Automatic Labelling of Tabla Signals

Olivier K. GILLET
GET-ENST (TELECOM Paris)
46, rue Barrault
75013 Paris, France
olivier.gillet@enst.fr

Gaël RICHARD
GET-ENST (TELECOM Paris)
46, rue Barrault
75013 Paris, France
gael.richard@enst.fr

Abstract

Most of the recent developments in the field of music indexing and music information retrieval are focused on western music. In this paper, we present an automatic music transcription system dedicated to Tabla - a North Indian percussion instrument. Our approach is based on three main steps: firstly, the audio signal is segmented in adjacent segments where each segment represents a single stroke. Secondly, rhythmic information such as relative durations are calculated using beat detection techniques. Finally, the transcription (recognition of the strokes) is performed by means of a statistical model based on Hidden Markov Model (HMM). The structure of this model is designed in order to represent the time dependencies between successives strokes and to take into account the specificities of the tabla score notation (transcription symbols may be context dependent). Realtime transcription of Tabla soli (or performances) with an error rate of 6.5% is made possible with this transcriber. The transcription system, along with some additional features such as sound synthesis or phrase correction, are integrated in a user-friendly environment called **Tablascope**.

1 Introduction

Due to the exponential growth of available digital information, there is a need for techniques that would make this information more readily accessible to the user. As a consequence, automatic indexing and retrieval of information based on content is becoming more and more important and represent very challenging research areas. Automatic indexing of digital information permits to extract a textual description of this information (i.e. meta data). In the context of music signals, if such a description would ultimately be a complete transcription (for example in the form of music scores), it should at least target to extract two types of descriptors from the audio signal:

- a set of general descriptors such as genre, style, musicians and instruments of a music piece.
- a set of more specific descriptors such as beat, notes, chords or nuances.

Automatic labelling of instrument sounds therefore represents one of the component of a complete indexing system. Previous efforts in automatic labelling of instruments sounds have mainly been dedicated to sounds with definite pitch ([Kaminskyj, 2001], [Martin, 1999] or [Herrera et al., 2000] for a critical review). Percussive sounds is a specific class of instrument sounds. They can be pitched (as for xylophone or marimba) or unpitched (as for drum or congas). There is much interest in devoting more effort to the latter class since there is a number of specific applications for this class of sounds (automatic drum loop recognition and retrieval, virtual dancer, drum-controlled synthesizers . . .).

In fact, more attention is now given to the class of unpitched percussive sounds (see for example [Herrera et al., 2003],[McDonald and Tsang, 1997], [Gouyon and Herrera, 2001])), but these efforts are still limited to western music and for most studies only isolated sounds are considered.

We propose in this paper a study on the automatic transcription of tabla performances. The tabla is a percussion instrument widely used in (North) Indian classical and semi-classical music, and which has gained more and more recognition among *fusion* musicians. As for a number of percussive instruments (such as congas, drum), the characteristics of the audio signal produced is rather impulsive and successive events are therefore rather easily detected. Another advantage for automatic analysis is that such instruments produce only limited sound mixtures (1 or 2 events at the same time where in polyphonic music over 50 notes can be played simultaneously). However, the transcription of tabla signals is quite challenging because there is strong time dependencies that are essential to take into account for a successful transcription. There are two types of time dependencies : firstly, the instrument can produce long, resonant strokes which will overlap and alter the timbre of the following strokes. Secondly, the symbol used to represent a stroke can depend on the context in which it is used. Prior works related to computerization of the Tabla include a phrase retrieval system using MIDI input [Roh and Wilcox, 1995] or specific controller and synthesis models [A. Kapur and Cook, 2002], but to our knowledge no transcription system of tabla performances have been reported.

Figure 1: The tabla is a pair of drums : a metallic bass drum (the *bayan*) and a wooden treble drum : the *dayan*

The paper is organized as follows. The next section presents the instrument and the specific transcription that is commonly used to describe tabla performances. Section 3 is dedicated to the description of the transcription system including a detailed section on the model proposed. Section 4 presents some evaluation results on real tabla performances. Finally, section 5 suggests some conclusions.

2 Presentation of the tabla

The tabla (see figure 1) is a pair of drums traditionally used for the accompaniment of Indian classical or semi-classical music. The *bayan* is the metallic bass drum, played by the left hand. It is used to produce a loud resonant sound or a damped non-resonant one. By sliding the palm of the hand on the drum head while striking, the perceived pitch can be modified. The *dayan* is the wooden treble drum, played by the right hand. A larger variety of sounds is produced on this drum. As for timbals for example, the drum can be tuned according to the accompanied voice or instrument. It is also important to note that for this class of instruments the main resonance (that corresponds to the main perceived pitch) is much above the resonance of the other harmonics. Further elements on the physics of the Tabla can be found in [Fletcher and Rossing, 1998] or in the early work of Raman [Raman, 1934].

