DRUM TRACK TRANSCRIPTION OF POLYPHONIC MUSIC USING NOISE SUBSPACE PROJECTION

Olivier Gillet and Gaël Richard
GET / Télécom Paris
CNRS LTCI,
37 rue Dareau
75014 Paris, France
[olivier.gillet, gael.richard]@enst.fr

ABSTRACT

This paper presents a novel drum transcription system for polyphonic music. The use of a band-wise harmonic/noise decomposition allows the suppression of the deterministic part of the signal, which is mainly contributed by non-rhythmic instruments. The transcription is then performed on the residual noise signal, which contains most of the rhythmic information. This signal is segmented, and the events associated to each onset are classified by support vector machines (SVM) with probabilistic outputs. The features used for classification are directly extracted from the sub-band signals. An additional pre-processing stage in which the instances are reclassified using a localized model was also tested. This transcription method is evaluated on ten test sequences, each of them being performed by two drummers and being available with different mixing settings. The whole system achieves precision and recall rates of 84% for the bass drum and snare drum detection tasks.

Keywords: Drum transcription, Rhythm analysis, high-resolution methods

1 INTRODUCTION

Traditionally, automatic music transcription and music retrieval systems essentially focus on the transcription of pitched melodic instruments. However, rhythmic information proves to be very useful for many music information retrieval tasks, as rhythm plays a key part in modern popular music - especially dance music - and since people without musical training exhibit better skills at rhythm-related tasks (tapping, "beatboxing", identifying a tempo) than at melody-related tasks (singing, humming). Automatic transcription of drum tracks allows several specific applications, such as drum-controlled sound synthesis, rhythm-driven sound effects, musical genre identification for dance music, automatic DJing, as well as content-based indexing.

Different approaches have been proposed to solve this problem. A first possible approach, suggested in Gouyon and Herrera (2001), is to segment the signal into individual events, and to classify each event using machine learning or statistical approaches. This method was applied in Gillet and Richard (2004) to the transcription of drum loops, and was subsequently integrated in a drum sequence retrieval system in which queries are formulated with spoken onomatopoeia (Gillet and Richard, 2005a). Such an approach is well suited for monophonic signals - that is to say, signals in which no other instrument than the drum kit is played - and for rather large taxonomies. However, in the context of polyphonic music signals, these methods usually perform poorly, due to the fact that the spectral or cepstral features used for classification are largely modified by the addition of harmonic instruments. An efficient way of dealing with this problem is to perform a first recognition step with very generic models, and then to retrain an adapted model on a selected set of the recognized occurences (Sandvold et al., 2004).

Template matching and adaptation is another possible approach which was introduced in Zils et al. (2002). Occurences of a temporal "seed" template are detected in the input signal using a cross-correlation measure. An adapted template is built from these occurences, and the process is iterated. This technique was refined by Yoshii et al. (Yoshii et al., 2004), by performing the template-matching in the time-frequency domain with a complex spectral distance. This technique showed very promising results for the detection of bass drum and snare drum in polyphonic recordings, but requires the timbral characteristics of each drum instrument to be constant across the entire song.

Concurrent approaches consider drum transcription as a source separation problem. Source separation aims at extracting individual sound sources from music recordings, using information gathered from different sensors - for example the two channels of a stereo recording (Barry et al., 2004) or microphone arrays. An increasing number of works recently focused on single source audio separation (Vincent and Rodet, 2004). Source separation is traditionally performed by means of statistical methods such as Independent Component Analysis, which optimizes an independence criterion on the separated source.

Once the different sources have been separated, and once each instrument of the drum kit has been identified among the separated sources, the transcription problem is equivalent to a simple onset detection. However, it is not clear how the separation should be performed. Prior knowledge about the spectral characteristics, or statistical properties of the drum track have to be introduced. Using prior subspaces or dictionaries of spectral shapes is a possible approach which was successfully followed by FitzGerald et al. (FitzGerald et al. (2003b), FitzGerald et al. (2003a)). Another separation approach followed by Dittmar and Uhle (Uhle et al. (2003), Uhle and Dittmar (2004)) requires the identification of percussive components among the separated sources.

