REAL-TIME TRANSCRIPTION AND SEPARATION OF DRUM RECORDINGS BASED ON NMF DECOMPOSITION

*Christian Dittmar**
Fraunhofer IDMT,
Ilmenau, Germany
dmr@idmt.fraunhofer.de

Daniel Gärtner
Fraunhofer IDMT,
Ilmenau, Germany
gtr@idmt.fraunhofer.de

ABSTRACT

This paper proposes a real-time capable method for transcribing and separating occurrences of single drum instruments in polyphonic drum recordings. Both the detection and the decomposition are based on Non-Negative Matrix Factorization and can be implemented with very small systemic delay. We propose a simple modification to the update rules that allows to capture time-dynamic spectral characteristics of the involved drum sounds. The method can be applied in music production and music education software. Performance results with respect to drum transcription are presented and discussed. The evaluation data-set consisting of annotated drum recordings is published for use in further studies in the field.

Index Terms - drum transcription, source separation, non-negative matrix factorization, spectral processing, audio plug-in, music production, music education

1. INTRODUCTION

The rapid development of music technology in the past decades has inevitably changed the way people interact with music today. As one result, music production has shifted almost entirely to the digital domain. This evolution made music recording affordable for amateurs and semi-professionals. Furthermore, it enabled the rise of completely novel approaches to music practice and education. With these developments in mind, our work focuses on the real-time processing of drum set recordings. We aim at transcribing and isolating the single drum instruments that are played in a monaural, polyphonic drum set recording. Thus, our topic is at the intersection of automatic music transcription and source separation, two major fields in *Music Information Retrieval (MIR)* research [1, 2]. Strictly speaking, we are performing drum detection rather than transcription, since our approach is agnostic to the underlying metric structure (relations of beats and bars). However, we will use the term drum transcription for the sake of simplicity throughout the paper. We also use the term decomposition as synonym for source separation.

Our paper is structured as follows. First, the goals of our work are outlined in Sec. 2. After a review of the related work in Sec. 3, we explain the proposed transcription and separation algorithm in detail in Sec. 4. Finally, Sec. 5 describes the evaluation conducted and Sec. 6 summarizes this work.

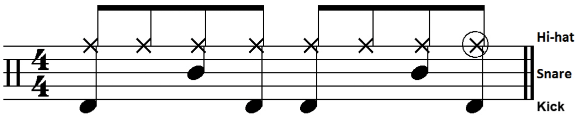

Figure 1: *A simple, one-bar drum rhythm in music notation. Taken from [3].*

2. GOALS

In professional music production, drum kits are usually recorded using several microphones that allow for separate processing of the different drum instrument signals via mixing desks. However, proper microphone setup is not trivial and even professional audio engineers often have to cope with heavy cross-talk between recording devices. In addition, amateur music producers might only have a single microphone available due to limited budget. Thus, our goal is to detect and separate occurrences of single drums within monaural polyphonic drum set recordings in real-time.

Our first application scenario is music production software, where post-processing of individual drum instruments in the mix plays an important role. In digital music production, so-called drum trigger plug-ins, such as Drumagog [1] or Steven Slate Trigger [2] are quite common. When applied to multi-channel drum set recordings, onsets can be detected in each drum channel and can be used to trigger additional digital audio samples. In a sense, these tools already perform monophonic drum transcription. One drawback of these plug-ins is the need for manual setting of trigger thresholds. Furthermore, they offer only conventional means (e.g., equalization, noise-gates) for attenuating cross-talk between drum channels. Of course, it would be desirable to better isolate the single drum-sounds automatically. As will be explained in Sec. 4, our approach requires to train the system with isolated drum sounds of the expected drum instruments. Having in mind that all drum instruments are played in succession during sound-checks, it is quire realistic to fulfill that requirement in practice.

The second application scenario are educational music games, such as Songs2See[3], BandFuse[4] and RockSmith[5]. Only a small number of music video games and music education software also offer the possibility to practice drums. In all cases, this functionality is enabled by using MIDI-fied drum sets. However, none of the existing applications allows users to practice on real-world acous-

[*] All correspondance should be adressed to this author.

[1] http://www.drumagog.com/
[2] http://www.stevenslatedrums.com/
[3] http://www.songs2see.com/
[4] http://bandfuse.com/
[5] http://rocksmith.ubi.com/

tic drum sets. We want to enable budding drummers to play along to a given rhythm pattern or song, while their performances, in terms of striking the correct drums to the correct points in time, are assessed in real-time. As a pre-requisite, it is necessary to recognize the different drum instruments in a monaural audio signal. Having beginners in mind, the system is constrained to detect onsets of three drum instruments as explained in Sec. 2.1. In educational music video games available on the market, it is pretty common to have a tuning stage before playing a song. In the same manner, we can require the use to play all drum instruments in succession for training the system.

