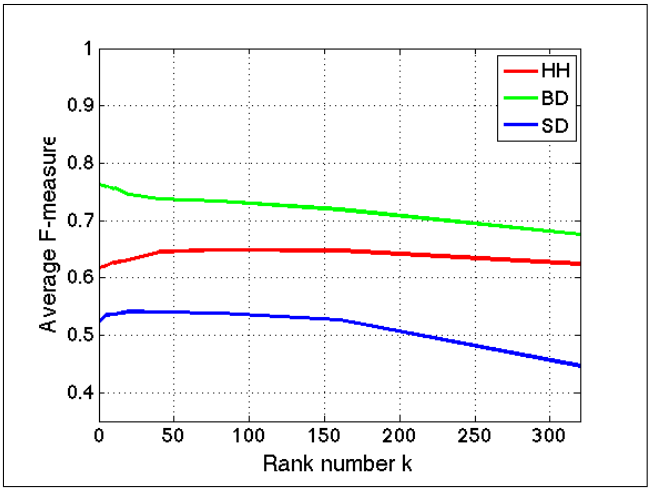

Figure 4. Average F-measure versus NMF rank

- BD: $f_{lp} = 150\,\text{Hz}$

The last column uses a similar setup, but only one high-pass filter for the HH class.

		Standard	HPF+LPF+BPF	HPF only
HH	P	0.605	0.671	0.675
	R	0.713	0.737	0.736
	F	0.654	0.702	**0.704**
BD	P	0.723	0.662	0.740
	R	0.837	0.855	0.839
	F	0.776	0.746	**0.786**
SD	P	0.678	0.591	0.676
	R	0.566	0.583	0.569
	F	0.617	0.587	**0.618**

Table 2. Transcription results using all training templates

It can be observed that using the filters does not necessarily improve the results. The combination of three filters in particular seems to improve results slightly for HH, but the drop in precision of the other classes results in a lower F-measure. In the third, high-pass-only, case we can identify a general trend for improved results in all classes, although the increase for SD and especially BD is rather negligible. These small increase might due the random initialization of the harmonic dictionary, but the deviations are not significant. Comparing with the reported F-measure of 77.7%, 65.0% and 64.8% for HH, BD and SD in [7], our results show better performance on BD, but slightly worse performance in SD and HH. However, in our method, the training process require less amount of training data to achieve the results at same level, which could potential provide a better use case for a more generic drum transcription system.

The results in the middle column of Table 2 indicate that by filtering the templates and signals with constant cutoff frequencies, the performance of BD and SD drop slightly. Some possible reasons might contribute to this result. First, although the filtering process could suppress

contents in the unwanted frequency bins, it also could remove many information that might help to differentiate instruments such as BD and Bass Guitar. Second, in current parameter settings, the band-pass and low-pass filtered signals might only have few bins to represent the spectrum, which might increase the difficulty for the semi-NMF to adapt. Third, when the signals are filtered, the current rank setting r_H might not be optimal for the task.

4.3.3 Evaluation II

LETS THINK OF WAYS TO CUMMARIZE THESE THREE TABLES IN ONE TABLE AND A GRAPH, MAYBE DISCARDED PRECISION AND RECALL. MAYBE ALSO OVERALL AVERAGE MEASURES (TRAINING WITH 1, TESTING WITH ALL)?

In order to investigate the dependency of our approach with respect to the similarity of training and test drum sounds, we conduct a cross-performer evaluation as shown in Figure 3. No filters have been applied during the process. There results are listed in Table 3, Table 4, and Table 5.

Training		Drummer 1		
Testing		Drummer 1	Drummer 2	Drummer 3
HH	P	0.662	0.621	0.564
	R	0.670	0.749	0.692
	F	0.666	**0.679**	0.622
BD	P	0.620	0.781	0.850
	R	0.393	0.914	0.900
	F	0.481	0.842	**0.874**
SD	P	0.639	0.800	0.589
	R	0.546	0.608	0.487
	F	0.588	**0.691**	0.533

Table 3. Transcription results using drummer 1 training templates.

Training		Drummer 2		
Testing		Drummer 1	Drummer 2	Drummer 3
HH	P	0.661	0.600	0.541
	R	0.667	0.787	0.699
	F	0.664	**0.681**	0.610
BD	P	0.466	0.846	0.699
	R	0.603	0.986	0.854
	F	0.525	**0.910**	0.769
SD	P	0.625	0.849	0.529
	R	0.550	0.624	0.486
	F	0.585	**0.719**	0.506

Table 4. Transcription results using drummer 2 training templates.

The results show a simple trend: the test set featuring drummer 2 give nearly always the best results, regardless of the training set. The Bass Drum of drummer 2 seems to be particularly easy to detect. An even more surprising result is that there does not seem to be an obvious increase of the F-measure when the training drummer and test set drummer are identical. These results indicate that this algorithm is relatively robust against differences between the drum template and the sound of the drum to be detected. This would allow to construct a template from different sound sources independent of the recording to be analyzed allowing far more general applications. However, the performance of this setup still needs to be confirmed with a cross-data set validation.

Training		Drummer 3		
Testing		Drummer 1	Drummer 2	Drummer 3
HH	P	0.614	0.599	0.523
	R	0.660	0.717	0.675
	F	0.636	**0.652**	0.589
BD	P	0.506	0.862	0.800
	R	0.636	0.965	0.932
	F	0.564	**0.910**	0.861
SD	P	0.477	0.615	0.564
	R	0.580	0.598	0.534
	F	0.523	**0.607**	0.548

Table 5. Transcription results using drummer 3 training templates.