In this paper, we extend our previous works on drum transcription to the case of polyphonic music signals, by proposing a novel transcription system. This system is based on a separation step followed by a more traditional machine learning approach. The source separation step aims at removing the contribution of the non-rhythmic instruments, in order to get as close as possible to a monophonic transcription problem, solved by support vector machine classifiers. Contrary to traditional source separation approaches that require the number of sources to be known in advance, and the separated sources to be identified and combined, the noise subspace projection considers the entire drum track as a single source. The paper is organized as follows: section 2 presents the overall principle of the system. Section 3 and 4 respectively detail the source separation and classification stages. Model adaptation is detailed in section 5, along with another possible post-processing module taking into account the time structure of the drum patterns. Following a section 6 presenting the evaluation results, section 7 suggests some conclusions.

2 SYSTEM ARCHITECTURE

The goal of our system is to transcribe the drum track of polyphonic music signals. The information we aim at extracting from the signal is thus a sequence of $(onset, instrument)$ pairs describing its drum track, where $onset$ is the onset time, and $instrument$ the drum instrument, or the combination of instruments played at this time. In the scope of this study, two instruments are used, the bass drum and the snare drum. While the drum kits played in popular music include other percussion instruments (cymbals, hi-hats, tom-toms, cowbell...), the bass drum and snare drum sequences are often sufficient to characterize the typical drum patterns of different musical genres. Moreover, all the existing query by voice (also known as "query by beatboxing") systems - Kapur et al. (2004), Gillet and Richard (2005a), Nakano et al. (2004) - rely on such a labelling of the content.

The input signals can be either monophonic or stereophonic. In case of stereophonic signals, a simple preprocessing stage aims at building an optimal monophonic mix from the left and right channels, by maximizing an impulsivity criterion. The next stage is the decomposition of the input signal $x(t)$ into eight non-overlapping subbands $x_k(t)$, $k = 1..8$. The noise subspace projection stage extracts the stochastic part of each sub-band signal $e_k(t)$.

Then, an onset detection algorithm identifies the onset times from the sub-band signals $e_k(t)$. For each detected event, a feature vector f is computed. Two parallel support vector machines are finally classifying the onset in one of the four possible categories (no event, bass drum, snare drum, bass drum and snare drum mixture).

An optional model adaptation stage trains localized SVM classifiers from the transcription. Optional preprocessing stages can also use language-modeling, or related methods, to incorporate higher-level information about drum patterns.

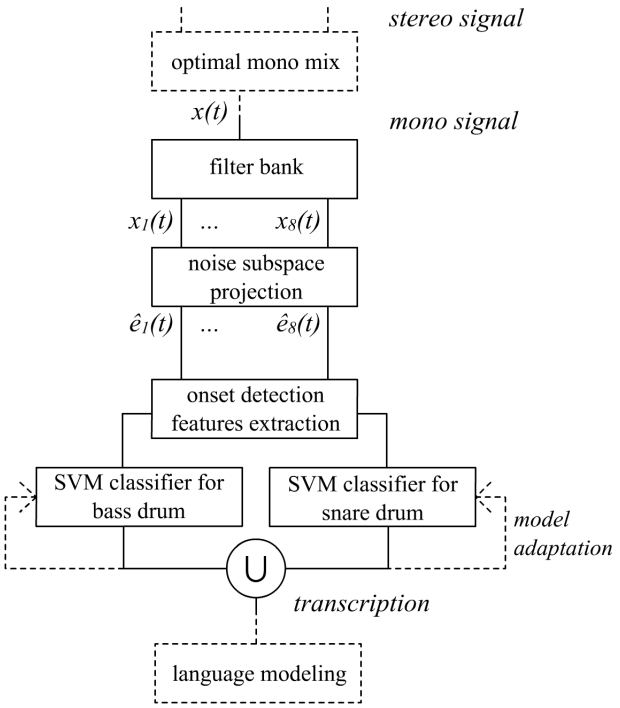

Figure 1: System architecture. Optional modules are drawn in dotted lines.

The overall architecture of the system is summarized in figure 1.

3 DRUM SIGNAL EXTRACTION

The drum signal extraction module used in this study shares common characteristics with the system described in Gillet and Richard (2005b), where it is used for a source-separation task rather than for transcription. This section summarizes its salient features.