2.1. The drum kit

A conventional drum kit usually consists of the drum instruments shown in Figure 2. They comprise the kick (1), snare (2), toms (3,4), hi-hat (5) and cymbals (6,7). The drums can be classified into membranophones (kick, snare, toms) and ideophones (hi-hat, cymbals). The sound is produced by striking them with drum sticks usually made of wood. In this work we are focusing on kick, snare and hi-hat. The kick is played via a foot pedal, generating a low, bass-heavy sound. The snare has snare wires stretched across the lower head. When striking the upper head with a drum stick, the lower head vibration excites the snares, generating a bright, snappy sound. The hi-hat can be opened and closed with another foot pedal. In closed mode, it produces a clicking, instantly decaying sound. In opened mode, it can sound similar to a cymbal with many turbulent high frequency modes. Real-world acoustic drums generate sound spectra that vary slightly with each successive stroke. Sample-based drum kits usually feature a limited number of pre-recorded drum sounds, while synthetic drum kits often provide just one particular sound (given the synthesis parameters are fixed).

Generally speaking, kick, snare and hi-hat can be ordered ascending by their spectral centroid. However, when polyphonic drum rhythms are played on a drum set, it is pretty common that different drums are struck simultaneously. In many common drum rhythms, the hi-hat plays all quarter or eighth notes and therefore coincides with kick and snare quite often. An example is shown in Figure 1. If such short rhythms of one to four bars are constantly repeated they are also called drum loop. In these cases, discerning the instruments by their spectral centroid is no longer possible, since only the mixed sound can be measured. In the worst case, a kick occurring simultaneously with a hi-hat could be mistaken for a snare drum. Besides the recognition of ghost-notes and other special playing techniques, the ambiguity in classifying polyphonic drum sounds poses the major challenge in automatic drum transcription.

3. STATE-OF-THE-ART

In this section, the most important directions of research in automatic drum transcription are presented. As described in [4], the existing approaches can be discerned into three different categories.

3.1. Source separation methods

The first category is also known as *separate and detect* because the signal is first decomposed into individual streams via source separation, before onset candidates are detected in each individual stream. The pre-requisite is typically a time-frequency transform (e.g., the *Short-term Fourier Transform (STFT)*). The generic signal model decomposes the resulting magnitude spectrogram X into a linear superposition of individual component spectrograms. The components are usually represented by fixed spectral basis functions B and corresponding time-varying amplitude (or gain) envelopes G. An intuitive interpretation is that the B describe how the constituent components sound, whereas the G describe when and how intense they sound. The approaches described in the literature mostly differ in the decomposition method as well as the constraints and initialization imposed on B and G.

Independent component analysis (ICA) computes a factorization

$$X = B \cdot G \tag{1}$$

such that the separated source spectra are maximally independent and non-Gaussian. *Independent subspace analysis (ISA)*, first described in [5], applies *Principal Component Analysis (PCA)* and ICA in succession for decomposing X. In order to classify the arbitrarily permuted and scaled components afterwards, feature extraction and classifiers such as *k-Nearest-Neighbor (kNN)* or *Support Vector Machines (SVM)* can be used [6]. An extension to ICA called *Non-Negative ICA (NICA)* has the constraint that the matrix B must be non-negative [7]. In [8], it is shown how to use NICA for transcription of kick, snare and hi-hat from polyphonic music.

Prior subspace analysis (PSA) was first proposed in [9] and utilizes a set of template spectrum basis functions in a matrix B_p. These consist of the averaged spectra drawn from a large collection of isolated drum sounds. A first approximation of G can be computed by

$$\hat{G} = B_p^+ \cdot X \tag{2}$$

where B_p^+ denotes the pseudo-inverse of B_p. The rows of matrix $\hat{G}$ contain the temporal activations of the template spectra in the spectrogram, but are not independent. In order to make them independent, ICA is applied afterwards. This results in an unmixing matrix W transforming $\hat{G}$ into independent amplitude gain functions according to

$$G = W \cdot \hat{G} \tag{3}$$

Subsequently, an improved estimate of the source spectra can be computed by

$$B = X \cdot G^+ \tag{4}$$

which now contains the source spectra adapted to the actual signal. Using this method, [10] reports an F-measure of 0.75 for the detection of kick and snare in polyphonic music.

An early work applying *Non-negative Matrix Factorization (NMF)* [11] for the separation of drums from polyphonic music is presented in [12]. It uses NMF minimizing the *Kullback-Leibler Divergence (KL)*, with random initialization of B and G. From the resulting components, spectral and temporal features are computed and classified with an SVM trained on the classes drums vs. harmonic. The reported results show that the NMF and SVM approach performed better than ISA and SVM. Another variant of NMF for drum transcription is described in [13]. The NMF is first applied to individual drum samples for kick, snare and hi-hat in order to derive B_p, which are later fixed during the NMF iterations. The method shows good performance on drum loops, yielding an average F-measure of 0.96 for kick, snare and hi-hat detection. In [14, 15], it is shown how source separation of instruments with time-varying spectral characteristics (such as drums) may benefit from an NMF extension called *Non-Negative Matrix Factor Deconvolution (NMFD)*. Recently, NMF-based methods have also

