DRUM DETECTION BASED ON NMF DECOMPOSITION

Ina Medebach, Markus Wende, Robert Pelzer

Audiocommunication and -technology
Technische Universität Berlin
{medebach,wende,pelzer}@campus.tu-berlin.de

ABSTRACT

In this paper we present a method to detect onsets in a polyphonic drum mixture. We use a semi - adaptive non-negative matrix factorization to separate single drums from a drum loop mixture, create a novelty curve and finally identify the played onsets using a dynamic threshold. We follow an approach that was introduced by Dittmar and Gärtner with the Fraunhofer IDMT. We also use their data set and compare the results. For further comparison with other work we use a modified data set with additional harmonic information. The best results (F-score of 92%) are accomplished with drum loop and training set from the same set and an F-score of 67% is achieved by adding harmonic information to the drum loop.

Index Terms— onset detection, none negative matrix factorization, novelty curve

1. INTRODUCTION

Drum detection is a common task in the field of music information retrieval (MIR) and is often used to provide additional data for recordings of acoustic drums. It can also be used to provide a midi track for a following analysis or to overdub an acoustic drum set in real-time with sampled sounds and hence, turning the acoustic drum set into something similar to a midi drum set. The greatest challenge in performing this task are the polyphonic drum sounds that are due to the nature of drum beats with simultaneous hits of Kick Drum, Snare Drum and Hi Hat, etc. Previous work (see Sec. 2) has shown that non-negative-matrix-factorization (NMF) can be used to decompose the magnitude spectrum of a drum loop into the underlying instruments of a drum set. The data set in Sec. 3 is used to generate a spectra of each training set and polyphonic drum mixture using Short-Time-Fourier-Analysis (STFT). The algorithm of the semi-adaptive NMF, shown in Sec. 4, is then used to decompose the different instruments from the test set and induce a novelty curve. By applying a threshold and choosing appropriate criteria the onsets can be selected. An overview is shown in Fig. 1. For the Evaluation in Sec. 5 we produce F-scores for each instrument and in total. Concluding in Sec. 6 and 7 the results are shown and further work is named.

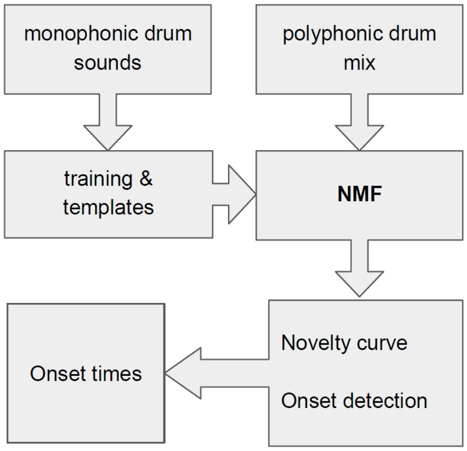

Fig. 1: Approach of the drum detection algorithm

2. RELATED WORK

Since Lee and Seung [1] introduced multiplicative update rules for NMF, this tool has been widely used by various MIR research groups. One of the most recent approaches is introduced by Wu and Lerch [2]. Their aim is to detect percussive events in complex mixtures of music with a minimal training set in which the dictionary is being pre-initialized with pre-defined drum templates. They conduct an adaption of the percussive templates that is partially fixed, which means that it stays fixed during the decomposition process and is only updated based on the results of previous convergence. They achieve F-scores of 0.78 and 0.72 in monophonic and polyphonic music for detecting 3 classes of drums. Another approach is introduced by Gärtner and Dittmar [3] who work with a semi-adaptive system which focuses on real-time transcription. They test their system with a simpler data set as the one that was mentioned by Wu and Lerch because it contains only the percussive instruments they wish to detect and no additional harmonic spectrum. Gärtner and Dittmar achieve a F-score of 95%.

This work mainly follows after the approach introduced by Dittmar and Gärtner and tries to confirm their results by working with the same data set. Additionally we also modified the data set by adding a harmonic signal to each drum loop for a better comparison with Wu and Lerch's result.

3. DATA SET

The data set used for this work was provided and created by Dittmar and Gärtner [3]. In total, 10 different drum sets were used to record a total of 140 test files with matching training files. The drum sets had a variety from classically played to techno styled sounds and used only three classes of drums: Snare Drums, Hi Hats and Kick Drums. For each instrument, they provided one training file. This file included samples of the used drum with different velocities and styles of playing, i.e. opened or closed Hi Hat. Additionally Dittmar and Gärtner also provided hand-annotated onsets for each test file as ground truth to evaluate the algorithm.