3.1 Pre-processing of stereo signals

While a large amount of music collections (CD-audio quality or compressed music files) consist in stereo signals, most of the algorithms used for music transcription, or for the extraction of high-level descriptors such as musical genre or tempo, operate on mono signals. This can be explained by the fact that automatic music transcription or description aims at extracting high-level informa-

tion which is preserved when the music signals is reduced to a single channel from a stereo pair. However, the extra information available in an additional channel can be used to build an enhanced signal which can be optimized for a specific task.

We observed that in popular music signals, in a rather large number of cases, a monophonic mix with an enhanced percussive content could be obtained by simply mixing the left and right channels of the recording with appropriate gains. This can be explained by the fact that many popular music recordings use the so-called "panoramic" mix, in which each instrument is recorded as a single monophonic source that is mixed with two different gains on the left and right channels.

Thus, our approach consists in selecting a pair of gains for each channels, in order to maximize an impulsiveness criterion on the envelope of the remixed monophonic signal.

We tested this approach on a collection of 55 signals of popular music. In 17 cases, a source (most of the time the bass) was removed in the mono signal $x(t)$. In 3 cases, the non-rhythmic instruments were barely audible in the optimal mono signal $x(t)$.

3.2 Extraction of the stochastic component

The principle of this stage is to use a band-wise harmonic/noise decomposition to obtain the stochastic part of the signal in different frequency bands. Since drums are mixed loudly in popular music, and since unpitched percussive sounds have a very strong stochastic component, it can be seen that the stochastic part of music signals is mostly contributed by the drum sounds. As an illustration, the spectrograms in figure 2 show the similarity between the stochastic part of a snare drum + guitar mixture, and the isolated snare drum sound.

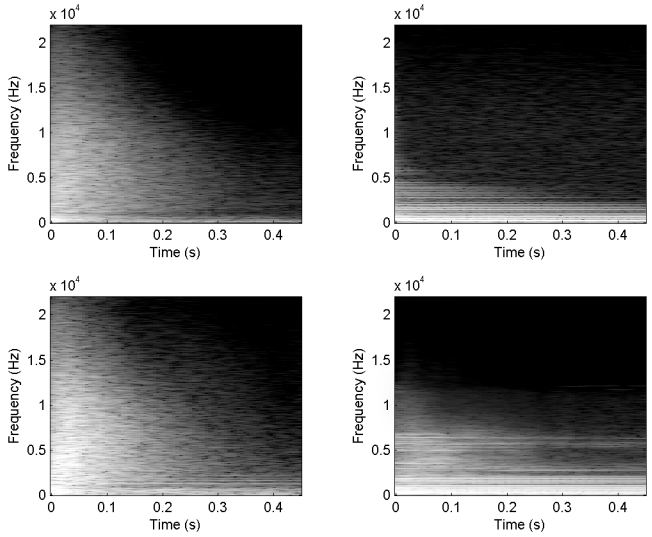

Figure 2: *Spectrograms of a snare drum and a guitar note (top). Stochastic and harmonic components of a snare drum+guitar mixture (bottom).*

An important aspect of this approach is that the estimation of the number of sources, as well as their identification is not needed.

3.2.1 Filter bank

The use of a filter bank is justified by two reasons. Firstly, the noise subspace projection performs better on narrow-band signals, in which the noise can be considered as white. Moreover, a filter bank with an octave decomposition allows the tracking of a fixed number of sinusoids per octave - a very suitable approach for mixtures of harmonic signals. Secondly, a polyphase implementation of the filter bank greatly reduces the computational cost of the noise subspace projection, by allowing the signals in each sub-band to be downsampled.

The filter bank used in our system is an octave-band (dyadic) filter bank, with $M = 8$ voices - each frequency band being one octave large. The sampling rate of the input being equal to 44100 Hz, it results in the following eight frequency bands (in Hz): [0,172], [172,345], [345,689], [689,1378], [1378,2756], [2756,5512], [5512,11025] and [11025, 22050]. The filter was implemented using a 100th order FIR filter as a prototype.

3.2.2 Noise subspace projection

The noise subspace projection stage is based on the *Exponentionally Damped Sinusoidal* (EDS) model (Badeau et al., 2002). According to this model, the signal can be decomposed in a *harmonic* part, modelled as a sum of sinusoids with an exponential decay; and a *noise* part defined as the difference between the original signal and the harmonic part.

