DRUM MUSIC TRANSCRIPTION USING PRIOR SUBSPACE ANALYSIS AND PATTERN RECOGNITION

Andrio Spich, Massimiliano Zanoni, Augusto Sarti, Stefano Tubaro

Dipartimento di Elettronica e Informazione,
Politecnico di Milano
Como, Italy
andrio.spich@mail.polimi.it
{zanoni,sarti,tubaro}@elet.polimi.it

ABSTRACT

Polyphonic music transcription has been an active field of research for several decades, with significant progress in past few years. In the specific case of automatic drum music transcription, several approaches have been proposed, some of which based on feature analysis, source separation and template matching. In this paper we propose an approach that incorporates some simple rules of music theory with the goal of improving the performance of conventional low-level drum transcription methods. In particular, we use Prior Subspace Analysis for early drum transcription, and we statistically process its output in order to recognize drum patterns and perform error correction. Experiments on polyphonic popular recordings showed that the proposed method improved the transcription accuracy of the original transcription results from 75% to over 90%.

1. INTRODUCTION

Quite a few automatic drum transcription methods have appeared in the literature in the past few years. Goto et al.[1], for example, used template matching methods for identifying drum events. Tanghe et al. [2] proposed a transcription system based on feature analysis and classification using Support Vector Machines. FitzGerald et al. [3] suggested a different approach based on source separation. This method, called Prior Subspace Analysis, uses previously computed drum subspaces to achieve source separation. All such solutions are referred to as low-level techniques [4], as they do not rely on data post-processing for error correction. On the other hand, the work of Yoshii et al. [5]; Paulus et al. [6]; and Gillet and Richard [7]; adopt a model-based approach for improving previously obtained transcription results, therefore are classified as high-level techniques. High-level transcription systems can therefore be seen as low-level transcription methods enriched with an error-correction layer. In this paper we propose a novel approach (Fig. 1) to high-level drum transcription that applies simple prior musicological knowledge to low-level drum transcription to reconstruct the rhythmic structure. Drum events, in fact, tend to occur on beat times according to specific patterns. This can be exploited to recover the tempo, to derive the measures and, finally, to reconstruct the patterns.

2. ALGORITHM DESCRIPTION

2.1. Overview

Our drum transcription method is based on a low-level transcription algorithm followed by error correction. The low-level transcription algorithm is an extension of the Prior Subspace Analysis method proposed by Barry at al. [8]. Fig. 1 shows the overall scheme of the whole processing chain. The first step after the PSA, consists of the identification of the tatum [9], which is defined as the smallest possible stepsize between two notes, through a coarse analysis of the low-level transcription. From this stepsize we then build the tatum grid, where all possible onset times for drum events are bound to lie. This guarantees that the transcription results will always be consistent with the tempo. After aligning the detected drum onsets with the tatum grid, we proceed with the identification of the bar measures. This process is based on the analysis of the drum patterns that are extracted with a given choice of measure. After identifying the measure that provides the most plausible set of patterns, we proceed with the actual error correction, which is based on the identification of a reduced set of plausible patterns that best describe the musical excerpt. Statistical pattern matching between actual patterns and those of the reduced set allows us to perform error correction.

We will see that this approach tends to improve its performance over extended excerpts. The more information we provide, in fact, the better the estimation of tatum and drum patterns that constitute the excerpt. At the end, the transcribed sequence tends to be musically self-consistent as it provides a certain degree of error concealment.

2.2. Low-level drum transcription

The Prior Subspace Analysis (PSA) [10] method is based on a separation of all drum instruments into individual audio streams, which can thus be analyzed for determining the likelihood of presence of an event on that instrument. The idea behind Subspace Analysis is that each sound source is represented by a low-dimensional subspace generated by a number of selected basis functions.