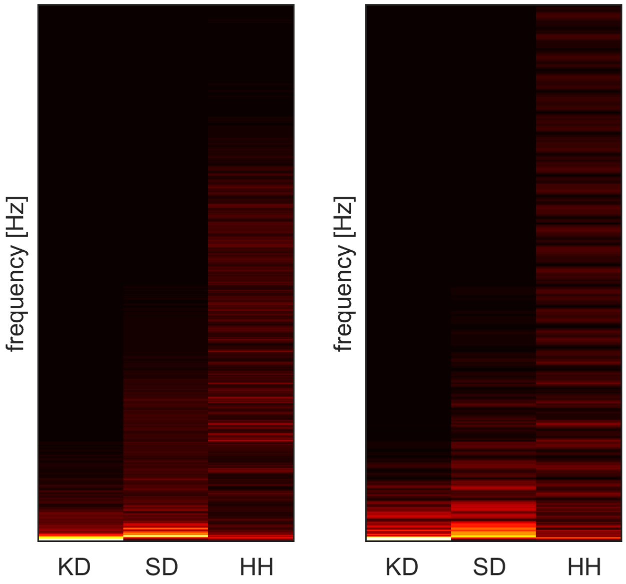

Fig. 2: Basis matrix W_{init} (left) and W (right) for Kick Drum (KD), Snare Drum (SD) and Hi Hat (HH)

4. ALGORITHM OVERVIEW

4.1. Semi-adaptive NMF

Following the approach in [3] we compute a basis vector for each drum instrument by conducting a training phase in which a STFT is being applied for the corresponding training set. The magnitude spectrum of the training set is then being averaged along the time axis. The basis matrix W_{init} (shown in Fig. 2 on the left side), which contains the basis vectors for Kick Drum, Snare Drum and Hi Hat is then delivered to the NMF function. On the mixed drum signal a STFT is also performed. We use NMF in a frame-wise procedure, which means that the signals magnitude spectrum is handed to the NMF calculation frame-by-frame together with the basis matrix. As shown in [3] we use the Kullback-Leibler Divergence (KL) for the update rules introduced by Lee and Seung [1]. The NMF computation performs a certain amount of iterations in which both, the spectral basis (1) and the amplitude envelopes (2) are being updated.

$$W \leftarrow W \bullet \frac{\left(V \bullet \Lambda^{\beta-2}\right) H^T}{\Lambda^{\beta-1} H^T} \tag{1}$$

$$H \leftarrow H \bullet \frac{\left(V \bullet \Lambda^{\beta-2}\right) W^T}{\Lambda^{\beta-1} W^T} \tag{2}$$

Following Dittmar and Gärtner we implemented a semi-adaptive approach in which the basis function W is being updated by every iteration. This means that the basis functions are allowed to deviate more from the initial state W_{init} towards the iteration limit. The difference between W_{init} and W is also visualized in Fig. 2. The weighting is conducted non-linearly according to (3) and (4).

$$W = \alpha \times W_{init} + (1 - \alpha) \times W \tag{3}$$

$$\alpha = (1 - \frac{k}{K})^\beta \tag{4}$$

Where α is the blending parameter, k the iteration counter and K the number of NMF iterations per frame. To use the Kullback-Leibler divergence β has to be one. The gain matrix H, shown in Fig. 3, consist of the three instruments we are using, likewise the three columns of the basis matrix W. By multiplying these two matrices we receive the approximated spectrum Λ, which converges to the original drum mixture V. The superscript T (e.g. in W^T) stands for the transposed vector W.

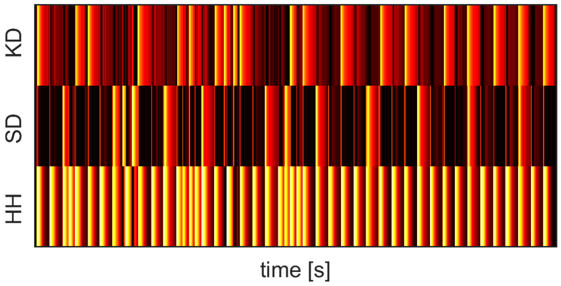

Fig. 3: Gain matrix H for Kick Drum (KD), Snare Drum (SD) and Hi Hat (HH)

