

# TABLA GYAN: A SYSTEM FOR REALTIME TABLA RECOGNITION AND RESYNTHESIS

*Parag Chordia*

Georgia Institute of Technology  
Music Technology Group

*Alex Rae*

Georgia Institute of Technology  
Music Technology Group

## ABSTRACT

We describe a realtime system that can listen to performances of tabla and identify the strokes as they are played. The system stores this information along with onset times in a symbolic score that is used as the basis for resynthesis and transformation. We describe playback and transformation possibilities of this system. Finally, we propose that such a system can be generalized with very few modifications to a variety of performance scenarios involving percussion interaction.

## 1. BACKGROUND AND MOTIVATION

Tabla is the most widely used percussion instrument in Indian music, both as an accompanying and solo instrument. Its two component drums are played with the fingers and hands and produce a wide variety of different timbres, each of which has been named. There are approximately fifteen acoustically distinct strokes that fall in three broad categories: 1) resonant, ringing strokes played on the treble drum, 2) non-resonant, noisy strokes played on either drum, 3) low, resonant strokes played on the bass drum. These named strokes form the basic vocabulary of tabla music and are played in sequence to form typical phrases. Tabla solo is a centuries-old tradition centered around extended structured improvisations. The demands of this format have led to the sophisticated use of timbre and rhythm as a foreground element. In this music, the choice of strokes is precise, each one functioning like a note in a melody; the timbral and rhythmic structures are equally important and carefully integrated into a singing line.

To date, little work has been done on percussion timbre recognition for interactive systems. Most of the work in this area has been focused on non-realtime scenarios, and many applications center on areas such as genre classification and music recommendation [11, 12]. There were two primary motivations for creating a realtime system. First, we wanted to be able to incorporate tabla into the context of interactive electronic music. Second, we wished to be able to use the tabla as a musical controller and input device. Without building a specialized hardware interface such as Kapur's ETabla [7], both of these tasks require identifying tabla strokes and their timing. With this approach, the system we have developed could be generalized to many other percussion instruments with no modi-

fications other than the creation of new training sets (described below in Section 2.2). We also envision that this realtime stream of information may be useful for a variety of other applications including automatic accompaniment.

### 1.1. Related Work

The problem is similar to the instrument or timbre identification problem that has received considerable attention from music information retrieval (MIR) researchers in the past decade [4, 8], in which an isolated tone or passage is presented to the system for classification. Percussion transcription has specifically been attempted by researchers such as Gillet and Richard [5], primarily in the context of drum-loop recognition. The current work directly builds on the previous work by Chordia on non-realtime tabla transcription [2].

## 2. METHOD

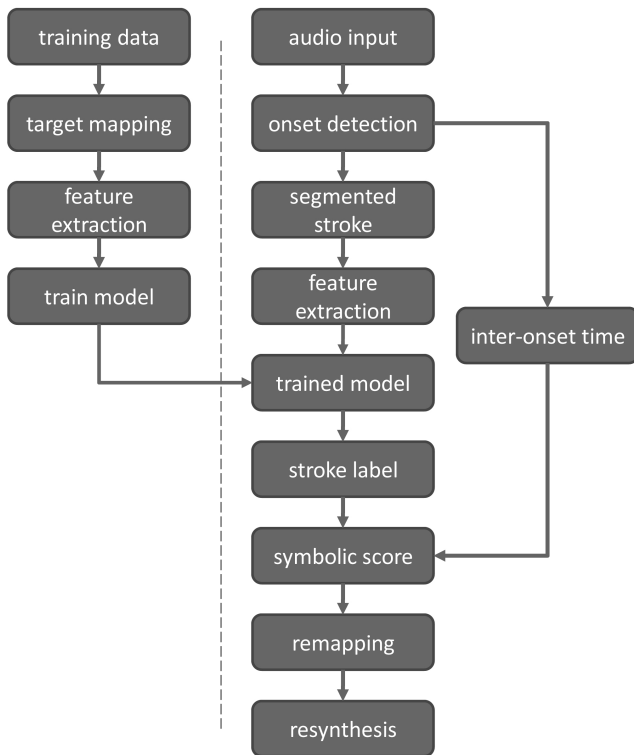
### 2.1. Architecture

Figure 1 shows the block diagram of the system. The system is coded in Java as an external for Max/MSP. The WEKA java package [13] is used for pattern recognition, and the jAudio package [9] FFT implementation is used as part of the feature extraction process. A short video showing the system in action can be found at <http://paragchordia.com/research/tablaGyan/>.