A mnemonic syllable or *bol* is associated to each of these strokes. Since the musical tradition in India is mostly oral, compositions are often transmitted from masters to students as sung *bols*. Common *bols* are : **Ge**, **Ke** (*bayan bols*), **Na**, **Tin**, **Tun**, **Ti**, **Te** (*dayan bols*). Strokes on the *Bayan* and *dayan* can be combined, like in the *bols* : **Dha** (**Na + Ge**), Dhin (**Tin + Ge**), Dhun (**Tun + Ge**)... Because of these characteristics, even if two drums are played simultaneously, the transcription can always be considered as monophonic - a single symbol is used even if the corresponding stroke is compound.

A specificity of this notation system is that two different *bols* can sound very similar. For example, Ti and Te nearly sound the same, but are played with different fingers. Though, it is possible to figure out which *bol* is used, since a fast sequence of these strokes is played by alternating between Ti and Te. Another characteristic is the existence of "words" made of several *bols* concatenated in stereotyped groups which are used as a whole, e.g. **TiReKiTe** or **GeReNaGe**. Such words are used as compositional units, and make the recitation and memorization of a piece easier.

A tabla sequence or rhythm is composed of successive *bols* of different durations (note, half-note, quarter note,...) however the rhythmic structure can be quite complex. The basic rhythmic structures (called *taal*) can have a large variety of beats (for example 6,7,8, 10, 12, 16,..) which are grouped in measures. The successives measures may have different number of beats (for example **Dhin Na** | **Dhin Dhin Na** | **Tin Na** | **Dhin Dhin Na** where the vertical bars symbolise the measure boundaries).

The grouping characteristics are in a sense similar to those observed in spoken or written languages where "words" can be grouped in sentences. These properties can be modelled by means of a *Language model* that will provide an estimation of the probability of a given sequence of words. For the tabla, since the association between a sound and the corresponding *bol* can be context-sensitive (i.e. depends on previous *bols*), it seems quite appropriate to use language modelling approaches to model these dependencies.

In this work, the transcription is limited to the recognition of the successives *bols* with their corresponding relative duration (note, half-note etc...) but does not intend to extract the complex rhythmic structure (i.e the measures).

3 Transcription of Tabla phrases

3.1 Architecture of the system

The architecture of our system is organized in several modules:

- **Parametric representation** the signal parametric representation is obtained by means of two main sub-modules: the *onset detection* which permits to segment the signal into individual strokes and the *features extraction* which provides an acoustic vector from the segment previously extracted.
- **Sequence Modelling:** the sequence of feature vectors (one vector per stroke) is then modelled by a classification technique such as k nearest neighbors (k-NN) or Hidden Markov Models (HMM).
- **Transcription:** The final transcription is given in the form of a succession of *bols* or words accompanied with a rhythmic notation obtained from the tempo detection module

Figure 2 provides the general architecture of the system including the transcription modules (in bold line), and the tabla sequence generator/synthesizer (in dotted line) that is further explained in section 5.2. It is important to note that the input to the system can either be an audio file (for example in `.wav` format) or audio streams that are processed in real time.

3.2 Parametric representation

3.2.1 Segmentation in strokes

As for many transcription system, the first step of our transcriber consists in performing a segmentation of the incoming stream in separate strokes. A number of approaches have been proposed in the literature for such a segmentation (see for example [Klapuri, 1999]). In the case of tabla signals, most of the strokes have a fast decaying exponential envelope. As a consequence, a simple envelope/threshold based approach seems adequate and is detailed below:

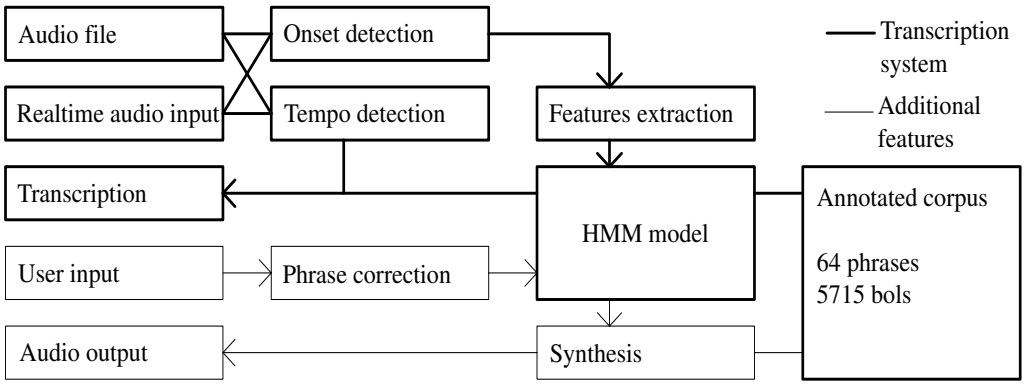

Figure 2: Architecture of the system: the transcription system is represented by bold lines and the tabla sequence generator or synthesis is represented by dotted lines

- **Sampling:** The original input signal is sampled (or resampled) at 44.1 kHz.
- **Envelope extraction:** The envelope, $env(t)$, is obtained in two steps. First, a low-frequency envelope signal (sampled at 220.5 Hz), $env1(t)$, is calculated by taking the absolute value of the maximum in 400 samples windows. Second, a local normalisation is performed by dividing the previous envelope by another rectified very-low-frequency envelope signal, $n(t)$, (sampled at 1 Hz) in order to compensate the dynamic variations of the input signal. Note that $n(t)$ is estimated by taking the absolute value of the maximum in 44100 samples windows.
- **Onset detection**: An onset is detected at locations where the difference between two successive samples of the rectified envelope signal exceeds a given threshold while another onset wasn't detected during the last 100 ms. Note that due to the normalisation approach, this threshold does not depend on the initial signal energy.

3.2.2 Duration and tempo extraction

Tempo extraction has also received much interest in the recent years (see [Scheirer, 1998] and [Alonso et al., 2003] for example). Due to the impulsiveness of the tabla signals, a simple approach is also preferred in this work. A local tempo at time T is estimated by finding a peak in the autocorrelation function of $env(t), t \in [T - 10s, T + 10s]$ in the range [60, 240] BPM (*Beats per minute*). Events are then aligned and quantized to the corresponding time grid. We keep for each stroke its relative duration as a fraction of the tempo (each events is then rhythmically described as a note, half-note, quarter noter etc....).

3.3 Features extraction

Early percussion instruments classification systems used energy in a set of well-chosen frequency bands as features. Recent systems uses more complex features sets, including temporal, spectral or cepstral features ([Herrera et al., 2003], [McDonald and Tsang, 1997], [Sillanpää et al., 2000]).

The power spectra of various *bols* are plotted in figure 3. It can be seen that the discrimination between the *bols* can be achieved according to the position, relative weight and width of the spectral peaks. It is then appropriate to consider that the power spectra of each stroke can be represented by a probability distribution and that it can be approximated by a weighted sum of N Gaussian distributions. For tabla signals, $N = 4$ represents an appropriate choice. The four Gaussian distributions are obtained as follows:

- An initial estimate is obtained by computing the empirical mean and variance in four pre-defined frequency bands $B_1[0, 150]Hz, B_2 = [150, 220]Hz, B_3 = [220, 380]Hz, B_4 = [700, 900]$. Despite that the bandwidth of these bands are sufficient to cope with significant tuning variation of the Dayan, it would be necessary to adapt this front-end to obtain a truly tuning-independent system. For example, Mel Frequency Cepstral Coeficients (MFCC) which have better generalization properties have also been tested but have lead to slightly degraded performances on our corpus.

- Then, an improved estimation is obtained by iterating the Expectation-Maximisation (EM) algorithm on a training data set. In practice, convergence is obtained in a few steps (typically in 4 iterations).

In summary, the feature vectors F are constituted with 12 elements $F = f_{1,..,12}$ (the mean, variance, and relative weight of each of the 4 Gaussians).

3.4 Learning and classification of *bols*

Due to the time dependencies discussed above, classifying each stroke independently of its context leads to unsatisfactory results. An efficient approach that integrates context (or time) dependencies is given by the Hidden Markov Model (HMM). This class of models is particularly suitable for modelling short term time-dependencies and it has been successfully used for a wide variety of problems ranging from speech recognition ([Rabiner and Juang, 1993]) to piano music transcription ([Raphael, 2002]). In such a framework, the sequence of feature vectors O_t is represented as the output of a Hidden Markov Model. The recognition is performed by searching the most likely states sequence, given the output sequence of feature vectors.