While it is possible to estimate the sinusoids using a classical Fourier analysis, this approach suffer from the the resolution limit of the short-term Fourier transform. Subspace-based approaches, also known as high-resolution methods do not have such limitations, and are therefore used in this study. A window of length L is extracted from the original signal, defining a signal vector x. The L-dimensional space containing x is split in a $p = 2n$-dimensional space containing the signal part, and a $L-p$-dimensional space containing the noise; where n is the number of exponentionally damped sinusoids tracked. The noise vector, corresponding to the stochastic part of x can be computed by directly projecting x on the noise subspace. An entire signal can be processed using an overlap-add method.

The tracking of the signal subspace itself is achieved using the classical EVD iterative algorithm (Badeau et al., 2002), with 46ms long windows, using a 3/4 overlap.

The number of sinusoids in each frequency band was manually adjusted. Two sinusoids are used for $x_1(t)$ (lowest frequency band, in which only the bass is playing), five for $x_2(t)$, ten for $x_3(t)$ and $x_4(t)$; and eight for the other bands. Using an insufficient number of sinusoids might leave harmonic components in the output signal; while using too many sinusoids might remove all the significant timbral information from the input signal. The number of sinusoids in each band can also be automatically selected by appropriate methods (Badeau et al., 2005), at the cost of an increased computational burden.

The output of the noise subspace projection is thus 8 sub-band noise signals $e_k(t)$. Because of the multirate implementation of the filter bank, these signals need to be resynchronized in time, by upsamling them and by applying a synthesis filter.

4 ONSET CLASSIFICATION

4.1 Onset detection

In the case of drum transcription of polyphonic music, a peculiarity of the onset detection problem is that we are not interested in detecting all the onsets - only the onsets corresponding to drum events are of interest. A first possible approach is to design the onset detection module in such a way that only onsets associated to drum instruments are detected. Unfortunately, even after the noise subspace projection, the residual noise signals still contain attacks or transients from pitched instruments. Another approach is thus to handle the case of non-percussive events later in the machine learning stage.

Most of the onset detectors are based on sub-band decompositions (Klapuri, 1999). For this reason, it seems relevant to directly use the sub-band noise signal to detect onsets. Each of these sub-band noise signals is half-wave rectified and low-pass filtered, the resulting signal being noted $b_k(t)$. While the first order relative difference function $\frac{d}{dt}log(b_k(t) + A)$ is often used to detected onsets, we observed that simply using a derivative gave a higher accuracy. Thus, onsets are found by peak-picking $\frac{d}{dt}b_k(t)$.

4.2 Features extraction

For each onset localized at time t, we compute the following features over a 100ms long window starting at t:

- The energy in the first 6 sub-bands. These features can be directly computed from the decomposition.
- The average of the 12 first MFCC (without c_0) across successive frames. The MFCC are computed on the noise signal $\sum_k \hat{e}_k(t)$

The inclusion of the first MFCC coefficient c_0 gave slightly worse results. The use of the 4 spectral moments did not increase the accuracy either - it is very likely that these features are highly sensitive to the noise subspace projection.

Different transformations were tested on this feature set. Performing a Principal Component Analysis on the data set did not significantly increase the performances; however, it could be seen that the first 12 components contributed in 96 % of the total variance. Performing the classification on these 12 first principal components reduced the computational cost of the learning / classification steps, without any significant accuracy loss.

4.3 SVM classification

The classification problem presented in this work is slightly different from a more traditional "segment and classify" approach. Firstly, some of the onsets to classify are not occurences of drum instruments, and must be recognized as such and discarded. Secondly, the small number of categories used in our studies is well suited for a binary classification approach. Thus, we decided to train two classifiers, one of them detecting the presence of bass drums, and the other detecting the presence of snare drums. When the input onset does not correspond to an occurence of a percussion instrument, the pair of classifiers will output the pair (non bass drum, non snare drum).

The classifiers used are Support Vector Machines (Vapnik, 1995), which are well suitable for binary classification problems, and show very interesting generalization properties. A general-purpose kernel (radial basis function) was used. The implementation chosen was SVM^{light} (Joachims, 1999).

The output of a SVM is classically an uncalibrated value - its sign being used for the decision, and its absolute value roughly expressing the distance to the decision boundary. A method to obtain posterior probabilities from this uncalibrated value has been described in Platt (2000). The output of the SVM $f(x)$ is mapped to the interval $]0, 1[$ with a sigmoid function: $p(x) = \frac{1}{1+e^{Af(x)+B}}$. The parameters A, B are fitted using maximum likelihood estimation on a subset of the training data. Typically, a large fraction of the training set is used to perform the SVM learning, and the remaining part is used to estimate the parameters A and B.