4.2. Onset Detection

To perform the onset detection of the decomposed spectra, first the spectra for each instrument was reconstructed. The novelty curve D (which indicates the change over time in the spectrum) is generated by splitting the spectra in five different bands which are: $0Hz$ to $500Hz$, $500Hz$ to $1250Hz$, $1250Hz$ to $3125Hz$, $3125Hz$ to $7812.5Hz$ and 7812.5 to $fs/2$, where fs is the sample frequency. In the first step each band is logarithmized and secondly differentiated. After differentiation the positive derivatives are summed up for each band while the negative derivatives are being dismissed to focus on onsets rather than offsets. Afterwards the average of all bands was used for further processing. For these steps we used an adapted version of the "Tempogram Toolbox" developed by Peter Grosche and Meinard Müller [4] ($audio\_to\_noveltyCurve.m$, lines 120 -167). Normalization delivered the final novelty curve D with $D \leq 1$. Similar to [3] we calculated a dynamic threshold T by first applying a compression of D^2, secondly applying an exponential moving average filter and thirdly an expansion of D^2. The mean of the threshold was multiplied with a boost factor b and added to the threshold to get the final threshold. An example of the novelty curve and threshold is shown in Fig. 4.

An onset was detected by comparing the novelty curve and the threshold. If D rose above T for at least a number of N instances and only when D dropped below T after the last onset. To get the right values of the logarithm constant C, boost factor b and the number of instances N comprehensive iterations over these three parameters were performed over all 140 test files to find the optimal values. The following settings with the fix hopsize, framesize (blocksize) and the exponential value α to smooth the threshold, delivered the best results with intent to get the best F-scores:

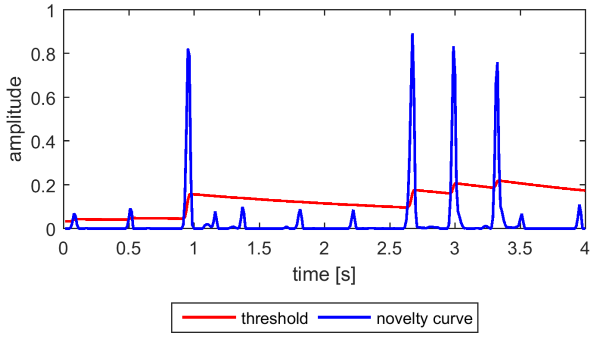

Fig. 4: Novelty curve and threshold of a Snare Drum (SD)

- hopsize of 512 frames
- framesize of 2048 frames
- logarithm constant $C = 1$
- boost factor $b = 0.3$
- number of instances $N = 3$
- exponential moving average filtering with $\alpha = 0.01$

For further information on onset detection and plausibility criteria, the work of [4] is highly recommended.

5. EVALUATION

To evaluate our Algorithm, we calculated the F-score for the found time onsets of our algorithm and the annotated ground truth of the data set. In Fig. 5 a sample result of the detected onsets compared to the ground truth is shown. The F-score is a single value, which measures the accuracy of the predicted onsets. It combines the precision and the recall shown in (5).

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{5}$$

Precision describes the ration between correct predicted onsets to all (wrongly and correctly) predicted onsets. Recall, on the other side, describes the ratio between correct predicted to all actual onsets.

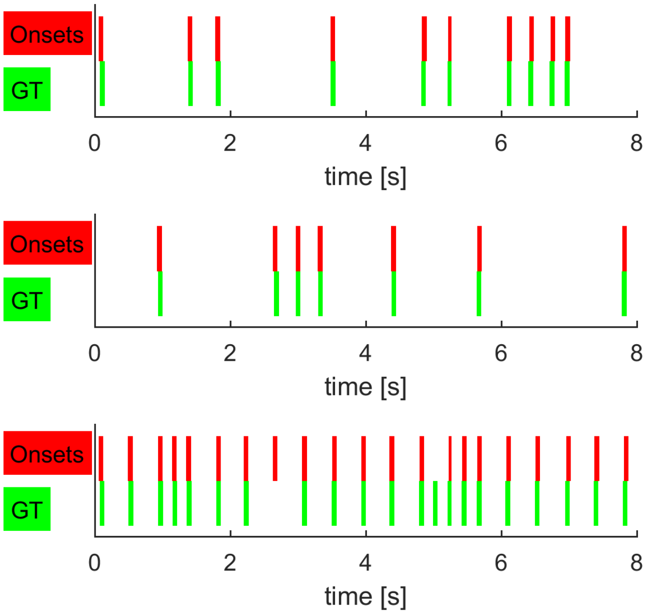

Fig. 5: Onsets and ground truth (GT) annotations of an example computation of our algorithm. The instruments are in the following order: Kick Drum (top), Snare Drum (middle) and Hi Hat (Bottom)

