DRUM TRANSCRIPTION VIA CLASSIFICATION OF BAR-LEVEL RHYTHMIC PATTERNS

Lucas Thompson, Matthias Mauch and Simon Dixon
Centre for Digital Music, Queen Mary University of London
contact@lucas.im, {m.mauch, s.e.dixon}@qmul.ac.uk

ABSTRACT

We propose a novel method for automatic drum transcription from audio that achieves the recognition of individual drums by classifying bar-level drum patterns. Automatic drum transcription has to date been tackled by recognising individual drums or drum combinations. In high-level tasks such as audio similarity, statistics of longer rhythmic patterns have been used, reflecting that musical rhythm emerges over time. We combine these two approaches by classifying bar-level drum patterns on sub-beat quantised timbre features using support vector machines. We train the classifier using synthesised audio and carry out a series of experiments to evaluate our approach. Using six different drum kits, we show that the classifier generalises to previously unseen drum kits when trained on the other five (80% accuracy). Measures of precision and recall show that even for incorrectly classified patterns many individual drum events are correctly transcribed. Tests on 14 acoustic performances from the ENST-Drums dataset indicate that the system generalises to real-world recordings. Limited by the set of learned patterns, performance is slightly below that of a comparable method. However, we show that for rock music, the proposed method performs as well as the other method and is substantially more robust to added polyphonic accompaniment.

1. INTRODUCTION

The transcription of drums from audio has direct applications in music production, metadata preparation for musical video games, transcription to musical score notation and for musicological studies. In music retrieval, robust knowledge of the drum score would allow more reliable style recognition and more subtle music search by example. Yet like related tasks such as polyphonic piano transcription [1], a versatile, highly reliable drum transcription algorithm remains elusive.

Audio drum transcription methods have been classified into two different strategies [10, 18]: *segment and classify* and *separate and detect*. Systems in the first category detect a regular or irregular event grid in the signal, segment the signal according to the grid, extract features such as MFCCs [19] or multiple low-level features [23] and then classify the segments using Gaussian Mixture Models [19], k nearest neighbour classification [21], or Support Vector Machines [23]. Systems in the second category first detect multiple streams corresponding to drum types, usually via a signal or spectral decomposition approach, e.g. [2, 7], or simpler sub-band filtering [15], and then identify onsets in the individual streams. Other methods combine aspects of both categories, via adaptation [24] or joint detection of onsets and drums [18]. To ensure temporal consistency (smoothness) many approaches make use of high-level statistical models that encode some musical knowledge, e.g. hidden Markov models [18]. The methods greatly differ in terms of the breadth of instruments they are capable of detecting; most detect only bass drum, snare drum and hi-hat [7, 14, 18, 23] or similar variants, probably because these instruments (unlike crash and ride cymbals) can be represented in few frames due to their very fast decay.

Despite the evident diversity of strategies, all existing methods aim directly at detecting individual or simultaneous drum events. As we will see later, our approach is qualitatively different, using higher-level patterns as its classi-

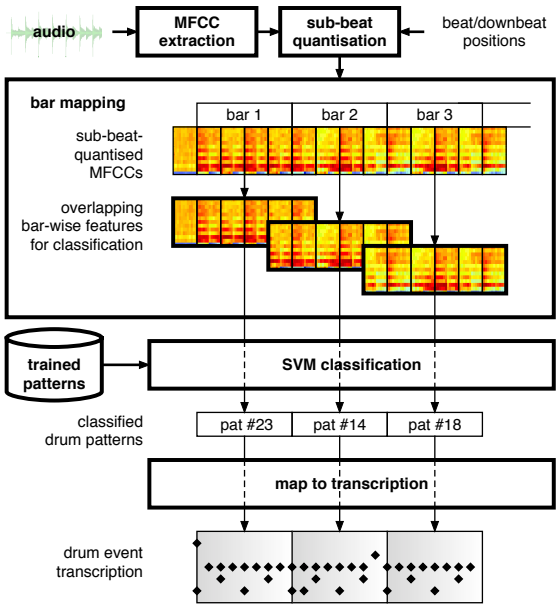

Figure 1. Overview of the system at prediction time.

fication target. One advantage of this is that the long decay of cymbals is naturally modelled at the feature level.