The availability of posterior probabilities allow further post-processing stages, such as those described in the next section. Moreover, it is easier to adjust the decision threshold with scaled, probabilistic values, than with an uncalibrated output. Such adjustements are necessary if the users of the transcription system need to adapt the ratio of "miss" and "false alarm" errors to their own specific applications.

5 POST-PROCESSING STAGES

5.1 Adaptation

We decided to follow an approach similar to the one described in Sandvold et al. (2004). This approach consists in performing a first recognition step using a general model - in our case this general model consists in the SVM classifiers presented in the previous section, the parameters of which have been learned on the whole training set. Then, the N recognized instances are ranked using a likelihood measure, and a subset of them (containing kN examples) from which the best recognition scores are achieved is selected. In our case, we used the probabilistic output of the SVM classifier as a likelihood measure, instead of manually ranking the recognized instances as it was done in the work of Sandvold et al. A "localized" or adapted model is subsequently learned on this small training set. The recognition is finally performed again on the whole sequence, this time using the excerpt-specific, localized model.

Different values have been tested for the value of k, the best results being achieved with $k = 0.4$ (40% of the recognized instances are used to retrain the system).

5.2 Periodic decisions

Different language-modeling techniques have been proposed to incorporate high-level information into drum transcription systems. Short-term models, such as n-grams (Gillet and Richard, 2004) usually model the time-dependencies in acoustic features caused by overlapping strokes. It also models simple stereotypical patterns, such as tom fills. In the context of our study, in which only two categories of instruments are used, such a model is not particularly useful. In fact, the different n-sequences of bass-drum and snare drums are almost equiprobable in our database. A similar problem occured with periodic n-grams (Paulus and Klapuri, 2003): as our database covers different styles, the different sequences were also almost equiprobable.

We finally decided to follow a different approach, which requires no prior training, and is only based on the repetitive nature of drum patterns. In order to classify an event occuring at time t, we fuse the classification results for the events occuring at time t, $t-M$ and $t+M$, where M is the duration of a bar or pattern. M can be automatically estimated from the audio signal (Klapuri, 2003), or from the symbolic transcription obtained at the previous stage (Meudic and St-James). To evaluate our method independently of pattern duration estimation errors, the pattern duration was manually annotated for each file. Different operators were tested for the fusion, such as weighted means, products, median, and the Yager t-norm (For a review of different aggregation operators, see Detyniecki (2001)).

6 EVALUATION AND RESULTS

6.1 Database

In order to avoid the tedious manual annotation of pre-existing material, and to enable a wide range of experiments, we recorded our own database. This database makes use of "training sessions", also known as "minus one" CDs. Such CDs are used for the teaching of drumming, and allow students to practice on the top of a music accompaniment from which the drum track has been removed. We selected ten excerpts from two "minus one" CDs. The excerpts are one minute long, cover various styles (blues, twist, metal, funk, celtic...) and are mostly played by acoustic instruments (bass, electric guitar, sax, accordion...) with a few synthetic keyboards (FM electric piano, organ).

Two professional drummers were asked to play a rhythmic accompaniment on the top of the excerpts, which were played through headphones. Each drummer brought his own drum kit. Inter-sequence variability was introduced by the use of different kinds of sticks (including bundled sticks) and by asking the drummers to adjust their playing style according to the genre of each sequence. Both drummers played in a rather nuanced style, which introduced intra-sequence variability - a characteristic not present in databases using synthetic or sampled drum sounds. The performances were recorded with 8 microphones (A Beyer 88 for the bass drum, a Shure SM57 for the snare drum, a Schoeps CMC with cardiod capsule for the hi-hat, two Shure SM58 for the highest tom-toms, a Sennheiser 441 for the low tom and two Audio-Technica AT4040 for the overheads), amplified by 4 Behringer Ultragain Pro Mic2200 dual pre-amplifiers, on a Tascam MX2424 digital multitracker (8 channels were used). A stereo mix was generated from the 8 tracks, using panning, equalization, and compression. This stereo mix and the original "minus one" excerpts were finally mixed with different relative levels. First of all, a reference mix was produced, in which the drums and other instruments were well-balanced. Then, two other mixes were produced, in which the drums were respectively amplified and attenuated by 6dB. The stereo drums mix was also kept. This results in 80 stereo different signals (10 excerpts $\times$ 2 drummers $\times$ 4 mixes).