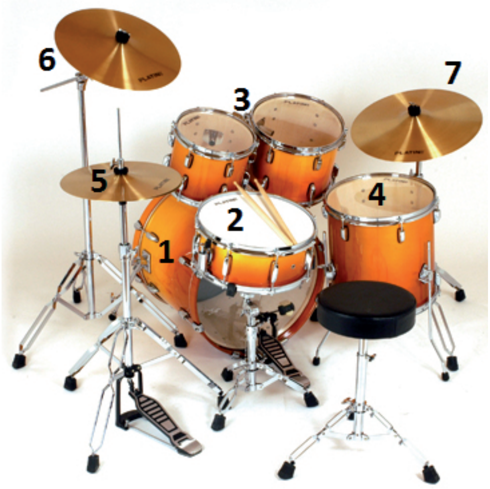

Figure 2: *A conventional drum kit with annotated drum instruments, taken from [3].*

been applied to real-time drum detection [16], where each drum onset is identified with Probabilistic Spectral Clustering based on the *Itakura-Saito Divergence (IS)*.

3.2. Template matching

The second category of drum transcription techniques follow a so-called *match and adapt* approach. It relies on temporal or spectral templates for the events that should be detected. In a first approximation, the occurrences of events that are similar to the template are detected. Afterwards, the templates are iteratively adapted to the given signal. The work presented in [17] uses *seed templates* for kick and snare, that are constructed from a collection of isolated drum sound spectrograms. First, onset detection determines possible candidates for drum sounds. At each onset candidate, a spectrogram snippet with the same size as the template is stored and compared with the templates. The reciprocal of the distance between the observed spectrogram and the template yields the reliability for that drum's occurrence. In the adapt stage, the seed templates are updated by taking the median power over all previously selected frames. This suppresses highly variable spectral peaks from pitched, harmonic instruments. The process of template adaption is applied iteratively, so that the output of this median filtering is used as the next seed template. The final stage determines whether the drum sound actually occurs at the onset candidate. Application of the template matching in conjunction with harmonic structure suppression, yielded an F-measure of 0.82 for kick and 0.58 for snare. A combination of template matching and sound separation is described in [18], where the candidates for template extraction are first detected using NMF. Instead of median filtering, a modified minimum filtering is applied. Another example of template matching is given in [19], where characteristic band pass filter parameters are learned. The training process is realized as an optimization of the characteristic filters with the *Differential Evolution (DE)* algorithm and fitness evaluation measures for determining each filter's ability to correctly detect the onset of the respective drum. The output of each filter represents the activations of the single drums and can be transcribed by means of envelope extraction and peak picking.

3.3. Supervised classification

The last category of transcription algorithms is referred to as *segment and classify*. It first employs temporal segmentation of the audio track into onset events. Usually, a fixed number of frames following each detected onsets is kept or a temporal grid of fixed periodicity is aligned to the audio track. Subsequently, each temporal event is identified by a classifier. Often, well-known machine learning methods such as SVM or GMM are used in conjunction with features extracted from each segment. The method in [20] uses a set of features comprising averaged MFCCs, various spectral shape parameters and the log-energy in six frequency bands corresponding to the spectral centroids of different drum instruments. The features are classified by a set of eight binary SVMs that have been trained on the classes kick, snare, hi-hat, clap, cymbal, rim shot, toms and percussion. Evaluated on a data-set of drum loops, the best configuration yielded a recognition rate of 83.9%. The method proposed in [21] uses a similar approach, but is applied for drum transcription in polyphonic music. The algorithm achieved an average F-measure of 0.61 for kick, snare and hi-hat. Finally, *Hidden Markov models (HMM)* are a machine learning method that can be used to model drum sequences. Although they are often counted as part of the *segment and classify* approach, they stand out as they are able to perform the segmentation and detection jointly. HMMs model temporal sequences by computing the probability that a given sequence of observed states were generated by hidden random variables, i.e., the activations of the drum classes. In [4], HMMs are used to model MFCCs and their temporal derivatives. The method achieves an F-measure of 0.81 for the recognition of kick, snare and hi-hat in drum loops and 0.74 in polyphonic music.