For further testing we created five different tasks. This allows a direct comparison with the achieved accuracy by Dittmar and Gärtners as well as a testing of the algorithm for more relevant and realistic purposes. The first task compares the test file with their own training data. This task was also used by Dittmar and Gärtner to evaluate their algorithm and represents the best case scenario. Task two to four evaluate harder scenarios within the data set and the final task, task 5 describes the most realistic case by adding a real song to the test data.

For the second task, we applied only one the training set WaveDrum02_56 to all test files. To generate an average training set, in task three we compared all 95 training sets to an average W and tested it with each test set. Task four compares an average W to unknown drums. Therefore we used the average of 45 different training files to the remaining 50 test files.

In the last test, task five, we evaluated the algorithm on a more realistic assignment. In realistic situations the algorithm should also able to predict onsets in real songs with guitars and vocals. For this reason we mixed Bon Ivers ”Towers” to all of the test files and tested the mixes the same way as in task one. The F-scores were calculated for each instrument and averaged over all test files. We allowed a tolerance up to 90 ms between predicted onsets and the ground truth.

	Kick Drum	Snare Drum	Hi Hat	Average
task 1	95%	92%	88%	92%
task 2	94%	81%	88%	88%
task 3	95%	82%	87%	88%
task 4	95%	82%	89%	89%
task 5	67%	50%	83%	67%

Table 1: F-scores of all tasks:
task 1: test files with corresponding training files,
task 2: one training file for all test files,
task 3: average training file for all test files,
task 4: average training file for part of the test files,
task 5: modified test files with added harmonic signal

6. RESULTS AND DISCUSSION

As can be seen in table 1 with an overall mean of 95% for Kick drums, task one reaches the 95 % that are also reported by Dittmar and Gärtner. Even the mean over all three instruments reaches with 92% nearly their achieved, but not further specified, accuracy. The second to fourth task scored equally with an overall average with approximate 88% while the last task was only able to generate a score of 67%. Overall we are confident to say, that our algorithm works good for Hi Hats. In all tasks it achieves a F-score of 88%. If there are no distortions, the Kick Drum achieves even higher results of 94% to 95%, but decreases in accuracy to 67% when guitar and vocals are added. The same happens for the Snare Drum, but even when it is not compromised the scores vary between 81% and 92%.

7. CONCLUSION

This paper presents a common way of drum transcription and evaluation. Our approach to solve this consists of two key components: the NMF and the novelty curve. The NMF separates the single instruments. The implementation of the novelty curve offers the ability to easily adjust thresholds and the acceptance criteria. Together they form a well functioning drum detection algorithm. As the evaluation shows, the algorithm works with an F-score of 92% for Dittmar and Gärtner’s data set. Thanks to the semi-adaptive basis function approach, the algorithm is able to achieve high F-scores even in more complex tasks. Our algorithm reaches state-of-the-art results, but has potential to increase its accuracy by testing more data sets and adjusting parameters. Further work will also include additional basis functions. As shown in task three, the long term goal would be being able to recognize drums without necessarily implementing an individual training phase for each drum set. As a prerequisite we want to use several different drums to generate a wider spectrum of different drum sounds. The difference to task three is, that we will not create an average instrument, but select several typical drum sounds and calculate the amplitude for each one separately. Afterwards the algorithm will for instance be able to decide whether a new sound is likely to be a kick drum or something else.

8. REFERENCES

[1] Daniel D. Lee and H. Sebastian Seung, “Algorithms for non-negative matrix factorization,” in *Advances in Neural Information Processing Systems 13*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds., pp. 556–562. MIT Press, 2001.

[2] Chih-Wei Wu and Alexander Lerch, “Drum transcription using partially fixed non-negative matrix factorization with template adaptation,” in *ISMIR*, 2015, p. 257–263.

[3] Christian Dittmar and Daniel Gärtner, “Real-time transcription and separation of drum recordings based on nmf decomposition,” in *DAFx*, 2014, p. 187–194.

[4] Peter Grosche and Meinhard Müller, “Extracting predominant local pulse information from music recordings,” in *IEEE Transactions on audio, speech and language processing, Vol. 19, NO. 6*, 2011, p. 1688–1701.