Let us consider a signal $x(n)$, whose time-frequency representation is obtained through STFT computation. Let $|X_n(k)|$ be the magnitude spectrum of $x(n)$ computed at the date n, for the frequency bin k. This spectrum can always be approximated as a linear combination of an orthogonal set of basis functions $F_i(k)$,

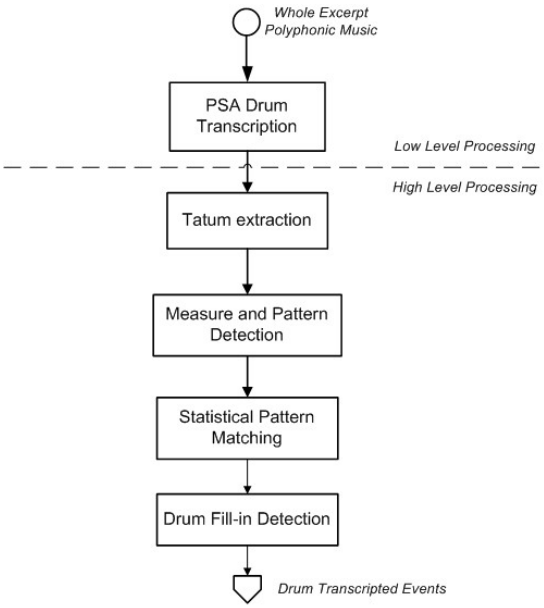

Figure 1: *Overall processing chain of the proposed method.*

$i = 1, 2, \ldots$, as

$$X_n(k) = \sum_i a_i F_i(k) \tag{1}$$

PSA attempts to extract separated subspaces assuming that there exists known prior frequency basis functions that are good initial approximations to the actual basis functions of the source of interest. Prior subspaces are built empirically by analyzing a large number of drum sounds of the actual drum instrument. In order to perform the algorithm over the whole songs, the input signal is divided in 2 seconds length blocks with an overlap of 1 window length. To obtain the time-frequency representation, the Short-Time Fourier Transform is applied to each block. In our case, we adopt a Window length of 2048 samples, an FFT size of 4096 samples and a hopsize between windows of 256 samples.

In order to improve PSA-based drum transcription, we use a spectral modulation technique [8]. Drum instruments, such as Bass drum and Snare, used in pop and rock music, are characterized by rapid broadband energy transitions followed by a fast decay. This is quite different from tonal instruments, which generally exhibit a concentration of the energy on the fundamentals frequency and on its harmonics. The spectral modulation technique that we use estimates the *percussivity* (the tendency to exhibit a broadband spectrum) of each already detected onset. Given the STFT $X_m(k)$ of the signal, we compute the log difference of the spectrogram as

$$X'_m(k) = 20 \log_{10} \frac{X_{m-1}(k)}{X_m(k)} = [X_{m-1}(k)]_{dB} - [X_m(k)]_{dB} \tag{2}$$

for all the values of m and k in the spectrogram. The measure of *percussivity* of the signal is defined as:

$$P_e(m) = \sum_k P_m(k) \tag{3}$$

where

$$P_m(k) = \begin{cases} 1 & if X'_m(k) > T \\ 0 & otherwise \end{cases}$$

T being a properly chosen threshold. P_e therefore counts how many frequency bins exceed that threshold. Applying the process to all frames we obtain a temporal profile that describes the percussion characteristics of the signal, used to improve PSA transcription.

2.3. Musicological Aspects

In order to be able to take advantage of prior musicological information for error correcting, it is important to define a minimal set of concepts that allow us to define a reference rhythmic structure.

The work that we propose is based on the analysis of drum patterns, which define the location of bass and snare drum onsets on the tatum grid.

Strictly connected to the concept of pattern is the bar (or measure) which is time interval defined as the given number of beats of a given duration. A musical excerpt is usually made of numerous bars of the same length. This inherent regularity is typical of modern music notation and is reflected in the fact that the number of beats that each bar is made of is specified at the beginning of the score by a fraction called time signature. The numerator of this fraction tells us how many beats each measure has, while the denominator expresses the duration of each beat. All such definitions are relative to the tempo, which defines the pace of the musical piece in beats-per-minute (BPM). The tempo tends to gradually change over the duration of the musical piece [11], therefore *following the tempo* means contracting or expanding the drum events according to such tempo changes. Drum event is defined as a single stroke of any drum instrument. Another important

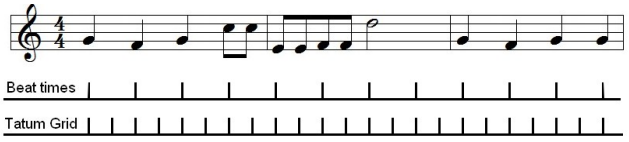

Figure 2: *Tatum grid vs. beat times in a music piece. The tatum grid represents the set of all possible note onsets.*

musicological concept useful for beats detection is the *inter-onset interval (IOI)*, which is defined as the time that elapses between consecutive onsets. This quantity does not account for the inherent duration of the events.