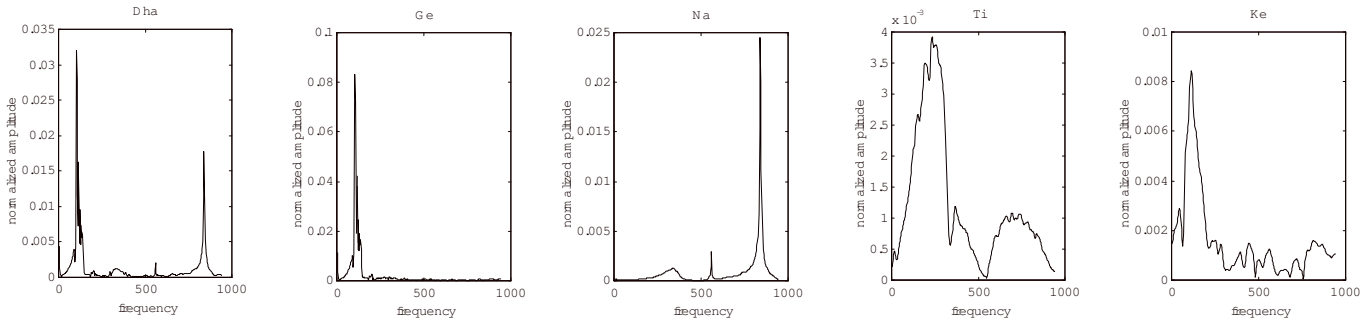

Figure 3: Spectra of common *bols* (from left to right Dha, Ge, Na, Ti, Ke).

3.4.1 Description of the model

Although many implementation of HMM are now available, it is felt important to recall the main stages of such a model and to describe how it is applied to the particular problem of tabla sequence recognition. The model includes states, transitions between states and emission probabilities.

States A couple of *bols* B_1B_2 is associated to each state q_t of the model. In other words, q_t represents "B_2 in the context of B_1 at time t". For example, for the sequence of *bols* $b = (*start*, *start*, Dha, Dhin, Dhin, Dha)$, the corresponding sequence of states is $q_0 = [*start* . * start*], q_1 = [*start * .Dha], q_2 = [Dha.Dhin], q_3 = [Dhin.Dhin], q_4 = [Dhin.Dha]$.

Transitions If the state i is labelled by B_1B_2 ; and j by B_2B_3, then the transition from state $q_{t-1} = i$ to state $q_t = j$ is given by:

$$\begin{aligned} a_{ij} &= p(q_t = j | q_{t-1} = i) \\ &= p(b_t = B_3 | b_{t-1} = B_2, b_{t-2} = B_1) \end{aligned}$$

where $p(b_t = B3)$ is the probability density of observing the *bol* B_3 at time t.

Initial probabilities Since two dummy states `*start*` are inserted at the beginning of each sequence, the first state is always $q_0 = [*start * . * start*]$.

Emission probabilities Each state i labelled by B_1B_2 emits a feature vector according to a distribution $b_i(x)$ characteristic of the *bol* B_2 preceded by B_1 :

$$\begin{aligned} b_i(x) &= p(O_t = x | q_t = i) \\ &= p(O_t = x | b_t = B_2, b_{t-1} = B_1) \end{aligned}$$

$b_i(x)$ represents the probability of observing x knowing that the state at time t is i. In this work $b_i(x)$ is either modelled by a single mixture (a Gaussian vector distribution with diagonal covariance matrix) or a mixture of two Gaussian distributions. For example, in the single mixture case, the feature vectors are modelled with a single vector distribution of 4 Gaussians distributions (where each Gaussian characterizes the mean, variance and relative weights of the frequency content in each pre-defined frequency bands, see section 3.3).

The HMM model described above is a **trigrams model** (3-grams) since the parameters of this model only depends on the two previous events. In this paper, 4-grams models (the three previous events are used to estimate the various probabilities) are also used.

3.4.2 Training

Transition probabilities are estimated by counting occurrences in the training database : If the state i is labelled by B_1B_2, j by B_2B_3, and if C(s) is the number of occurrences of subsequence s in the annotated corpus, then :

$$a_{ij} = p(b_t = B_3 | b_{t-1} = B_2, b_{t-2} = B_1) = \frac{C(B_1B_2B_3)}{C(B_1B_2)}$$

If i is labelled by B_1B_2, emission probabilities for this state are estimated with :

- empirical mean and variance estimators on the set of feature vectors : $A_i = \{x_t / b_t = B_2, b_{t-1} = B_1\}$ - in the case of a simple Gaussian model
- 8 iterations of the EM algorithm after a random affectation to mixtures in the case of a mixture model.