The annotation was obtained semi-automatically, by a simple onset detection algorithm on the bass drum / snare drum tracks, the output of which was manually checked and corrected. The average number of events (on both the bass drum and snare drum tracks) per excerpt is 178.

6.2 Evaluation metric

The correctness of the transcription was evaluated by precision and recall measures. Let N_d be the total number of events detected by the system, N_c the number of correct events detected by the system; and N the actual number of events to be detected. Precision and recall are defined as:

$$\text{precision} = \frac{N_c}{N_d}$$

$$\text{recall} = \frac{N_c}{N}$$

As it is possible to adjust the decision rules to favor precision or recall, we chose a decision rule in which the two kinds of errors, "false alarms" and "misses" are roughly balanced. The f-measure, which is defined as:

$$\text{F-measure} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

is another measure of the accuracy of the system, which is independent of the chosen precision/recall tradeoff.

It is worth noting that a small deviation is allowed between the actual onset and the detected onset: events are considered as correctly detected when they are detected within 50ms of the reference onset.

6.3 Evaluation protocol

Experiment 1: Robustness In order to test the accuracy and robustness of the transcription system under different kind of mixing conditions, we repeated for each of the 4 mixes (drum only, balanced, attenuated drums, amplified drums) the following procedure:

- Train the SVM on the events detected from the 10 excerpts played by drummer A.
- Evaluate the SVM on the 10 excerpts played by drummer B.

- Repeat the process after having exchanged A and B.

The precision, recall and F-measure obtained for each drummer and excerpt are averaged. It is worth to mention that the stereo pre-processing stage was not used in this first experiment.

More generally, considering the available data, this two-folds protocol is the most adapted to show the generalization capabilities of the learning algorithms. However, it is necessary to keep in mind, while interpreting the results of our experiments, that the training set is relatively small.

Experiment 2: Performance of the stereo pre-processing The same experiment was repeated using a stereo pre-processing stage, and the results were compared.

Experiment 3: Post-processing Results for a "baseline" system are obtained using a protocol similar to the one used in experiment 1, except that only the balanced mix is used. Results are then computed with the model adaptation stage, and with the periodic decision stage.

6.4 Results and discussion

Results for the robustness experiment are given in table 1. The best scores are achieved with recordings in which the drums are mixed loudly, but acceptable results are also obtained when the drums are attenuated. With balanced mixes, which correspond to the situation encountered in real world recordings, the performances of the system are roughly comparable to those given in Yoshii et al. (2004), even though direct comparison is not possible since a different dataset was used.

Table 1: Results of the robustness experiment

Mix	Recall	Precision	F-measure
Drums -6dB	75.8%	71.1%	73.4%
Balanced mix	83.9%	84.2%	84.0%
Drums +6dB	87.4%	91.2%	89.2%
Drums only	83.7%	92.7%	88.0%

The evaluation of the stereo pre-processing stage is given in table 2. It can be seen that this stage significantly increases the accuracy of the transcription when the other instruments are mixed more loudly than the drums.

Table 2: Impact of the stereo pre-processing stage

Mix	Recall	Precision	F-measure
Drums -6dB	76.2%	78.4%	77.3%
Balanced mix	82.0%	88.5%	85.1%
Drums +6dB	84.3%	90.8%	87.4%
Drums only	83.7%	92.7%	88.0%

The different post-processing stages are compared in table 3. It can be seen that none of the methods described in section 5 improve the recognition.

Table 3: Impact of the post-processing stages

Method	Recall	Precision	F-measure
Baseline	83.9%	84.2%	84.0%
Adaptation	78.1%	71.0%	74.3%
Periodic decision	87.2%	78.4%	82.6%

Different reasons can explain the failure of the localized models. First of all, the local models are trained using only a subset of the detected onsets. This results in a very small training set. Increasing the fraction of recognized instances used to train the local model does not help either, since it becomes more and more likely that some of these instances are indeed misclassified. Secondly, we noticed that our features set, computed on the residual noise signal, did not exhibit a lot of variability from one track to another, contrary to the feature set used in Sandvold et al. (2004) which was computed on the original signal, rather than on a residual noise signal. Finally, we noticed that the selected instances were mostly loud or solo strokes, most of them played off-beat. It means that the adapted model will specialize itself in identifying such strokes, and will become unable to identify strokes with different timbral characteristics or dynamics appearing within the same track. It seems that the use of localized models would give best results with synthetic or sampled drum tracks, in which there is very little variation between the different snare drum or bass drum sounds.