5. CONCLUSION

We have presented a drum transcription system for polyphonic music using semi-supervised NMF. This method uses a pre-trained dictionary matrix to decompose the target signal and extract the activation matrix. This evaluation results show that this method is able to achieve 61-77% accuracy in polyphonic music. It has the following advantages: First, simultaneous sounds can be detected separately. Traditional systems for drum detection often have difficulties with simultaneous events, especially when the number of instruments increases. Second, adjustment of the parameter r allows the algorithm to adapt to different different types of polyphonic music. Third, cross-perfomer evaluation results indicate a robustness against template mismatches, possibly allowing the application in situations with minimum prior knowledge. Fourth, the approach requires only trivial extensions to be able to be used as a drum separation system. Last but not least, the approach only requires a few drum samples to train the dictionary matrix, and the evaluation results indicate that the performance is comparable with the existing methods. This means the current approach has less constraints on the user input data, and could potentially be realized as a generic drum transcription system .

There are some possible directions for the future works: first of all, a comparison between this approach with Probabilistic Latent Component Analysis (PLCA) and looking for means to adapt the template during the decomposition might be a way to further generalize the current method. Also, other distance metrics such as KL-divergence or beta divergence could be implemented for better approximations. Finally, finding a method to automatically determine

the rank r_H could optimize this approach against different polyphonic music. To achieve a complete drum transcription system in polyphonic music, however, more factors such as playing techniques and different drum setups still need to be addressed in the future.

6. REFERENCES

[1] DS Alves, Jouni Paulus, and J Fonseca. Drum transcription from multichannel recordings with non-negative matrix factorization. *17th European Signal Processing Conference (EUSIPCO 2009)*, (Eusipco):894–898, 2009.

[2] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, July 2013.

[3] Christian Dittmar. Drum detection from polyphonic audio via detailed analysis of the time frequency domain. *1st Music Information Retrieval Evaluation eXchange*, 2005.

[4] Christian Dittmar and Christian Uhle. Further steps towards drum transcription of polyphonic music. *Audio Engineering Society Convention 116*, 2004.

[5] D FitzGerald, Bob Lawlor, and Eugene Coyle. Drum transcription in the presence of pitched instruments using prior subspace analysis. In *The Irish Signals & Systems Conference*, number 3, 2003.

[6] D FitzGerald, R Lawlor, and Eugene Coyle. Sub-band Independent Subspace Analysis For Drum transcription. In *Digital Audio Effects Conference (DAFX02)*, number 5, pages 65–69, 2002.

[7] O. Gillet and G. Richard. Transcription and separation of drum signals from polyphonic music. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(3):529–540, March 2008.

[8] Olivier Gillet and G Richard. Enst-drums: an extensive audio-visual database for drum signals processing. In *Proceedings of the 7th International Symposium on Music Information Retrieval*, 2006.

[9] M Helén and Tuomas Virtanen. Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine. In *Proceedings of the 13th European Signal Processing Conference*, 2005.

[10] P Herrera, A Dehamel, and F Gouyon. Automatic labeling of unpitched percussive sounds. *Audio Engineering Society 114th Convention*, 2003.

[11] Perfecto Herrera, Alexandre Yeterian, and Fabien Gouyon. Automatic classification of drum sounds: a comparison of feature selection methods and classification techniques. *Music and Artificial Intelligence*, pages 69–80, 2002.

[12] Arnaud Moreau and Arthur Flexer. Drum transcription in polyphonic music using non-negative matrix factorization. In *International Symposium on Music Information Retrieval*, 2007.

[13] Jouni Paulus and Tuomas Virtanen. Drum transcription with non-negative spectrogram factorisation. In *Proceedings of the 13th European Signal Processing Conference*, 2005.

[14] D Seung and L Lee. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, number 13, pages 556–562, 2001.

[15] P Smaragdis and JC Brown. Non-negative matrix factorization for polyphonic music transcription. *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, pages 177–180, 2003.

[16] K Tanghe, S Degroeve, and B De Baets. An algorithm for detecting and labeling drum events in polyphonic music. In *Proceedings of 1st Music Information Retrieval Evaluation eXchange (MIREX)*, 2005.

[17] Gil Weinberg, A Raman, and T Mallikarjuna. Interactive jamming with Shimon: a social robotic musician. *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 233–234, 2009.

[18] Jiho Yoo, Minje Kim, Kyeongok Kang, and Seungjin Choi. Nonnegative matrix partial co-factorization for drum source separation. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 1942–1945, 2010.

[19] Kazuyoshi Yoshii, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and HG Okuno. Drumix: An audio player with real-time drum-part rearrangement functions for active music listening. *IPSJ Journal*, 48(3), 2007.

[20] Kazuyoshi Yoshii, Masataka Goto, and HG Okuno. Drum sound identification for polyphonic music using template adaptation and matching methods. In *ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing*, 2004.

[21] Kazuyoshi Yoshii, Masataka Goto, and Hiroshi G. Okuno. Drum Sound Recognition for Polyphonic Audio Signals by Adaptation and Matching of Spectrogram Templates With Harmonic Structure Suppression. In *IEEE Transactions on Audio, Speech and Language Processing*, volume 15, pages 333–345, January 2007.