3.4.3 Recognition

Recognition is performed through the classical iterative Viterbi algorithm. After an initialisation step with $v_{q_0}(0) = 0$ and $v_{q \neq q_0}(0) = -\infty$, the algorithm includes:

Iteration

$$v_l(t+1) = \log b_l(f_{t+1}) + \max_{q \in Q} \{v_q(t) + \log(a_{ql})\}$$

$$backtrack_l(t) = argmax_{q \in Q} \{v_q(t) + \log(a_{ql})\}$$

Backtracking The most likely *bols* sequence b^* is the one which maximizes the log-likelihood and is calculated by :

$$q(T) = argmax_{q \in Q} \{v_q(T)\}$$

$$q(t) = backtrack_{q(t+1)}(t) \quad \text{and} \quad b(t) = Etq(q(t))$$

where $Etq(q(t))$ represents the label (i.e the *bol*) of state $q(t)$.

Implementation issues In practice, it is needed to use the log-likehood version of this algorithm which permits to avoid the scaling problems. Another practical problem is the inability of the model to recognize some *bols* sequences which were not present in the training set. To authorize such sequences to be recognized (i.e. to permit all transitions: $a_{ij} \neq 0$), the iteration step of the Viterbi algorithm has been slightly modified to : $v_l(t+1) = \log b_l(f_{t+1}) + \max_{q \in Q} \{v_q(t) + \log((1-\varepsilon)a_{ql} + \varepsilon)\}$

4 Experimental results

4.1 Database and evaluation protocol

4.1.1 The database

The database used for this study is made of 64 phrases with a total of 5715 *bols*. Some of these phrases are long compositions with themes / variations (*kaida*), some others are shorter pieces (*tukra*) or basic *taals*. This database is sub-divided in three sets, each one corresponding to a specific model of tabla:

- *Tabla #1:* recorded using a cheap quality tabla whose strokes are much less resonant than professional instruments. The *Dayan* of the tabla was tuned in C#3 and the recording was made using good home-studio equipment (Sennheiser e825s microphone), in low reverberation and low noise conditions.
- *Tabla #2:* recorded using a high quality instrument and with a *Dayan* tuned in D3. The recording was made with good home-studio equipment (Sennheiser e825s microphone), in low reverberation and low noise conditions.
- *Tabla #3* recorded using another high quality instrument and with a *Dayan* tuned in D3. This set was recorded in a noisier environment and with a greater distance between the microphone and the musician.

4.1.2 Evaluation Protocol 1

For evaluation, the usual cross-validation approach was followed (often called ten-fold procedure in the literature [Herrera et al., 2003]). It consists in splitting the whole database in 10 subsets randomly selected and in using nine of them for training and the last subset (i.e. 10 % of the data) for testing. The procedure is then iterated by rotating the 10 subsets used for training and testing. The results are computed as the average values for the ten runs. In this protocol, one iteration thus consists in using 90 % of each data set (90 % of *Tabla #1, Tabla #2* and *Tabla #3*) and to use the remaining data (coming from each set) for testing. In this protocol, it is important to note that all instruments and conditions are known in the training phase.

4.1.3 Evaluation Protocol 2

To evaluate the generalization properties of our algorithm, a second protocol is used where training sets and testing set correspond to different instruments and recording conditions. In this protocol, the training set consists of all signals from 2 sets (for example 100 % of *Tabla #1* and *Tabla #2*) and the testing set consists of all signal from the third set (i.e. 100 % of *Tabla #3*). This protocol therefore allows to evaluate the generalization properties of our approach since the training set does not include signals that share the same environment acoustical properties and instrument than those of the testing set.

4.2 Results

The results obtained for our novel approach based on HMM language model is compared to simpler classifiers such as Kernel density estimator, k-NN (k nearest neighbors) and a naive Bayesian approach. These simple classifiers are briefly described below:

k-Nearest Neighbors (k-NN) The k-NN approach is a popular technique and has already been used in several studies on musical instruments (see for example [Herrera et al., 2003],[Kaminskyj, 2001]). It consists in building a set of couples associating a feature vector F with the corresponding *bol* B_j. If $k = 1$, a stroke B with feature vector F is then identified as the *bol* B_j whose feature vector $F_j = \overline{f}_{1,..,12}$ is the closest to the feature vector F. For k greater than 1, the decision is taken on the most represented *bol* B_j amongst the k nearest feature vectors F_j. The distance used is a classical Euclidian distance on the normalised feature vectors:

$$\overline{f_i} = \frac{f_i - \mu_{f_i}}{\sigma_{f_i}}$$

where μ_{f_i} and σ_{f_i} respectively correspond to the mean and standard deviation of the parameter f_i on the training set.