The use of the modified decision function taking into account the periodicity of drum patterns does not increase the classification results either. However, a thorough analysis of the classification errors shows that this method modifies the kind of errors made by the system. Classification errors on the steady, typical component of the drum pattern are less frequent, while many recognition errors are localized in breaks or in variations at the end of a pattern. It is not clear which one of these two classes of errors is more acceptable. Applications such as automatic accompaniement generation, or score following would probably require better classification results of loud strokes, played on strong beats. On the other hand, playing style analysis probably requires a very accurate transcription of breaks, soli and variations. This suggests the use of a problem-specific evaluation metric with a different cost for classification errors occuring on strong beats / loud strokes; and the rest of the sequence.

Finally, the detailed results of the baseline system are given in table 4. Drummer 2 has a very nuanced style, with a lot of variations in the dynamic of the strokes, while Drummer 1 has an energic, steady, style with fewer variations in the dynamic of strokes. It can be seen that the algorithm performs better when trained on Drummer 2 and evaluated on Drummer 1. Thus, for large scale applications, our system will need to be trained on a larger database containing multiple variations in timbre and dynamics. The two sequences on which the algorithm gives the worst results, *Groove 5/4* and *Celtic* are played by both drummers with a lot of ghost notes - quiet beats which help the drummer in keeping the tempo more accurately. While we annotated ghost notes and included them in the

Table 4: Detailed results of the baseline system

		Bass drum			Snare drum		
Sequence	**Drummer**	**Rec.**	**Prec.**	**F-meas.**	**Rec.**	**Prec.**	**F-meas.**
Blues	1	72.1%	96.1%	0.82	95.6%	100.0%	0.98
	2	86.1%	82.7%	0.84	87.8%	87.8%	0.88
Blues rock	1	92.2%	92.2%	0.92	100.0%	100.0%	1.00
	2	89.5%	91.7%	0.91	82.2%	80.4%	0.81
Celtic	1	75.4%	94.9%	0.84	77.8%	33.3%	0.47
	2	70.1%	87.8%	0.78	80.3%	68.1%	0.74
Funk	1	79.8%	62.5%	0.70	87.8%	97.0%	0.92
	2	77.6%	90.6%	0.84	81.5%	91.7%	0.86
Jazz funk	1	94.7%	87.3%	0.91	97.9%	85.2%	0.91
	2	78.6%	94.2%	0.86	84.6%	76.7%	0.80
Groove 5/4	1	85.2%	54.1%	0.66	82.8%	96.0%	0.89
	2	91.9%	77.5%	0.84	86.3%	62.9%	0.72
Metal	1	90.3%	77.8%	0.84	83.3%	88.7%	0.86
	2	75.5%	72.1%	0.74	77.9%	75.9%	0.77
Rock	1	90.5%	77.9%	0.84	88.5%	97.7%	0.93
	2	74.1%	88.6%	0.81	88.0%	89.0%	0.88
Shuffle	1	74.4%	81.5%	0.78	85.7%	85.7%	0.86
	2	67.6%	93.1%	0.78	68.9%	81.6%	0.75
Twist	1	97.6%	75.0%	0.85	91.9%	95.8%	0.94
	2	84.8%	98.1%	0.91	78.9%	98.1%	0.87

evaluation of this work, it is not clear if errors on such strokes are acceptable or not, and if they should be taken into account.

7 CONCLUSION AND FUTURE WORK

This paper proposed a novel drum transcription system for polyphonic music and evaluated its performances, as well as the effect of different pre-processing and post-processing stages. Promising results (83.9 % recall, 84.2 % precision) were obtained on a test database under standard recording conditions. However, the failure of the different post-processing stages tested raises interesting questions.

Firstly, the different kind of errors produced with or without language modeling suggests that evaluation metrics could take into account the position and importance of the misdetected events within the rhythmic patterns. The different tasks and applications for which drum transcription is needed should be clearly identified, and task-specific evaluation metrics should be devised for each of them.