4. PROPOSED METHOD

In the preceeding section, we showed that good results have already been achieved in drum loop transcription. However, only a fraction of the methods is capable of real-time processing and only very few are suited for sound separation as well. As laid out in Section 2, our approach should cover both aspects. An overview about our proposed method is given in Figure 3. As in other works, we

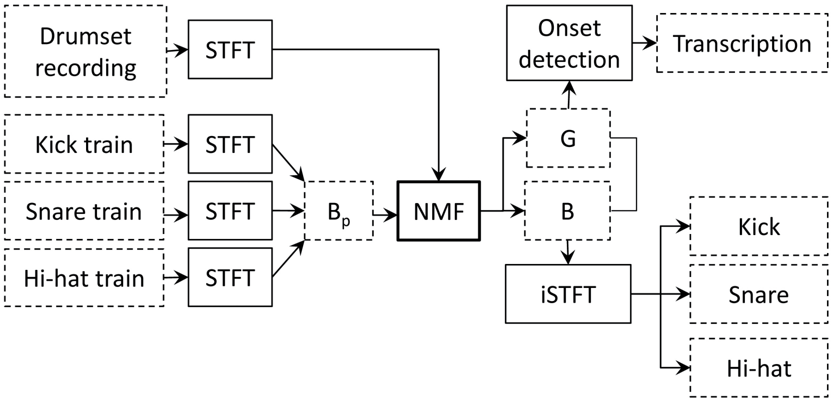

Figure 3: *Overview of the proposed method. Prior basis vectors B_p are derived from isolated drum sound spectrograms. Drum set recordings are split into spectral frames individually subjected to NMF. The resulting B and G are used for onset detection as well as inverse STFT in order to obtain isolated drum instrument recordings.*

also transform the drum recording to the time-frequency domain via STFT. As indicated in Sec. 2, we follow the approaches described in [13, 16]. We assume that an initial training phase can be conducted, where the individual drum sounds expected in the drum recordings are available in isolation. During training, we compute one prior basis vector B_p per drum instrument by just averaging along the time axis of each training spectrogram. The choice of just a single basis vector per drum is motivated by the findings in [22] as well as our general goal to spare computation time for real-time applicability. Of course, it is possible to use more than one component per drum and still reach real-time capability. In order to keep the number of samples required for processing as small as possible, the NMF decomposition is applied to each spectral frame of the drum recording individually, thus generating a succession of activations for kick, snare and hi-hat in G. In the following, three variants of the NMF decomposition are detailed.

4.1. NMF decomposition with adaptive bases

For decomposition, we use the KL Divergence resulting in the well known update rules [11] for both the spectral bases (5) as well as the amplitude envelopes (6):

$$B \leftarrow B \cdot \frac{\frac{X}{BG} G^T}{1 B^T} \tag{5}$$

$$G \leftarrow G \cdot \frac{B^T \frac{X}{BG}}{B^T 1} \tag{6}$$

It should again be noted, that X represents an $N \times 1$ matrix corresponding to one individual spectral frame with N linearly spaced frequency bins. The matrix 1 consists of all ones in the appropriate dimensions. The spectral basis matrix B is initialized with B_p as proposed in other works [13, 23, 16].

4.2. NMF decomposition with fixed bases

As proposed by other authors [24], we optionally omit the update of B in Eq. 5 and just replace B with the fixed prior basis B_p. This way, it can be ensured that only the expected spectra will lead to activations in G. It can be assumed, that NMF with only one fixed basis vector per instrument will not be able to model time-dynamic spectral characteristics of drum sounds, which is in line with the findings of [16], where separate NMF templates for head and tail of a drum sound are used. Intuitively, this method is also likely to produce spurious activations in case the incoming signal consists of other components than the previously trained drum sounds. The NMF updates rules will try to model the currently observed spectra as good as possible given the fixed prior basis vectors, thus yielding activations of all drum components in the form of cross-talk. Consequences for the resulting approximation of X will be explained in 5.3.

4.3. NMF decomposition with semi-adaptive bases

In our novel approach, we introduce a modification imposing semi-adaptive behavior on B during the NMF iterations. In contrast to the procedure described in Sec. 4.1, we do not just initialize B with B_p and let them iterate freely afterwards. Instead, we allow the spectral content in B to deviate more from the initial value, the closer we are to the NMF iteration limit. This behavior is simply achieved by blending between the initial B_p and B from the current iteration as given in Equation 7. The blending parameter α depends on the ratio of current iteration count k to iteration limit K taken to the power of β as show in Equation 8.

$$B = \alpha \cdot B_p + (1 - \alpha) \cdot B \tag{7}$$

$$\alpha = (1 - \frac{k}{K})^{\beta} \tag{8}$$

Thus, the NMF components are first pushed towards the expected drum sounds. The adaption to subtle variations in the incoming spectra are allowed later. It should be noted, that the proposed procedure is not equal to *Online Non-Negative Matrix Factorization (ONMF)* algorithms (e.g., [25, 26]). Instead of learning the final NMF decomposition of an infinite stream of spectral frames over time by updating B with every incoming spectral frame, we revert to B_p prior to the NMF decomposition of every individual frame.