2.4. Tatum grid estimation

The first operation to do is to identify the beats and their pace (tempo). This is done by first determining the tatum, which is the smallest temporal quantum in the musical piece. With very few exceptions, all drum events occur at integer multiples of the tatum, therefore we can safely assume that they must lie on the tatum grid. The construction of the tatum grid is based on the extracted onsets of each drum instrument. In order to remove event outliers some preprocessing is needed. First we identify multiple events, which are defined as being less then 30ms apart from each other. Such multiple events are then reduced to single events through the removal of the weaker ones. We also assume that there is no

tempo change for the whole duration of the excerpt. The tatum is then computed from Inter-Onset Interval (IOI) of drum events that more likely to occur. In order to better define stronger onset events a noise reduction step is performed considering, for our process, only those with the amplitude higher than the average amplitude of the excerpt.

The tatum is estimated in an iterative fashion. We first select the tatum candidates by constructing a histogram of IOIs, as described in [12], from the whole set of drum onsets (all instruments together). Peaks are detected and only those whose amplitude is lower than 10% of the highest one are retrieved. For each tatum candidate, a candidate tatum grid is estimated. Inter-onset events that falls out of the candidate tatum time, considering the detected actual onset times, are removed. Gaps are then filled according the actual tatum grid. Once all the gaps are filled, the candidate tatum grid is aligned and evaluated comparing with Low-level drum onsets. The candidate with highest matching rate is selected as the final tatum grid.

2.5. Signature selection based on pattern analysis

We need constraints on the patterns to consider. Knowing the signature sets, such constraints are determined in a natural fashion and this reduces the numerosity of the sample space of all possible patterns. We determine the signature in a simplified fashion, not as an estimation process, but as a selection among possible plausible signatures. Restricting our focus to western popular music, allows us to do so.

The first step is to align the detected onsets with the tatum grid. From here, we can proceed with the signature selection based on an analysis of the patterns that are generated by different choices of the measure length.

As the measure depends on the music meter and the time-signature, we will assume that the all signatures of interest are either 2/4 or 3/4 or multiples thereof. In addition to this, we will assume that patterns tend to repeat over and over (up to small differences) and we will exploit that in our analysis. These are strong assumption but they covers a wide range of situations that tend to occur in occidental pop music. Generalizations to more complex rhythmic patterns can be achieved through modest complications of the system here described.

The key point of our analysis is that our best guess of the measure will correspond to the one that generate patterns with maximum number of repetitions and correlation score (self-similarity).

As all drum events that make a pattern are forced to be aligned with the tatum grid, we can encode the onsets as in Table 1. The timing of the events on the grid is also described in Table 2, which shows the pattern on the tatum grid.

Code	Meaning
0	No drum present
1	Bass Drum
2	Snare Drum
3	Bass + Snare Drum

Table 1: *Alphabet created for Drum grid representation.*

At this point, we can construct the *Drum Grid Matrix*, which describes the sequence of events (encoded as in Table 1) on the drum grid, organized in rows, whose length is decided depending on the measure.

Bass Drum	0.31				1.55	1.86		
Snare Drum			0.93		1.55		2.17	2.48
Tatum Grid	0.31	0.62	0.93	1.24	1.55	1.85	2.17	2.48
Resulting pattern	**1**	**0**	**2**	**0**	**1**	**1**	**2**	**2**

Table 2: *Pattern Representation.*

Given a choice of key signature, we will obtain a matrix structure, each row of which tell us which event is possibly present at each tatum grid point. For example, if the tatum is 1/64 and we are trying a key signature of 3/4, each row of the drum grid matrix will contain 48 numbers that tell us what is going on in each tatum grid point of that measure.