Secondly, our results show that the use of localized models not taking into account all the information learned in the generic model is not the best way to perform adaptation. Further works will focus on the use of incremental learning methods for support vector machines, in which the original generic model is updated or transformed, rather than discarded.

It is also planned to improve the noise subspace projection stage by automatically selecting the number of sinusoids in each frequency band. Finally, our system will be trained and tested on a larger corpus, which will include labels for other categories of drum instruments such as hi-hats or cymbals. Such a larger corpus could be created by mixing pre-recorded drum loops, from which an annotation is available, with the "minus one" sequences. Our present corpus could then be used as a testing set.

ACKNOWLEDGEMENTS

The authors would like to thank Frederic Rottier and Bertrand Clouard who performed the drum sequences used in this study, as well as the sound engineer Michel Desnoues for the high quality recordings.

This work was partly supported by the MusicDiscover project of the ACI Masse de données.

REFERENCES

R. Badeau, R. Boyer, and B. David. Eds parametric modeling and tracking of audio signals. In *Proceedings of 5th International Conference on Digital Audio Effects (DAFX'02)*, September 2002.

R. Badeau, B. David, and G. Richard. Selecting the modeling order for the esprit high resolution method: an alternative approach. In *Proceedings of the 2004 International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)*, May 2005.

D. Barry, B. Lawlor, and E. Coyle. Sound source separation: Azimuth discrimination and resynthesis. In *Proceedings of the 7th International Conference on Digital Audio Effects (DAFX'04)*, October 2004.

M. Detyniecki. Numerical aggregation operators: State of the art. In *International Summer School on Aggregation Operators and their Applications*, 2001.

D. FitzGerald, B. Lawlor, and E. Coyle. Drum transcription in the presence of pitched instuments using prior subspace analysis. In *Proceedings of the Irish Signals and Systems Conference (ISSC 2003)*, July 2003a.

D. FitzGerald, B. Lawlor, and E. Coyle. Prior subspace analysis for drum transcription. In *Proceedings of the 114th AES Convention*, March 2003b.

O. Gillet and G. Richard. Automatic transcription of drum loops. In *Proceedings of the 2004 International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)*, May 2004.

O. Gillet and G. Richard. Drum loops retrieval from spoken queries. In *Journal of Intelligent Information Systems*, volume 24:2/3, pages 159–177. Springer Science, 2005a.

O. Gillet and G. Richard. Extraction and remixing of drum tracks from polyphonic music signals. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'05)*, October 2005b.

F. Gouyon and P. Herrera. Exploration of techniques for automatic labeling of audio drum tracks. In *Proceedings of MOSART: Workshop on Current Directions in Computer Music*, November 2001.

T. Joachims. Making large-scale svm learning practical. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.

A. Kapur, M. Benning, and G. Tzanetakis. Query by beatboxing: Music information retrieval for the dj. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, October 2004.

A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *Proceedings of the 1999 International Conference on Acoustics, Speech, and Signal Processing (ICASSP'99)*, March 1999.

A. Klapuri. musical meter estimation and music transcription. In *Proceedings of the Cambridge Music Processing Colloquium*, March 2003.

B. Meudic and E. St-James. Automatic extraction of approximate repetitions in polyphonic midi files based on perceptive criteria. In *Lecture notes in Computer science, LNCS 2771*. Springer Verlag.

T. Nakano, J. Ogata, M. Goto, and Y. Hiraga. A drum pattern retrieval method by voice percussion. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, October 2004.

J. Paulus and A. Klapuri. Conventional and periodic n-grams in the transcription of drum sequences. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME'03)*, July 2003.

J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classiers*, pages 61–74, 2000.

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)*, October 2004.

C. Uhle and C. Dittmar. Further steps towards drum transcription of polyphonic music. In *Proceedings of the 116th AES convention*, May 2004.

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 (ICA2003)*, April 2003.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

E. Vincent and X. Rodet. Underdetermined source separation with structured source priors. In *Proceedings of the 5th Symposium on Independent Component Analysis and Blind Signal Separation (ICA2004)*, April 2004.

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)*, October 2004.

A. Zils, F. Pachet, O. Delerue, and F. Gouyon. Automatic extraction of drum tracks from polyphonic music signals. In *Proceedings of WEDELMUSIC2002*, December 2002.