Since drum transcription from polyphonic audio is only partially solved, music information retrieval tasks rely on "soft" mid-level audio features to represent rhythm. Fluctuation patterns [17] summarise the frequency content of sub-band amplitude envelopes across 3-second windows and were used to evaluate song similarity and to classify pop music genres; they have also been used to describe rhythmic complexity [13]. Bar-wise rhythmic amplitude envelope patterns have been shown to characterise ballroom dance music genres [5] and bar-wise pseudo-drum patterns have been shown to correlate with popular music genres [6]. Rhythmic patterns have also formed the basis of beat tracking systems [11] and been used for downbeat detection [8]. These methods share a more musically holistic approach to rhythm, i.e. they summarise rhythmic components in a longer temporal context. Drum tutorials, too, usually focus on complete rhythms, often a bar in length, because "with the command of just a few basic rhythms you can make your way in a rock band" [22]. In fact, we have recently shown that drum patterns are distributed such that a small number of drum patterns can describe a large proportion of actual drum events [12].

Motivated by this result and by the effectiveness of more holistic approaches to rhythm description, we propose a novel drum transcription method based on drum pattern classification. Our main contribution is to show that the classification of bar-length drum patterns is a good proxy for predicting individual drum events in synthetic and real-world drum recordings.

2. METHOD

The proposed method is illustrated in Figure 1. It can broadly be divided into two parts: a feature extraction step, in which MFCC frame-wise features are calculated and formatted into a bar-wise, sub-beat-quantised representation, and a classification step, in which bar-wise drum patterns are predicted from the feature representation and then translated into the desired drum transcription representation. For the sake of this study, we assume that correct beat and bar annotations are given.

2.1 Feature extraction

Following Paulus and Klapuri [19], we choose Mel-frequency cepstral coefficients (MFCCs) as basis features for our experiments. MFCCs are extracted from audio sampled at 44.1 kHz with a frame size of 1024 samples (23ms) and a hop size of 256 samples (6ms), using an adaptation of the implementation provided in the VamPy plugin examples. [1] We extract 14 MFCCs (the mentioned implementation uses a bank of 40 Mel-filters) per frame, but discard the 0^{th} coefficient to eliminate the influence of overall signal level.

In order to obtain a tempo-independent representation, we assume that we know the positions of musical beats and quantise the feature frames into a metrical grid. This is needed for subsequent bar-wise segmentation. Whereas beat-quantised chroma representations usually summarise chroma frames within a whole inter-beat interval [16], drum information requires finer temporal resolution. Hence, following [12] we choose 12 sub-beats per beat, which is sufficient to represent the timing of the most common drum patterns. The MFCC frames belonging to each sub-beat are summarised into a single value by taking the mean over the sub-beat duration to give 12 quantised frames per beat.

Since we assume we know which beat is the downbeat, it is now trivial to extract bar representations from sub-beat-quantised MFCC features. For example, in a $\frac{4}{4}$ time signature, one bar corresponds to $4 \times 12 = 48$ sub-beat-quantised MFCC frames. However, slight deviations in timing and natural decay times of cymbals and drum membranes mean that information on a bar pattern will exist even outside the bar boundaries. For this reason we also add an extra beat either side of the bar lines (further discussion in Section 3), leading to the overlapping bar representations illustrated in Figure 1, each $6 \times 12 = 72$ frames long. The features we are going to use to classify $\frac{4}{4}$ bars into drum patterns will therefore comprise 936 elements (72 frames $\times$ 13 MFCCs).

2.2 Classification and transcription mapping

As our classifier, we use the one-vs-one multi class Support Vector Machine implementation provided in the *sklearn.svm.SVC* [2] package of the Python machine learning library, *scikit-learn* [20], with the default settings using a radial basis kernel, $K(x, x') = e^{-\gamma||x-x'||^2}$, where $\gamma = \frac{1}{N}$ and $N = 936$ is the feature dimension. Once the classifier has predicted a drum pattern for a particular bar, we perform a simple mapping step to obtain a drum transcription: using the information about the actual start and end time of the bar in the recording, each of the drum events that constitute the pattern are assigned to a time stamp within this time interval, according to their position in the pattern.