Naive Bayes This approach consists in building a Gaussian model for each *bol* B_j. All training data for a given *bol* are then used to build a Gaussian model for each element f_i of the feature vector leading to a set of 24 parameters per *bol* B_j (each element f_i of a given *bol* B_k is then characterized by its mean and variance, respectively $\mu_{(f_i,B_k)}$ and $\sigma_{(f_i,B_k)}()$. The chosen *bol* B_k is then the one that maximises the likelihood knowing the observed feature vector F:

$$p(B = B_k | o = F) = \prod_{i=1}^{12} \frac{1}{\sigma_{(f_i,B_k)}\sqrt{2\pi}} . e^{-\frac{(f_i - \mu_{(f_i,B_k)})^2}{2\sigma^2_{(f_i,B_k)}}}$$

Kernel Density estimator This approach does not generate a real abstract model of the data. It merely approximates the distribution of the data by a sum of functions called the "kernel". This Kernel can have different properties and can for example lead to very smooth approximation of the distribution. Note that one of the advantage of this approach is that it can describe non Gaussian distributions.

For all three classifiers described above the implementation proposed in Weka is used ([Waikato, 2003]).

As mentioned in section 4.1, two evaluation protocols are used. The transcription results obtained using these two protocols are respectively summarized in Table 1 and Table 2.

First, it is important to note that due to the sound quality of the database, the segmentation is very efficient and resulted in 98.3% correct segment detection (only 97 errors - 70 strokes not detected, 27 wrongly detected onsets) on the whole database.

Database # of *bols*	All 5715	Tabla #1 1678	Tabla #2 2216	Tabla #3 1821
Classification using only features of stroke n				
Kernel density estimator	81.7%	81.8%	82.4%	85.2%
5-NN	83.0%	81.7%	83.3%	85.6%
Naive Bayes	76.6%	79.4%	78.6%	78.5%
Classification using features of stroke $n, n-1, n-2$				
Kernel density estimator	86.8%	86.0%	88.7%	92.0%
5-NN	88.9%	87.2%	88.4%	90.6%
Naive Bayes	81.8%	86.5%	83.8%	85.8%
Classification using language modelling				
HMM, 3-grams, 1 mixture	88.0%	90.6%	89.9%	92.6%
HMM, 4-grams, 2 mixtures	**93.6%**	**92.0%**	**91.9%**	**93.4%**

Table 1: Recognition scores using evaluation protocol 1

In Table 1, it can be observed that the simple classifiers (Kernel density estimator, k-NN, Naive Bayes) already obtain satisfactory results and therefore indicate that the basic parametric representation chosen is valuable. However, these models cannot deal with time dependencies. A simple way of modelling these dependencies is to extend the feature vectors with the features of the 2 previous *bols*. In these experiments, the feature vectors are simply the concatenation of the feature vector with those of the two previous strokes. Clearly, this simple approach succeeds to model some of the time dependencies and relevantly improves the recognition score, especially for non-parametric models. However, the best results are achieved with the HMM language model approach that appears to be a very appropriate solution for the transcription of Tabla signals.

The Table 2 summarizes the results obtained with the protocol 2 where the training sets and test set are recorded on different instruments and in different conditions. In other words the condition and specificities of the test set are not known in the training phase. In this condition, the classifier which gave the best recognition scores with protocol 1 failed to properly generalize and to adapt to other instruments or recording conditions. This is clearly another advantage of the HMM approach where performance remain at a very high level even with signal recorded in different conditions. Though, it is interesting to note that in this case the simpler HMM model leads to the best results. This may be explained by the fact that using more complex mixture models as distributions for the acoustic features may lead to an overfitting of the training data.

In further analyzing the errors of our algorithms, it can be noticed that most of them occur with similar strokes. If *bols* are grouped in 5 categories corresponding to a similar production mechanism (see Table 3), one can observe that most of the recognition errors happen within the same stroke category (*"non-resonant bayan strokes"*).

It is worth to note that the overall category recognition rate is fairly high since it reaches 95.7% with the best model (HMM, 4-grams, 2 mixtures models).