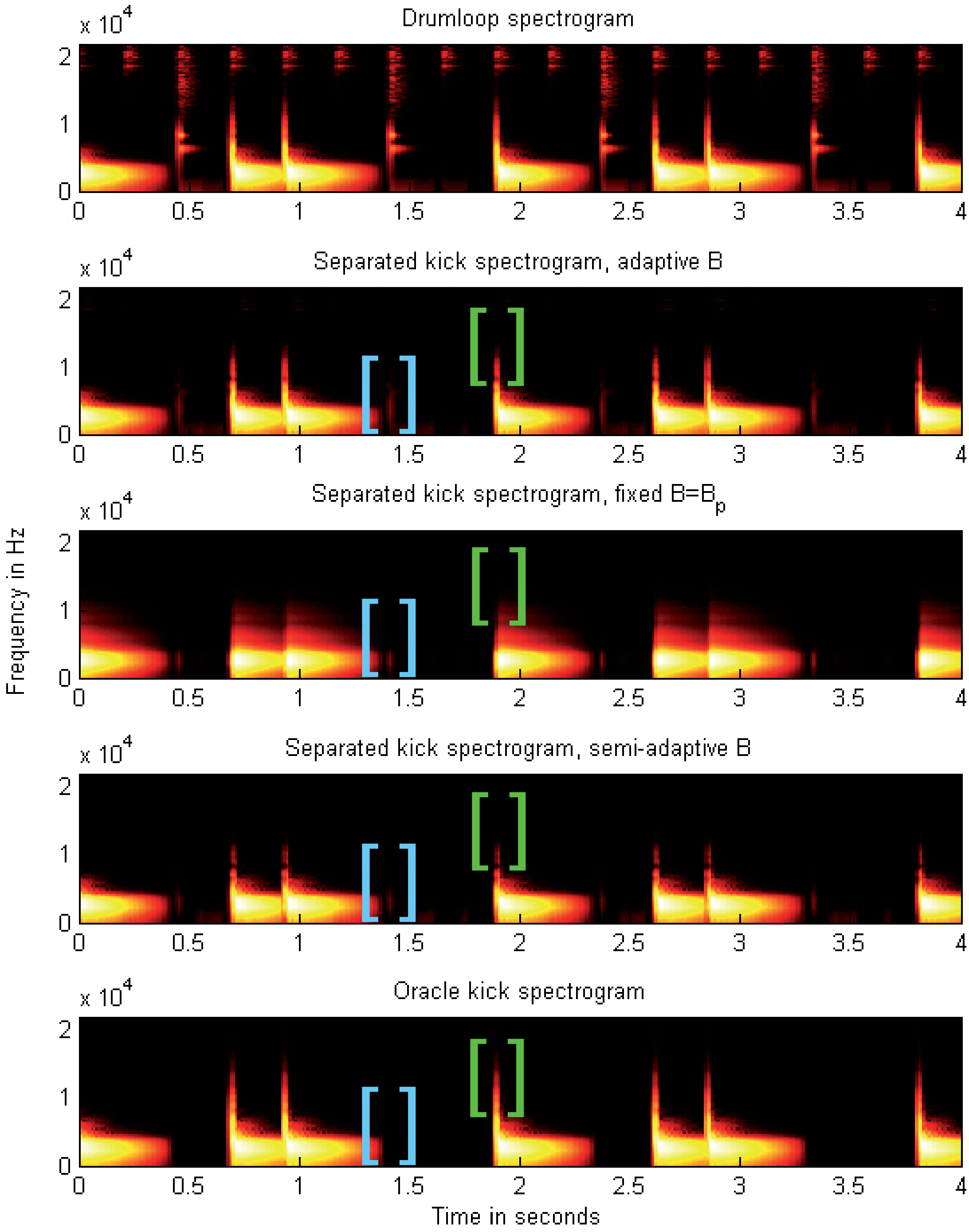

Figure 4: *Comparison of drum loop spectrograms obtained from the different decomposition methods. The top spectrogram is obtained from the input drum loop. The bottom spectrogram shows the idealized (oracle) target when separating the kick. The second, third and fourth spectrogram show the separation results obtained with adaptive B, fixed $B = B_p$ and semi-adaptive B, respectively. The kick separated using fixed B is clearly inferior compared to the oracle kick. This is evident by the smeared transient (light green brackets). It does exhibit less cross-talk from the snare (light blue brackets) yielding better transcription results than adaptive B (see Sec. 5.2). Thus, semi-adaptive B seems to be the optimal compromise between both.*

4.4. Onset detection

After decomposition, frame-wise matrix multiplication of the activations in G corresponding to a single drum with the corresponding columns in B yields well separated individual spectrograms for kick, snare and hi-hat. Based on these, onset detection is performed in a straightforward manner by means of peak-picking. While other authors used the amplitude envelopes in G directly, we encounter different spectra in every frame for the adaptive and semi-adaptive bases. Thus, we take the extra step of spectrogram reconstruction prior to onset detection. Following the approach proposed in [27], novelty curves D are extracted from the successive spectral frames for each drum by differentiating the logarithmic magnitude along time. Afterwards, all bins per frame are summed up and half-wave rectification is applied since only salient positive peaks corresponding to onsets are of interest. Inevitably, cross-talk artifacts that can occur due to imperfect decomposition may lead to erroneous spikes that can be mistaken as drum onsets. Thus, an adaptive threshold procedure is applied to the novelty curve. The threshold T is derived by element-wise nonlinear compression $D^{0.5}$, subsequent application of an exponential moving average filter and nonlinear expansion of the result $D^{2.0}$. A variable boost factor b can be used to adjust the additive offset of T manually. This is done by simply multiplying b with the arithmetic mean of T and adding the result to T. In real-time mode, the long-term arithmetic mean is derived by a frame-wise iterative update. If the novelty curve rises above T for several frames and fulfills additional plausibility criteria (see [16]), it is marked as an onset. Finally, the onset detection stage returns a list of onset times per drum instrument, yielding the final transcription result.