3. EXPERIMENTS AND RESULTS

We conducted three experiments to test the effectiveness of the proposed method, one with synthesised test data, and two with real recordings of human performances. In all experiments, the drum pattern data for training was encoded as MIDI and then synthesised using the FluidSynth software. Our drum pattern dictionary contains the top 50 most common drum patterns, including the empty pattern, in a collection of 70,000 MIDI files (containing only **bd** - kick, **sd** - snare, **hh** - closed hi-hat, **oh** - open hi-hat, **ri** - ride and **cr** - crash cymbals) [12]. [3] Figure 2 details how each drum class is distributed. Data examples and further

[1] http://www.vamp-plugins.org/vampy.html

[2] http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[3] http://www.eecs.qmul.ac.uk/~matthiasm/ndrum/patternstats/full_1-2-3-4-5-6/patternvisual_reduced

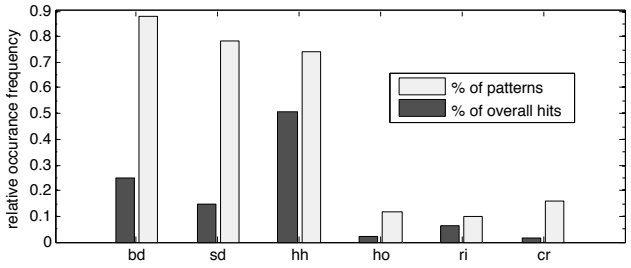

Figure 2. Relative occurrence of the drum classes in terms of overall number of drum events and the number of patterns containing each class. There are 50 patterns with an average of 11 events per pattern.

information can be found on the web page that accompanies this paper. [4]

We evaluate both the pattern classification performance and the quality of transcription of drum events. Pattern accuracy is defined as

$$A = \frac{\text{number of correctly classified bars}}{\text{total number of bars in test set}}. \qquad (1)$$

The transcription of drum events is evaluated using precision, recall and the F-measure (their harmonic mean)

$$P = \frac{N_c}{N_d}, \qquad R = \frac{N_c}{N}, \qquad F = \frac{2PR}{P+R}, \qquad (2)$$

where N_d is the number of detected drum hits, N_c is the number of correctly detected drum hits and N the number of drum hits in the ground truth. The individual drum hits are solely based on the presence or absence of a drum hit at a particular discrete position in the pattern grid used in the dictionary. In Sections 3.2 and 3.3 the ground truth drum hits, given as onset times, are quantised to a position in the grid. Tanghe's method [23] (Sections 3.2 and 3.3) is evaluated against the original ground truth with an acceptance window of 30ms, as in the original paper.

3.1 Multiple Synthesised Drum Kits

The aim of this experiment is to see how well the proposed classifier performs on synthetic data generated using multiple drum kits.

3.1.1 Training and Test Data

In order to create varied training and test data, we first generate 100 unique *songs*, each of which is simply a randomly permuted list of the 50 drum patterns from our dictionary. These *songs* are encoded as MIDI files, and we introduce randomised deviations in note velocity and onset times (velocity range 67–127, timing range $\pm$20 ms) to humanise the performances. All MIDI files are then rendered to audio files (WAV) using 6 drum kits from a set of SoundFonts we collected from the internet. In order to avoid complete silence, which is unrealistic in real-world scenarios, we add white noise over the entirety of each song at a SNR of 55 dB. We then calculate the bar-wise beat-quantised MFCC features as described in section 2.1. This yields a dataset of $6 \times 100 = 600$ files.

We use a random 70:30 train/test split of the 100 songs, where each of the 70 training songs appears in five variations synthesised from different drum kit SoundFonts. The remaining 30 songs, synthesised by the sixth drum kit, are used for testing. In order to assess performance on different drum kits, we cycle the use of the test drum kit in a leave-one-out fashion.

[4] http://www.eecs.qmul.ac.uk/~matthiasm/drummify/

drum-kit	overall drum events R	P	F	classification accuracy
00	98.9 (98.5)	98.9 (98.7)	98.9 (98.6)	91.1 (88.8)
01	97.1 (97.1)	97.6 (97.7)	97.4 (97.4)	89.2 (87.9)
02	98.3 (97.9)	98.3 (98.0)	98.3 (98.0)	87.7 (86.5)
03	84.8 (80.3)	82.3 (85.8)	83.6 (83.0)	50.1 (47.5)
04	92.7 (92.2)	91.2 (90.8)	92.0 (91.5)	72.0 (66.4)
05	97.2 (97.1)	98.5 (98.5)	97.9 (97.8)	91.6 (88.6)
mean	94.8 (93.9)	94.5 (94.9)	94.7 (94.4)	80.3 (77.6)

Table 1. Mean classification accuracy (%) and overall drum event R, P and F metrics for left out drum-kit from leave-one-out cross validation on 6 different drum-kits (see Section 3.1). Results for non-overlapping bars are in brackets.