5 Tablascope : a fully integrated environment

5.1 Description

The tabla transcription algorithm has been embedded in a fully integrated environment called Tablascope. This system has been developed following a client/server approach which provides a great level of flexibility and portability. All the signal processing / recognition modules are implemented in a server written in C++, while the graphical client is a Java application. A protocol based on the serialization of objects as XML messages has been chosen for the communication between the two modules. The database and the transcriptions are also stored using the XML format. Tablascope also integrates the possibility to export transcription as Csound scores. An overview of the environment is presented in figure 4.

5.2 Applications

Since the recognition algorithm runs in realtime, a number of other applications are possible including:

Tabla sequence generation or synthesis In that context, the user types in the desired *bols* sequence and the overall tempo. Then, the HMM sequence model can check the phrase input by the user, i.e. checks whether it contains unknown *bols* or forbidden *bols* sequences. In this case, Tablascope suggests the nearest correct phrase according to an edit distance. Synthesis is then performed by an overlap-add procedure slightly adapted to permit pitch shifting.

Tabla-controlled synthesizer Tablascope can also be used as a MIDI controller, that is to control a synthesizer with input coming from a Tabla without using any other sensor than a microphone. In such an application, a given *bol* can be associated to a specific instrument of the synthesizer giving the Tabla player a wider range of performance capabilities.

6 Conclusion

Most of the music transcription systems are focused on melody and western music. In this paper, we presented a transcription system for Tabla - a north Indian percussion instrument - based on a statistical machine learning approach. The identification of Tabla *bols* with a low (6.5%) error rate is achieved through the use of a HMM model in a similar way that language models are exploited in large vocabulary speech recognition systems. Additional functions such as tabla sequence correction and generation (or synthesis) are integrated in a user friendly environment called **Tablascope**. Although this study is conducted on Tabla signals, it can easily be generalized to other types of percus-

Training set Test set	Tabla #1 & Tabla #2 Tabla #3 (noisy rec.)	Tabla #2 & Tabla #3 Tabla #1 (cheap quality)
5-NN	79.8 %	78.2 %
HMM, 3-grams, 1 mixture	**90.2 %**	**88.4 %**
HMM, 4-grams, 2 mixtures	84.5 %	85.0 %

Table 2: Generalization test using evaluation protocol 2

a	b	c	d	e	¡- classified as
1241	22	2	7	8	a: resonant *dayan* strokes (Tin, Na, Tun...)
20	1076	2	1	5	b: *bayan+dayan* strokes (Dhin, Dha...)
1	3	766	5	20	c: resonant *bayan* strokes (Ge, Gi...)
8	2	2	448	61	d: non-resonant *bayan* strokes (Ke, Ki...)
11	7	6	50	1938	e: non-resonant *dayan* strokes (Te, Ti, Tek...)

Table 3: Confusion matrix by *bol* category (with HMM, 4-grams, 2 mixtures classifier)