5. EVALUATION

In order to assess the drum transcription performance, experiments with manually transcribed drum set recordings were conducted. The well known Precision, Recall and F-measure were used as evaluation metrics with a tolerance of 50 ms between annotated and detected onsets.

5.1. Test data

A training set was created for initialization of single drums (kick, snare, hi-hat) in [3]. In order to capture the individual characteristics, the drums were hit separately with varying velocity. For recording, an overhead microphone at a fixed height of 1 m was used. The recordings were made with 10 different drum kits, consisting of different drum sizes and a broad range of materials. The size of the kick drum ranges from 18 inch to a 24 inch diameter, and depths of 16 inch up to 22 inch. Materials were birch, mahogany or maple. The snare drums all had the same size of 14 inch diameter and 6.5 inch in depth but different materials (such as metal, wood or acrylic). The sizes for hi-hat ranged from 13 inch to 15 inch. A second subset was generated using sample-based drum sets from the BFD[6] plug-in. The third part of the set featuring purely synthetic drum kits was generated using Steinberg's Groove Agent[7] plug-in. The onsets were transcribed manually by an experienced drummer using the software Sonic Visualiser [28].

In total, the test set consisted of 33 drum sequences which were fairly simple groove patterns of kick, snare and hi-hat. The tempo of the performed drum rhythms varies between 100 and 140 BPM. Overall, 10 minutes of audio were recorded (in 44.1 kHz, mono, 16 Bit) resulting in 3471 annotated onsets. The shortest annotated interval between consecutive onsets is 107 ms (16th note at 140 BPM). The combined data-set is available online as a public benchmark for drum transcription[8].

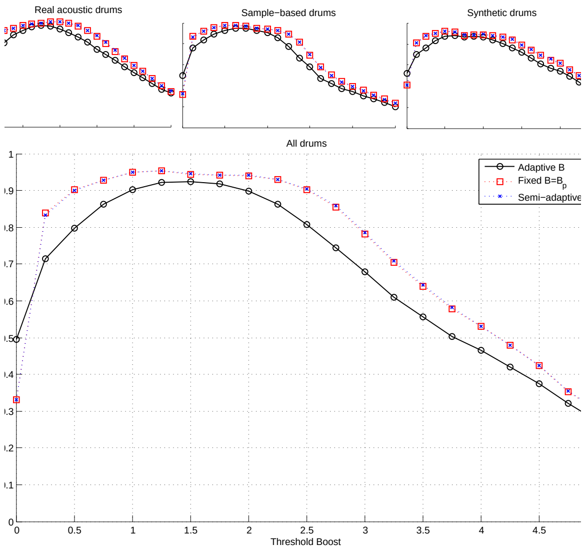

Figure 5: *Overview of transcription F-measure rates versus the threshold boost b on the different drum sets. The largest plot shows the combined results for all drum sets. The F-measures for fixed $B = B_p$ and semi-adaptive B are very similar, thus the corresponding curves are almost indistinguishable.*

5.2. Results

Using the described test data, an extensive grid search was performed in order to estimate the optimal set of parameters. We omit the details and just explain that the most influential parameters were the threshold boost and the number of NMF iterations used during decomposition. The best average F-measure of 0.95 across all drum kits was obtained with $H = 512$ samples hop-size, $N = 2048$ bins spectrum size, $b = 1.25$ threshold boost, $K = 25$ NMF iterations and $\beta = 4$ blending non-linearity in case of the semi-adaptive bases. Most surprisingly, the acoustic and sample-based drum kits lead to better F-measure scores than

[6] www.fxpansion.com/BFD

[7] http://www.steinberg.net/en/products/vst/groove_agent/groove_agent.html

[8] http://www.idmt.fraunhofer.de/en/business_units/smt/drums.html

the synthetic drum kits. This is somewhat counter-intuitive, since we expected the drum transcription performance to decrease when dealing with drum recordings under larger natural variation in the single drum sounds. We interpret this as a benefit of the semi-adaptive bases, which can be seen by the comparison between the different approaches in Figure 5. There, we show the influence of b on the F-measure scores across the different drum kits as well as the three different adaption degrees of the spectral bases. It can clearly be seen, that the adaptive B yield slightly worse transcription results, which we account to the more pronounced cross-talk artifacts. Differences between F-measure scores of fixed B and semi-adaptive B are extremely small. Nevertheless, the discussion in Sec. 5.3 shows that fixed basis vectors have their weaknesses when the drum sounds to be separated exhibit high spectral variability over time.