The incoming signal is segmented into discrete strokes by detecting the onsets, and a stroke is considered to be the waveform captured between onsets. The incoming audio is buffered, and the detection of a new onset causes the contents of that buffer to be sent to the feature calculation block. The last stroke is analyzed when the system determines the phrase to have ended. The buffering and analysis then halts until a new phrase is initiated by the performer.

### 2.2. Feature Calculation

Frequency analysis is performed and a variety of features are calculated for each stroke. The primary features are twenty-four MFCCs (not including the 0th), which give a summary of the spectral shape. Additionally, we calculate spectral centroid, variance, skewness, kurtosis, slope (linear and logarithmic), and roll-off. The inspiration for



**Figure 1.** Block diagram for the Tabla Gyan System

these features comes largely from statistics, where they are used to describe the shape of probability distributions. The precise definitions of these and other features can be found in [6]. A cutoff of 90% of total energy was used for the spectral roll-off. The decision to use spectral features, rather than temporal or other, is due to their established success in instrument recognition, and their robustness to segmentation errors. Specifically, previous work by Chordia [2] showed this feature set to be effective for tabla stroke classification.

Prior to realtime classification, a model must be trained on a large set of samples, each of which must be analyzed as described above. Each sample consists of a soundfile of one hand-annotated stroke. A large database, consisting of 9,532 of these samples from three tabla performers, two professional and one amateur, was used to train the system. The recordings were made at different times using different drums under widely varying recording conditions. Each soundfile was read, its name parsed to derive the target, and the features calculated. The parsing was not simply a one-to-one mapping, due to idiosyncrasies in naming practices for tabla strokes. Some strokes have several different names based on where they occur in a phrase, or to facilitate rapid vocal recitation; some have different names because they are played in a slightly different manner but share essentially the same timbre; most problematically, some timbrally different strokes share the same name, and are normally differentiated based on context. The solution implemented was to create a mapping based on acoustic rather than nominal categories.

The resulting feature vector and category label was gen-

erated for each stroke and stored in a feature matrix, which was used to train the classification model. Depending on the anticipated performance context, the stroke selection was varied to train a more finely tuned model. Compared with a more general classification system, in which there may be no foreknowledge of the specific characteristics of the material, a performance context often allows us to know the performer, performance style, specific choice of instrument, etc.

### 2.3. Classification

To build the classifier, we used the WEKA Java package. A variety of different techniques were tried, including multivariate Bayesian models and classification trees, but we found that the Sequential Minimal Optimization implementation of a Support Vector Machine classifier gave slightly better performance [10]. In a ten-fold cross validation evaluation with fifteen different strokes, accuracy was 84.3%.

Feature vectors extracted from incoming strokes are input to the trained classifier as soon as the feature calculation is completed. The classifier returns a target category label, which is then stored in a text file along with the peak amplitude calculated over the stroke, and the duration (inter-onset interval). This text file then serves as a symbolic score.