drum-type	overall drum events R	P	F
bd	96.2 (96.0)	95.4 (95.0)	95.8 (95.5)
sd	96.5 (95.4)	99.3 (99.3)	97.8 (97.0)
hh	96.5 (95.3)	93.7 (94.8)	95.0 (95.0)
ho	59.9 (57.1)	77.3 (77.3)	61.1 (56.8)
ri	86.3 (86.4)	98.3 (99.5)	88.2 (89.0)
cr	84.4 (75.8)	97.0 (96.5)	89.3 (82.5)

Table 2. R, P and F for each drum type, taken over the whole test set and all kits from leave-one-out cross validation on 6 different drum-kits (see Section 3.1). Results for non-overlapping bars are in brackets.

3.1.2 Results

As Table 1 shows, our method achieves a high average accuracy of 80.3%, despite strong variation between drum kits. Irrespective of whether overlapping bar-wise features were used, the accuracy on drum kits 00, 01, 02 and 05 exceeds 85%. Performance is substantially worse on drum kits 03 and 04 (accuracies of 50.1% and 72.0%, respectively). Listening to a subset of the synthesised songs for drum kits 03 revealed that the recording used for the closed hi-hat sounds contains hi-hats that are slightly open, which is likely to cause confusion between the two hi-hat sounds.

To demonstrate the benefit of considering extra beats either side of the bar boundaries, Table 1 includes the results for non-overlapping bars. In this case we can see that the context given by the neighbouring beats increases classification accuracy (mean increase $\approx$3 percentage points). The greatest increase in accuracy ($\approx$6 percentage points) is observed in drum-kit 04.

To gain an insight into the types of patterns being misclassified, we consider those patterns for each drum-kit that are misclassified more than a quarter of the time. Figure 3 contains a few example cases. The single undetected

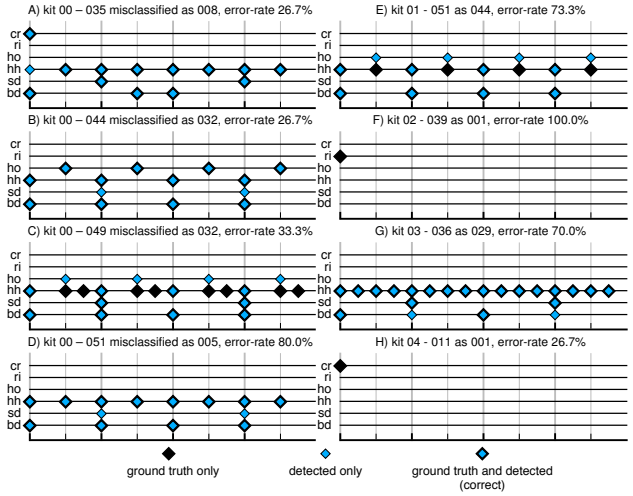

Figure 3. Examples of misclassified patterns (see Section 3.1.2).

ride or crash cymbals on the first beat in the ground-truth (cases F and H) are likely to be caused by the system confusing them for remainders of the previous bar. For cases A, B, D and G, the differences are subtle. In case A, the patterns differ by one hi-hat on the first beat. Cases B, D and G show that on occasions the classifier chooses a pattern where the majority of the drum events are correct, apart from a few inserted bass or snare drum events.

If we compare the individual drum events of the predicted pattern against the ground-truth and use precision and recall measures (see Table 2) we see that the system achieves high F-measures for the majority of drum classes (mean 0.88–0.97 for bd, sd, hh, ri, cr over all kits), but not for the open hi-hat class (mean F-measure 0.61).

Using audio features with overlapping bars leads to a substantial increase of over 8 percentage points in the recall of crash cymbal hits (84.4%) with respect to using no overlap (75.8%). The majority of the crash hits in our pattern dictionary occur on the first beat of the bar, and many of the patterns which were misclassified without the benefit of the overlapping neighbouring beats are such patterns, highlighting that the added context helps distinguish the pattern from those with a decaying hi-hat or other cymbal at the end of the previous bar. Note that since crash cymbals usually occur no more than once per bar, the classification accuracy in Table 1 shows larger improvement than the overall drum event precision and recall values.

3.2 Real drums Without Accompaniment

Having evaluated the performance of our system on synthesised data, we now test its robustness to real acoustic drum data.