5.3. Influence of basis adaption

We present an illustrative example for the different degrees of adaptivity. The uppermost plot in Figure 4 shows the spectrogram of a synthetic drum loop consisting of kick, snare and hi-hat playing the rhythm given in Figure 1. It should be noted that the magnitude of the spectrograms has been converted to dB and has been resampled to a logarithmically spaced frequency axis for visualization purposes only. The bottom plot shows the oracle spectrogram of the kick playing in isolation. This kick, sampled from a Roland TR 808 drum computer, is obviously rather invariant across the repeated onsets but exhibits a very time-dynamic behavior per onset. One can clearly see a strong vertical head-transient caused by the sharp attack. Afterwards, a slightly decreasing center frequency can be observed in the tail. In the second plot of Figure 4 we see the kick spectrogram obtained from NMF decomposition with adaptive bases. The third plot shows the approximation of the kick spectrogram achieved with only one fixed spectral basis vector per drum instrument. The modeling of the attack transient is inferior, since it is smeared into the tail of the drum sound. The fourth plot shows the kick spectrogram resulting from decomposition with semi-adaptive spectral bases. When compared to the oracle spectrogram, one can clearly see that the attack transients are preserved very well. On closer inspection, all NMF variants exhibit cross talk from hi-hat and snare in the kick spectrogram (marked with light blue brackets). They are most pronounced for the fully adaptive B and can cause erroneous onset candidates during onset detection (see Sec. 4.4).

5.4. Real-time capability

The proposed algorithm has been implemented as VST plugin. A screen-shot of the user interface is shown in Figure 6. Three different drum sounds can be trained via live input or prepared audio files. Alternatively, artificial spectral basis templates can be used and refined in an iterative update. The plugin works in quasi-real-time, the systemic delay is only dependent on the input delay of the audio hardware and the used hop-size. For the optimal parameter settings given in Sec. 5.2, we could measure a delay of approximately 6 ms.

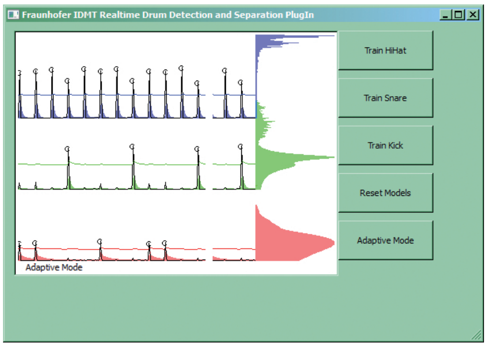

Figure 6: *Screen-shot of a VST plug-in encapsulating the proposed algorithm. The semi-transparent colored curves visualize the extracted G of the individual drums, the colored spectra on the right show the extracted B. Blue corresponds to hi-hat, green to snare and red to kick. The individual onset detection functions D are overlayed as black lines and the dynamic thresholds T as solid colored lines.*

6. CONCLUSIONS

This paper presented a method for real-time transcription and separation of drum sounds from drum set recordings. It is based on NMF decomposition initialized with prior spectral basis templates for the expected drums. Under the assumption, that the isolated drum sounds are available for training, the transcription performance for polyphonic drum input featuring the specific instruments is on par with state-of-the-art results. The novel concept of semi-adaptive spectral bases does not yield improvements in transcription but seems promising for enhancing the perceptual quality of drum sound separation. Our collected data-set used for evaluation is contributed to the research community in order to foster reproducible research results. Future work will be directed to systematically evaluate alternative decomposition strategies, such as ONMF and NMFD. Furthermore, the applicability to a larger variety of different drum instruments (toms, cymbals, etc.) will be assessed allowing the inclusion of commonly used test corpora, such as the ENST drums data-set.

7. ACKNOWLEDGMENTS

Parts of this research have been developed during the SyncGlobal project. SyncGlobal was a 2-year collaborative research project between Piranha Womex AG, Bach Technology GmbH, 4FriendsOnly AG and the Fraunhofer IDMT in Ilmenau, Germany. The project was co-financed by the Germany Ministry of Education and Research in the framework of the SME innovation program (FKZ 01/S11007).

8. REFERENCES

[1] M. Plumbley, S. Abdallah, J. P. Bello, M. Davies, G. Monti, and M. Sandler, "Automatic music transcription and audio source separation," *Cybernetics & Systems*, vol. 33, no. 6, pp. 1–21, 2002.

[2] C. Dittmar, E. Cano, S. Grollmisch, J. Abeßer, A. Männchen, and C. Kehling, *Springer Handbook for Systematic Musicology*, chapter Music Technology and Music Education, Springer, 2014.

[3] F. Weber, "Development of a real-time algorithm for drum-

sound detection," Diploma thesis, Ilmenau University of Technology, 2013.