Once we have this matrix, we can analyze all patterns (rows) and count repetitions of each one of them. We will end up with the new description

$$\{P_1, r_1; P_2, r_2; \ldots\}$$

where P_i form the minimal set of different patterns, and r_i counts how many times the pattern P_i repeats in the excerpt.

Given G the set of patterns and G_u the set of unique patterns (not considering their repetition) in the excerpt and $\#G$ and $\#G_u$ their cardinality (number of rows), for each Pattern P_i in G_u the *number of repetition* r_i is calculated. The *correlation score* between patterns P_i and P_j is defined as follow:

$$c(P_i, P_j) = \frac{1}{d(P_i, P_j)} \tag{4}$$

where $d(P_i, P_j)$ is the *distance function* defined as follow:

$$\begin{cases} d(P_i, P_j) = \sum_{h=1}^{N} |P_i(h) - P_j(h)| \leftrightarrows \begin{cases} P_i(h) > 0 \; and \; P_j(h) = 0 \\ P_i(h) = 0 \; and \; P_j(h) > 0 \end{cases} \\ d(P_i, P_j) = 1 \leftrightarrows P_i(h) > 0 \; and \; P_j(h) > 0 \end{cases} \tag{5}$$

$P_i(h)$ and $P_j(h)$ are respectively drum events in column h of the matrix. If $P_i(h)$ and $P_j(h)$ are both greater than zero means that the drum event is present in both pattern and the correlation is 1. The *Key Signature Score* K is defined as:

$$K = \frac{\#(G)}{\#(G_u)} \cdot \max\left(r_i \cdot c(P_i, P_j)\right) \tag{6}$$

for each P_i, P_j in G_u.

After all key signature candidates are process the chosen meter is the one with highest Key Signature Score K.

2.6. Error Correction through Statistical Pattern Matching

The error correction technique that we propose is also based on the idea that a limited number of patterns are repeated within an excerpt. Therefore, it is possible to identify a unique set of *valid patterns*, which is the set of unique ones played in the song. The correction is performed through a distance score evaluation, replacing patterns of the low-level drum transcription output with the most similar pattern in the valid set. Drum fill-ins are treated separately.

The G_u, set of unique patterns, found in previous step, could represents the set of *valid patterns* in the excerpt. However, the low-level process could introduce negative or positive false and it

would induce the existence of patterns not present in the excerpt. Therefore, a subset V, which satisfy the relation eq. (7) is extracted from G_u.

$$\begin{cases} r_i \cdot c(P_i, P_j) > \max\left(r_i \cdot c(P_i, P_j)\right) \cdot T \\ r_i > 1 \\ T > 0 \; and \; T <= 1 \end{cases} \tag{7}$$

for each P_i and P_j in G_u and T being a properly chosen threshold.

Once the *valid patterns set* is obtained, all patterns output from low-level process are replaced with one in V. This is done by estimating a similarity function s between a give pattern P_i in G and each pattern P_j in V. In this step the similarity has to consider not only as the difference between amplitudes, but also considering the nature of the event; which instrument is playing. For that reason, s is defined based on the distance function $d(P_i, P_j)$, such as defined in eq. (5), and on a our extension of Hamming distance, such as defined in eq. (8). A specific case of drum pattern is drum silent measure which is often present in classic popular/rock music and characterized by the assence of percussive sound. In order to better estimate s, an empty pattern (all 0) is inserted in V.

$$\begin{cases} 1 & if P_i(k) \neq P_j(k) \; and \; P_i(k) \neq 3 \; and \; P_j(k) \neq 3 \\ 0.5 & if [P_i(k) = 3 \; and \; P_j(k) > 0] \; or \; [P_j(k) = 3 \; and \; P_i(k) > 0] \\ 0 & if P_i(k) = P_j(k) \end{cases} \tag{8}$$

where $P_i(k)$ $P_j(k)$ are respectively drum events in column k of the matrix. Given a pattern P_i in G, the similarity s, defined in eq. (10), is estimated.

$$s(P_i, P_j) = d(P_i, P_j) \cdot h(P_i, P_j) \tag{9}$$

for each P_j in V. P_i will be replaced with the pattern P_j which minimize the similarity function s. Once the correct drum event matrix is obtained, we can use it to build a pentagram score notation or to generate MIDI events in order to re-play the drum track, or even for a de-mixing process in order to extract the drum track from the polyphonic mix.