3.2.1 Training and test data

We use the set of 100 songs described in the previous experiment (Section 3.1.1) synthesised on all 6 drum kits ($6 \times 100 = 600$ files). Since we have shown that overlapping bar-wise features provide higher accuracy (Section 3.1.2), we use only this feature configuration to train a re-usable model, which is used in the remainder of the experiments.

As test data we use the ENST-Drums database [9], which contains a wide range of drum recordings and ground-truth annotations of drum event onset times. We selected 13 *phrase* performances (15-25 s) which contain a number of similar patterns to ones in our dictionary, with expressional variations and fills, and one song from the *minus-one* category, a 60's rock song, which contains extensive variations and use of drum fills for which there are no similar patterns in our dictionary. In order to convert the provided ground-truth annotations to bar length drum pattern representations of the same format as those in our pattern dictionary, we annotated the beat and downbeat times in a semi-automatic process using Sonic Visualiser [3] and a Vamp-plugin implementation [5] of Matthew Davies' beat-tracker [4].

3.2.2 Results

The results for the ENST-Drums tracks are given in Table 3. The system's performance strongly varies by track. Our system performs particularly well on the disco and rock genre recordings (F-measure 0.479-0.924), for which our pattern dictionary contains very similar patterns. The shuffle-blues and hard-rock patterns perform much worse (F-measure 0.037–0.525), which is largely due to the fact that they utilise patterns outside our dictionary, bringing the mean F-measure down to 0.563. In order to understand the impact of out-of-dictionary patterns, Table 3 also provides the maximum possible F-measure F_{max} calculated from our dictionary by choosing the transcription that results in the highest F-measure for each bar, and computing the overall F-measure of this transcription.

For example, ENST recording 069 only achieves an F score of 0.288, falling short of $F_{max} = 0.583$, as it mostly consists of a typical shuffle drum pattern utilising the ride cymbal which is outside of the dictionary. However, the pattern which the system predicts is in fact one that contains a ride cymbal, from a total of five (see Figure 2). The hard rock recordings make extensive use of the open hi-hat, which is not utilised in the same fashion in our dictionary; here, the classifier most often predicts an empty bar (hence the very low scores). Note that all scores are obtained on a very diverse set of 6 drum and cymbal types.

For comparison, we obtained an implementation of an existing drum transcription method by Tanghe [23] and ran it on the ENST recordings, using the default pre-trained model. Since Tanghe's method only considers bass drum, snare drum and hi-hat, we constrain the evaluation to those drum types, and map the open and closed hi-hat events from our algorithm to single hi-hat events. Table 4 shows that our system has an F-measure of 0.73; Tanghe's system performs better overall (0.82), which is largely due to excellent bass drum detection. Note however that our system obtains better performance for the snare drum (F-measure 0.74 vs 0.70) particularly with respect to precision (0.93 vs

[5] `http://vamp-plugins.org/plugin-doc/qm-vamp-plugins.html#qm-barbeattracker`

	genre	tempo	detected drum events R	P	F	F_{max}
038	disco	slow	72.6	84.9	78.3	86.7
039	disco	medium	90.2	94.8	92.4	95.8
040	disco	fast	93.1	87.1	90.0	100.0
044	rock	slow	48.1	47.6	47.9	59.8
045	rock	medium	52.7	47.5	50.0	58.5
046	rock	fast	54.5	52.5	53.5	63.6
055	disco	slow	75.9	63.8	69.3	98.3
061	rock	slow	93.8	84.3	88.8	92.5
069	SB	slow	25.6	32.8	28.8	58.3
070	SB	medium	50.0	55.4	52.5	59.5
075	HR	slow	1.9	50.0	3.7	58.7
076	HR	medium	3.8	100.0	7.4	53.2
085	SB	slow	49.5	49.0	49.2	79.5
116	minus-one (60s rock)		76.8	77.1	77.0	81.7
mean			56.3	66.2	56.3	74.7

Table 3. Real drums without accompaniment: results in percent for ENST-Drums dataset. SB: shuffle-blues; HR: hard rock.

method	metric	bd	sd	hh	overall
Proposed	R	70.2	62.0	73.1	69.9
	P	60.6	92.7	83.5	76.3
	F	65.1	**74.3**	77.9	73.0
Tanghe et al.	R	87.0	65.0	89.8	83.8
	P	99.3	75.8	73.9	80.6
	F	**92.8**	70.0	81.1	**82.1**

Table 4. Real drums without accompaniment: Results in percent for drum classes reduced to bd, sd, hh (including ho) for comparison with Tanghe et al. [23].