[4] J. Paulus, *Signal Processing Methods for Drum Transciption and Music Structure Analysis*, Ph.D. thesis, Tampere University of Technology, Tampere, Finland, 2009.

[5] M. Casey, "Separation of mixed audio sources by independent subspace analysis," in *Proceedings of the International Computer Music Conference*, 2000.

[6] C. Uhle, C. Dittmar, and T. Sporer, "Extraction of drum tracks from polyphonic music using independent subspace analysis," in *Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation*, 2003.

[7] M. Plumbley, "Algorithms for non-negative independent component analysis," *IEEE Transactions on Neural Networks*, vol. 14, pp. 30–37, 2003.

[8] C. Dittmar and C. Uhle, "Further steps towards drum transcription of polyphonic music," in *Proceedings of the AES 116th Convention*, 2004.

[9] D. FitzGerald, B. Lawlor, and E. Coyle, "Prior subspace analysis for drum transcription," in *Proceedings of the 114th AES Convention 114th Convention*, 2003.

[10] A. Spich, M. Zanoni, A. Sarti, and S. Tubaro, "Drum music transcription using prior subspace analysis and pattern recognition," in *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx)*, 2010.

[11] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, 2001.

[12] M. Helén and T. 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 (EUSIPCO)*, 2005.

[13] J. Paulus and T. Virtanen, "Drum transcription with non-negative spectrogram factorisation," in *Proceedings of the 13th European Signal Processing Conference (EUSIPCO)*, 2005.

[14] M.N. Schmidt and M. Mørup, "Nonnegative matrix factor 2-d deconvolution for blind single channel source separation," in *Independent Component Analysis and Blind Signal Separation*, pp. 700–707. Springer, 2006.

[15] P. Smaragdis, "Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs," in *Independent Component Analysis and Blind Signal Separation*, pp. 494–499. Springer, 2004.

[16] E. Battenberg, V. Huang, and D. Wessel, "Live drum separation using probabilistic spectral clustering based on the itakura-saito divergence," in *Proceedings of the AES 45th Conference on Time-Frequency Processing in Audio*, Helsinki, Finland, 2012.

[17] K. Yoshii, M. Goto, and H. Okuno, "Automatic drum sound description for real-world music using template adaption and matching methods," in *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)*, 2004.

[18] C. Dittmar, D. Wagner, and D. Gärtner, "Drumloop separation using adaptive spectrogram templates," in *Proceedings of the 36th Jahrestagung fuer Akustik (DAGA)*, 2010.

[19] A. Maximos, A. Floros, M. Vrahatis, and N. Kanellopoulos, "Real-time drums transcription with characteristic bandpass filtering," in *Proceedings of the 7th Audio Mostly Conference: A Conference on Interaction with Sound*, 2012.

[20] O. Gillet and G. Richard, "Automatic transcription of drum loops," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2004.

[21] K. Tanghe, S. Degroeve, and B. De Baets, "An algorithm for detecting and labeling drum events in polyphonic music," in *Proceedings of the 1st Annual Music Information Retrieval Evaluation eXchange (MIREX '05)*, 2005.

[22] D. Fitzgerald, *Automatic drum transcription and source separation*, Ph.D. thesis, Dublin Institute of Technology, Dublin, Ireland, 2004.

[23] S. Ewert and M. Müller, "Score-informed voice separation for piano recordings," in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, 2011.

[24] M. N. Schmidt, *Single-channel source separation using non-negative matrix factorization*, Ph.D. thesis, Technical University of Denmark, Aalborg, Denmark, 2008.

[25] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal on Machine Learning Research*, 2010.

[26] B. Cao, D. Shen, J.T. Sun, X. Wang, Q. Yang, and Z. Chen, "Detect and track latent factors with online nonnegative matrix factorization," in *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, 2007, pp. 2689–2694.

[27] P. Grosche and M. Müller, "Extracting predominant local pulse information from music recordings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1688–1701, 2011.

[28] C. Cannam, C. Landone, and M. Sandler, "Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files," in *Proceedings of the International Conference on Multimedia*, 2010, pp. 1467–1468.

[29] A. Zils, F. Pachet, O. Delerue, and F. Gouyon, "Automatic extraction of drum tracks from polyphonic music signals," in *Proceedings of the 2nd International Conference on Web Delivering of Music (WedelMusic2002)*, 2002, p. 5.

[30] J. Paulus and A. Klapuri, "Conventional and periodic n-grams in the transcription of drum sequences," in *Proceedings of the IEEE International Conference Multimedia and Expo*, 2003.

[31] J. Paulus and A. Klapuri, "Drum sound detection in polyphonic music with hidden markov models," *EURASIP Journal on Audio, Speech, and Music Processing*, 2009.

[32] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[33] J. Abeßer and O. Lartillot, "Modelling musical attributes to characterize two-track recordings with bass and drums," in *Proceedings of the International Society of Music Information Retrieval (ISMIR)*, 2011.