2.6.1. Drum Fill-ins

In western music, fill-ins are little variations of a basic pattern, which are generally used to emphasize transitions. The drum pattern matching algorithm here described, processes drum onsets that are aligned on the tatum grid in order to apply the most frequent drum patterns for error-correction purposes. However, some particular drum events may occur out of tatum grid resolution. This is the case for drum fill-ins that are generally characterized by the presence of rapid drum sequences, usually before the measure lines. In order to deal with drum fill-in sections, a simple solution is proposed. For each measure, the last beats are checked in order to look for *out-of-tatum* drum events, which, after an amplitude threshold check, are added to the drum event time line. Our solution does not solve the problem of looking for drum fill-ins that involve a whole measure.

3. PERFORMANCE ANALYSIS

3.1. Test Data

The proposed work was implemented and tested in MATLAB environment. The system was tested over our database consisting of 20 whole pop/rock songs. Songs presents different levels of repeating patterns, silence measures, fill-ins and different recording techniques for the drum set and they have been chosen to cover as wide a range of styles from the sixties till up to date excerpts, from pop to folk and rock sub-genres, with 4/4 or 6/8 as time signature. For comparison purposes to original PSA algorithm, the PSA results, using parameters used in [10], were also collected.

3.2. Evaluation and Results

The system performance is evaluated considering true positive a drum event which is correctly detected and correctly transcribed: the system correctly detects an onset and the drum instrument is playing. We considered as errors both false negative and false positive. Among many evaluation measurements used in the information retrieval field, the F-measure was chosen in order to provide evaluation results. The measures used are defined in eq. (10) (as in [5]).

$$\begin{cases} r_p = \frac{N_{co}}{N_{ao}} \\ r_r = \frac{N_{co}}{N_{do}} \\ d_{FM} = \frac{r_p \cdot r_r}{r_p + r_r} \end{cases} \tag{10}$$

where r_p is the *precision rate*, r_r is the *recall rate*, N_{co} is the number of correctly detected onsets, N_{ao} is the number of actual onsets, N_{do} is the number of detected onsets d_{FM} if the F-measure value.

The evaluation results are collected in Table 3, which shows the F-Measure results for bass and snare drums, and the overall result. For comparison purpose in Table 3 is also shown PSA overall evaluation results. From the 20 excerpts evaluated, only in three cases did the PSA algorithm performs better than the proposed approach. In these three cases, the main reason for poor performance was the lack of presence of drum patterns found in the PSA transcription step, inducing the proposed method to recognize wrong drum patterns. Also from the results in the Table 3 we can deduce the advances in using an high-level error-correction over an only low-level system based. We achieved good performance for both bass drum and snare drum, and the overall performance of the system improved from 75% to 92,2%.

	Proposed Approach		**Original PSA**	
	Bass	*Snare*	*Bass*	*Snare*
Precision	93.8%	92.9%	93.7%	91.6%
Recall	88.6%	85.3%	60.7%	66%
F-Measure	91.1%	89%	73.7%	76.7%
Overall	**92.2%**		**75%**	

Table 3: *The proposed High-level approach results and the original Low-level PSA results.*

One important aspect of the proposed approach is that, being a high-level technique, does not depend on the choice of PSA and can be applied over any other low-level transcription technique. This makes the approach more flexible and allows extension on further works.