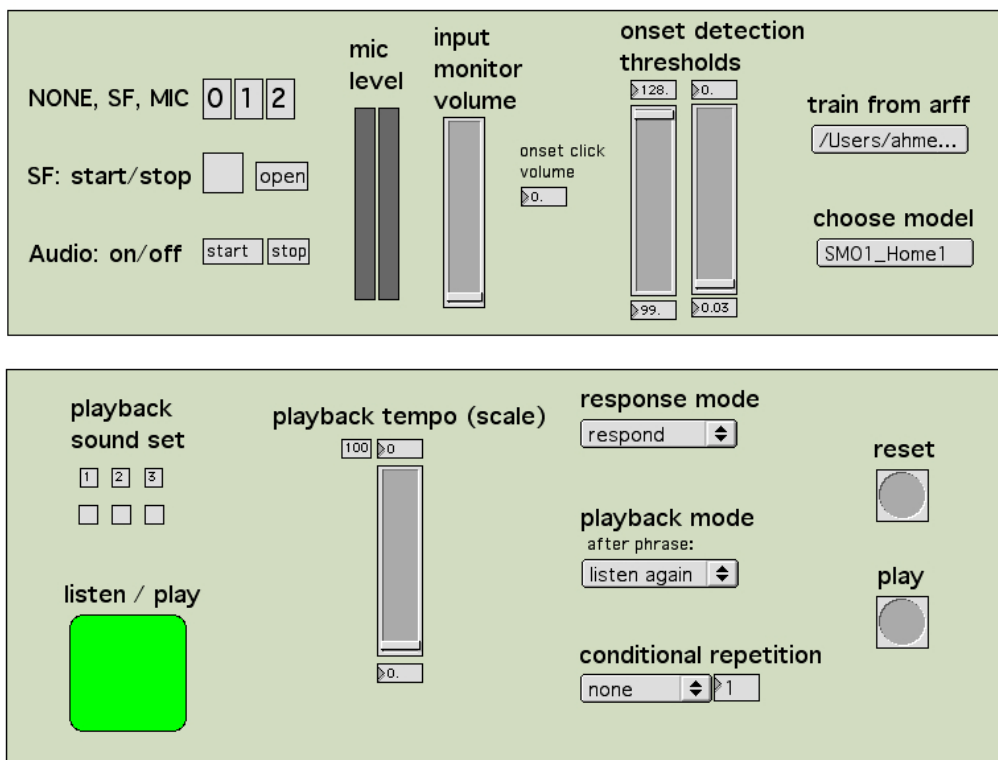
## 3. INTERACTION AND PLAYBACK

We created a user interface in Max/MSP, shown in Figure 2, allowing the user to build the classification model or choose a previously trained model, select a sound source, select a sound set for the output, and manipulate playback of the score. The interface also allows a number of parameters to the algorithm to be set.

From within the Max/MSP patch, a user can easily build a new classification model by loading a feature matrix stored in WEKA's arff format (Attribute-Relation File Format) [13]. Equivalently, one can load a previously trained model. This allows for easy customization of the classification algorithm to specific instruments or performers.

The system was designed to respond after it heard a phrase, in a call and response manner. Our working definition of phrase was a sequence of strokes followed by a silence greater than the average duration plus two times the standard deviation of the duration. This provided a reasonable balance between responsiveness and false starts.

On the output, the stroke categories could be mapped to a fewer number of categories. The motivations for this depend on the context in which the system is being used. Primarily, it serves to reduce the perceptual inaccuracy of the system by binning perceptually and timbrally similar categories. It can also be used flexibly, to map the output to a variable number of sound types for aesthetic reasons. These mappings can be switched dynamically.



**Figure 2.** User Interface for Tabla Gyan System

A set of sound banks was created for playback; each stroke target is mapped to a particular sound as described above. Several similar versions of each sound category were produced. The resynthesis is accomplished quite simply, by playing one of the appropriate soundfiles, scaled by the peak amplitude of that stroke, for the duration of that stroke. Beyond simple resynthesis, one of our goals in designing the sound banks was to preserve the timbral relationships of the tabla while using a new, and potentially very different sounds.

Compared with an audio recording, the system’s ability to interpret the signal and describe it in symbolic terms gave us great flexibility during playback. In addition to remapping the timbres, we were able to easily change the tempo as well as to make transformations that were conditional on the stroke type. For example, we implemented conditional repetition, where a certain type of stroke could be doubled or tripled (i.e. two or three strokes in the space of one stroke) or generally n-tupled. This proved to be an interesting way of creating rhythmic patterns that were variations of the original.

Inspired by the tabla form *qaida*, a theme and variation form that is central to solo tabla music, we implemented a simple variation generator based on similar principles. In tabla music, phrases are built from sequences of strokes, that form hierarchical units, similar to language where sentences are built from words which are in turn composed of phonemes. In *qaidas*, the theme is divided into sub-phrases that are rearranged, repeated, and omitted to create variations. Some initial experiments were done to ex-

tract sub-phrases based on the tactus of the phrase, where the tactus was determined using a simple algorithm based on the inter-onset intervals. The sub-phrases were then rearranged to form larger phrases for playback. Modeling of *qaida* variation is an interesting subject that we hope to explore further.