0.76). With a larger dictionary, our method would be able to capture more details, such as drum fills, so we expect a similar system with larger dictionary to perform better.

3.3 Polyphonic Music

For the *minus-one* recording, the ENST-Drums database provides additional non-percussive accompaniment, which allows us to test our system on polyphonic music.

3.3.1 Training and Test Data

As in the previous experiment, we use the pre-trained model from all the synthesised drum data from the experiment described in Section 3.1. The test data consists of the *minus-one* recording considered in the previous experiment. We add the polyphonic accompaniment at different levels: 0dB (fully polyphonic, no attenuation), -6dB, -12dB, -18dB, -24dB and -30dB.

3.3.2 Results

The overall F-measures obtained by the system for the various levels of attenuation are detailed in Figure 4. We provide the performance of the system on the recording with no accompaniment as a baseline (overall F-measure 0.77, as in Table 3). The system's performance on all drums decays rapidly between -24 dB and -18 dB, but then stays relatively robust for the most difficult levels considered (0dB to -18dB, overall F-measure scores of 0.48–0.58).

We compare the performance of our system to Tanghe's method once more on the reduced drum type set (bd, sd, hh). It is interesting to observe that while the F-measure on

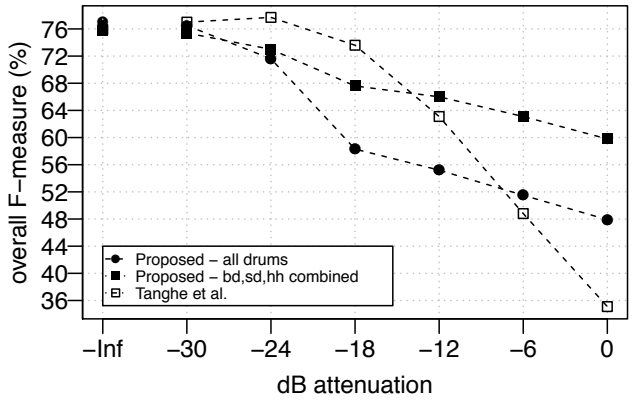

Figure 4. Overall drum events F-measure for ENST recording 116, mixed in with accompaniment at various levels of attenuation.

the pure drums is nearly the same (Tanghe: 0.76, proposed: 0.77), susceptibility to additional instruments strongly differs between the methods. The F-measure of Tanghe's method first increases for low levels of added polyphonic music (attenuation -30, -24 dB), due to the increased recall as a result of the accompaniment being detected as correct drum hits. For increasing levels of added accompaniment, performance rapidly decreases to an overall F-measure of 0.35 for 0 dB. By direct comparison, the proposed method achieves an F-measure of 0.60 even at 0 dB, demonstrating its superior robustness against high levels of accompaniment (-12, -6, 0 dB). Even for the more difficult task of recognising all 6 drum types, the proposed method (F-measure 0.48) outperforms Tanghe's.

4. DISCUSSION

Our results show not only that the proposed bar-wise drum pattern classification method is an effective, robust way to transcribe drums, but also that the first step for immediate improvement should be to increase the dictionary size in order to obtain better coverage. In addition, relaxing the strict holistic pattern approach by classifying patterns of individual instruments would allow for the recognition of combinations of patterns and hence of many new, unseen patterns. Another obvious route for improvement is to train our classifier on drum data with added polyphonic music content, which is likely to further increase robustness in polyphonic conditions.

The general approach of bar-wise drum classification is not exhausted by our particular implementation, and we expect to be able to gain further improvements by exploring different classifiers, different amounts of neighbourhood context or different basic features (e.g. non-negative matrix factorisation activations). Furthermore, to use the method in an interactive annotation system, it would be interesting to investigate bar-wise confidence scores for user guidance. Genre-specific training data could improve the performance of such systems. Finally, using more holistic features instead of single frames may also be applicable to other music informatics tasks such as chord transcription.