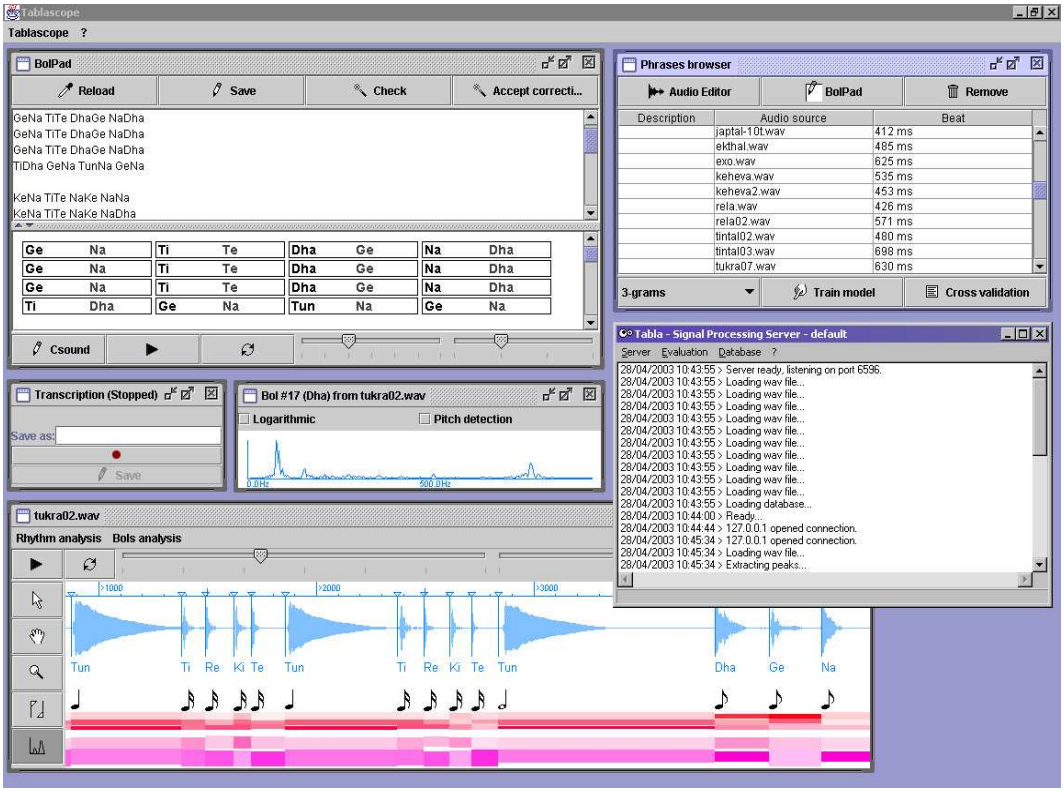

Figure 4: An overview of the **Tablascope** environment : phrase edition (BolPad), audio annotation, realtime transcription, corpus management and tuning windows

sive signals such as percussion or drum loops which also show strong time dependencies. As a matter of fact, except for the signal parametric representation (i.e the acoustic front-end) that is rather specific to Tabla, all other modules are fairly generic.

7 Acknowledgements

The authors wish to thank Sanjay Pandit and Uday Deshpande for their precious advice about Tabla theory, Claire Waast-Richard for fruitful discussions on current speech recognition technologies and Nicolas Lelandais for the tabla performances used in this work.

References

[Alonso et al., 2003] Alonso, M., Badeau, R., David, B., and Richard, G. (2003). Musical tempo estimation using noise subspace projections. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA '03)*.

[A. Kapur and Cook, 2002] A. Kapur, G. Essl, P. D. and Cook, P. R. (2002). The electronic tabla controller. In *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02)*.

[Fletcher and Rossing, 1998] Fletcher, N. and Rossing, T. (1998). *The Physics of Musical Instruments*. Springer verlag, 2nd edition edition.

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

[Herrera et al., 2000] Herrera, P., Amatriain, X., Batlle, E., and Serra, X. (2000). Towards instrument segmentation for music content description: A critical review of automatic musical instrument classification. In *In Proc. of ISMIR2000*.

[Herrera et al., 2003] Herrera, P., Dehamel, A., and Gouyon, F. (2003). Automatic labeling of unpitched percussion sounds. In *114th AES Convention*, Amsterdam, The Netherlands.

[Kaminskyj, 2001] Kaminskyj, I. (2001). Multi feature musical instrument sound classifier. In *Proceedings of Asutralasian Computer Music Conference*.

[Klapuri, 1999] Klapuri, A. (1999). Sound onset detection by applying psychoacoustic knowledge. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, Phoenix, Arizona.

[Martin, 1999] Martin, K. (1999). *Sound Source Recognition: A Theory and Computational Model*. Ph.d., Massachussetts Institute of Technology.

[McDonald and Tsang, 1997] McDonald, S. and Tsang, C. (1997). Percussive sound identification using spectral centre trajectories. In *Proceedings of 1997 Postgraduate Research Conference*.

[Rabiner and Juang, 1993] Rabiner, L. and Juang, B. (1993). *Fundamentals of speech recognition*. Englewood Cliffs, NJ.

[Raman, 1934] Raman, C. (1934). The indian musical drum. In Science, P. I. A., editor, *Reprinted in Musical Acoustics: selected reprints, ed. T.D. Rossing Am. Assn. Phys. Tech., College Park, MD, 1988*.

[Raphael, 2002] Raphael, C. (2002). Automatic transcription of piano music. In *Proceedings of ISMIR2002*, pages 15–16.

[Roh and Wilcox, 1995] Roh, J. and Wilcox, L. (1995). Exploring tabla drumming using rhythmic input. In *Proceedings of the CHI '95*, pages 310–311.

[Scheirer, 1998] Scheirer, E. D. (1998). Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America*, 103(1):588–601.

[Sillanpää et al., 2000] Sillanpää, J., Klapuri, A., Seppänen, J., and Virtanen, T. (2000). Recognition of acoustic noise mixtures by combined bottom-up and top-down approach. In *Proceedings of European Signal Processing Conference, EUSIPCO-2000*.

[Waikato, 2003] Waikato (2003). Weka 3: Machine learning software in java. In *http://www.cs.waikato.ac.nz/ml/weka/*.