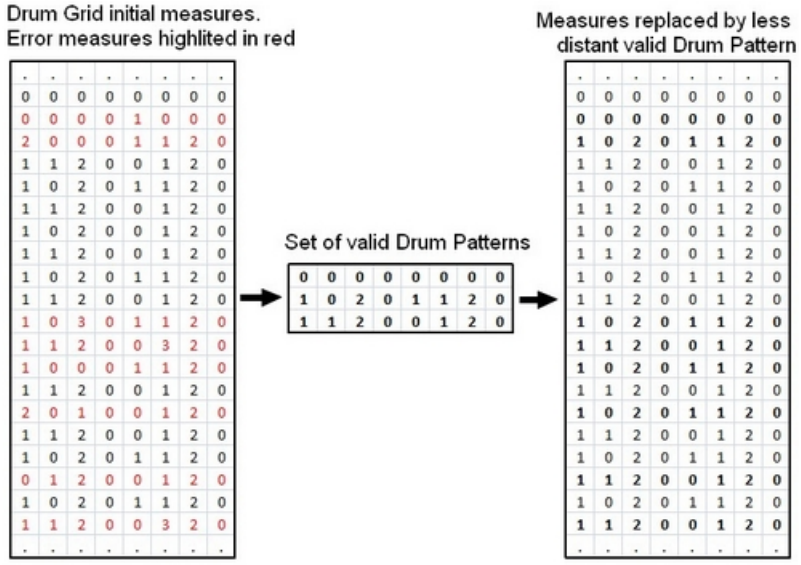

Figure 3: *The Drum Instrument Grid and the Pattern Matching Algorithm. The final transcription is composed only by the set of detected drum patterns.*

4. CONCLUSIONS

The proposed work represents an advance in addressing the problem of polyphonic percussion transcription. Based on musicological concepts, such as tatum grid, measure estimation and pattern approach, extracted analyzing the whole excerpt, the novel approach offers an alternative paradigm in the high-level methodologies scenario. The approach turned out to be very effective and the test, performed over drum events of 20 popular recordings, showed that the proposed method improved the transcription results up to 20% over the low-level approach that was chosen for comparison. The system was designed to work on a limited subset of percussion instruments and key signatures. An obvious direction for future works is the extension of the methods proposed to deal with increased numbers of different types of percussion instruments, such as Hi Hat and Tom Tom, and with a wide set of key signatures. It could also be interesting to consider a different technique for retrieving the drum patterns or consider the use of a database of pre-defined patterns for this purpose.

5. REFERENCES

[1] M. Goto and Y. Muraoka, "A sound source separation system for percussion instruments," in *The Transactions of the Institute of Electronics, Information and Communication Engineers*, 1994, vol. D-II, J//-D-II, pp. 901–911.

[2] K. Tanghe, S. Dengroeve, and B. De Baets, "An algorithm for detecting and labeling drum events in polyphonic music," in *In First Annual Music Information Retrieval Evaluation eX-change - London, UK*, Sept 2005.

[3] D. Fitzgerald, B. Lawlor, and E. Coyle, "Drum transcription in the presence of pitched instruments using prior subspace analysis," in *Irish Signals and Systems Conference - ISSC 2003 - Limerick*, 2003.

[4] Martin Haro Berois, "Detecting and describing percussive events in polyphonic music," *Master's thesis, Universitat Pompeo Fabra, Spain*, 2008.

[5] K. Yoshii, K. Komatani, T. Ogata, M. Goto, and H. Okuno, "An error correction framework based on drum pattern periodicity for improving drum sound detection," in *In ICASSP '06*, May 2006.

[6] J. Paulus and T. Virtanen, "Drum transcription with non-negative spectrogram factorisation.," in *In 13. European Signal Processing Conference, EUSIPCO*, 2005.

[7] O. Gillet and G. Richard, "Supervised and unsupervised sequence modeling for drum transcription," in *In 8th International Conference on Music Information Retrieval,ISMIR*, 2007, vol. 3, pp. 205–208.

[8] D. Barry, D. FitzGerald, E. Coyle, and B. Lawlor, "Drum source separation usign percussive feature detection and spectral modulation," in *In ISSC 2005, Dublin*, 2005.

[9] C. Uhle and J. Herre, "Estimation of tempo, microtime and time signature from percussive music," in *In DAFX '03*, Sept 2003.

[10] Derry Fitzgerald, *Automatic Drum Transcription and Source Separation*, Ph.D. thesis, Dublin Institute of Technology, 2004.

[11] J. K. Paulus and A. P. Klapuri, "Conventional and periodic n-grams in the transcription of drum sequences," *In Proc. International Conference on Multimedia and Expo ICME '03*, vol. 2, pp. 37–40, 2003.

[12] J. Sappanen, "Tatum grid analysis of musical sugnals," *IEEE Workshop on Application of Signal Processing to Audio and Acoustics*, 2001.

[13] Andrio Spich, "Percussive music transcription based on prior subspace analysis," *Master's Thesis - Politecnico di Milano*, 2009.