5. CONCLUSIONS

We have presented a novel approach to drum transcription from audio using drum pattern classification. Instead of detecting individual drums, our method first predicts whole drum patterns using an SVM classifier trained on a large collection of diverse synthetic data, and then maps the drums from the recognised patterns to the relative timestamps to achieve a transcription. The method performs very well on synthetic data, even with tempo and velocity variations on previously unseen sampled drum kits (mean pattern accuracy: 80%). Even though the pattern accuracy range differs between drum kits (50.1%–91.6%) many drum events are still classified with high precision and recall (F-measure 0.836–0.989). Unlike existing techniques, our drum detection includes open hi-hat, closed hi-hat, crash and ride cymbals, which are all reliably detected in most cases. Extending the bar patterns by one beat either side and thus obtaining overlapping patterns leads to better accuracy, mainly due to improved recognition of crash cymbals. On real drum recordings performance strongly depends on genre (F-measure for rock and disco: 0.479–0.924; hard-rock and shuffle-blues: 0.037–0.525), mainly due to the limited types of drum patterns in our current dictionary. This results in a performance slightly below that of a comparable method. However, we show that for rock music, the proposed method performs as well as the other method (F-measure: 0.77) and is substantially more robust to added polyphonic accompaniment.

6. ACKNOWLEDGEMENTS

Matthias Mauch is supported by a Royal Academy of Engineering Research Fellowship.

7. REFERENCES

[1] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic music transcription: Challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, 2013.

[2] E. Benetos, S. Ewert, and T. Weyde. Automatic transcription of pitched and unpitched sounds from polyphonic music. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014)*, May 2014.

[3] C. Cannam, C. Landone, M. B. Sandler, and J. P. Bello. The Sonic Visualiser: A visualisation platform for semantic descriptors from musical signals. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, pages 324–327, 2006.

[4] M. E. P. Davies and M. D. Plumbley. Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):1009–1020, 2007.

[5] S. Dixon, F. Gouyon, and G. Widmer. Towards characterisation of music via rhythmic patterns. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 509–516, 2004.

[6] D. P. W. Ellis and J. Arroyo. Eigenrhythms: Drum pattern basis sets for classification and generation. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 101–106, 2004.

[7] D. FitzGerald, R. Lawlor, and E. Coyle. Prior subspace analysis for drum transcription. In *Proceedings of the AES 114th International Convention*, 2003.

[8] D. Gärtner. Unsupervised Learning of the Downbeat in Drum Patterns. In *Proceedings of the AES 53rd International Conference*, pages 1–10, 2014.

[9] O. Gillet and G. Richard. ENST-Drums: An extensive audio-visual database for drum signals processing. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, pages 156–159, 2006.

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

[11] M. Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30(2):159–171, 2001.

[12] M. Mauch and S. Dixon. A corpus-based study of rhythm patterns. In *Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR 2012)*, pages 163–168, 2012.

[13] M. Mauch and M. Levy. Structural change on multiple time scales as a correlate of musical complexity. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, pages 489–494, 2011.

[14] M. Miron, M. E. P. Davies, and F. Gouyon. Improving the real-time performance of a causal audio drum transcription system. In *Proceedings of the Sound and Music Computing Conference (SMC 2013)*, pages 402–407, 2013.

[15] M. Miron, M. E. P. Davies, and Fabien Gouyon. An open-source drum transcription system for Pure Data and Max MSP. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2013)*, pages 221–225. IEEE, 2013.

[16] Y. Ni, M. McVicar, R. Santos-Rodriguez, and T. De Bie. An end-to-end machine learning system for harmonic analysis of music. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6):1771–1783, 2012.

[17] E. Pampalk, A. Flexer, and G. Widmer. Improvements of audio-based music similarity and genre classificaton. In *Proceedings of the 6th International Conference on Music Information Retrieval*, pages 634–637, 2005.

[18] J. Paulus and A. Klapuri. Drum sound detection in polyphonic music with hidden Markov models. *EURASIP Journal on Audio, Speech, and Music Processing*, 2009:14, 2009.

[19] J. K. Paulus and A. P. Klapuri. Conventional and periodic n-grams in the transcription of drum sequences. In *Proceedings of the International Conference on Multimedia and Expo (ICME 2003)*, volume 2, pages II–737. IEEE, 2003.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[21] V. Sandvold, F. Gouyon, and P. Herrera. Percussion classification in polyphonic audio recordings using localized sound models. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 537–540, 2004.

[22] J. Strong. *Drums For Dummies*. John Wiley & Sons, 2011.

[23] 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 2005)*, pages 11–15, 2005.

[24] K. Yoshii, M. Goto, and H. G. Okuno. Automatic drum sound description for real-world music using template adaptation and matching methods. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 184–191, 2004.