#### 4. EVALUATION

We evaluated the system by observing its response to a variety of tabla compositions, both performed live and from recordings. We selected compositions spanning different performers, different stroke patterns, and different tempi. In all cases the compositions were recognizable even when there were errors, and timbral information significantly improved recognizability beyond what was indicated by onset times alone.

One of the challenges of a performance system is that it must sound good. Relatively high accuracy is no guarantee of this. For example, even if we were able to achieve the non-realtime threshold of 90%, this would mean a typical phrase of four seconds at moderate tempo would contain between two and four misclassified strokes. If these were timbrally dissimilar it would significantly alter the percept of the phrase. In our system, the accuracy rate is closer to 70-75%, often due to segmentation errors that led to missing attacks. In the case of percussion, this is particularly damaging since most discriminative information occurs early in the stroke. Most of the errors we observed were due to the most commonly occurring stroke,

a non-resonant stroke called *te*, being substituted for the true stroke. Listening to the original training samples suggested an explanation: when *te* is preceded by a resonant stroke the sound continues to ring through the *te* stroke, which may itself be of low amplitude. This means that the timbral profile of the *te* stroke is very similar to that of a resonant stroke without its attack portion. Incorporation of musical context and more accurate detection of onsets will help to alleviate this problem.

We also did preliminary work on extending the system to mrdangam, the main South Indian percussion instrument. By substituting our training database with a large set of labeled mrdangam strokes, we were able to use the same system without further modification and with similar results. In general, this substitution would allow for interaction with many other percussion instruments.

## 5. FUTURE WORK

We are currently working on several extensions to the system. A more robust approach to beat detection than the current one is to use the autocorrelation of a detection function [3]. This approach will also allow us to include a synchronous interaction mode in addition to call and response.

Most rhythm sequences are highly structured with the current stroke highly dependent on previous strokes. Most of the mistakes that the current system makes are due to acoustical ambiguities that could be clarified by an awareness of musical context. One approach to solving this problem is to use an n-gram model, where the probability of the current stroke is based not only on the current feature vector but recent past feature vectors as well [5]. In addition to improving classification accuracy it would allow us to perform variations at a more meaningful phrase level, since typical phrases would show up as commonly occurring n-grams.

Finally, we are working to use the symbolic information to create notation and analytic scores for tabla music based on the *\*\*bol* representation and notation system created by Chordia [1].

## 6. CONCLUSIONS

We have described a system that can segment and label incoming drum strokes for flexible playback and transformation in performance situations. Specifically, we tested this architecture on tabla music, a complex percussion tradition that systematically uses a wide variety of timbres. Additionally we have implemented several transformations, such as conditional repetition and *qaida*-inspired variation, that depend on such signal understanding. We believe that the architecture presented will prove broadly useful for realtime percussion interaction.

## 7. REFERENCES

- [1] P. Chordia. Automatic transcription and representation of solo tabla music. *Computing in Musicology*, 13, 2003.
- [2] P. Chordia. Segmentation and recognition of tabla strokes. In *Proceedings of International Conference on Music Information Retrieval*, 2005.
- [3] M. E. P. Davies and M. D. Plumbley. Beat tracking with a two state model. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2005.
- [4] S. Essid, G. Richard, and B. David. Musical instrument recognition based on class pairwise feature selection. In *Proceedings of International Conference on Music Information Retrieval*, 2004.
- [5] O. Gillet and G. Richard. Supervised and unsupervised sequence modeling for drum transcription. In *Proceedings of International Conference on Music Information Retrieval*, 2007.
- [6] P. Herrera, G. Peeters, and S. Dubnov. Automatic classification of musical sounds. *Journal of New Music Research*, 32(1):3–21, 2003.
- [7] A. Kapur, G. Essl, P. Davidson, and P. R. Cook. The electronic tabla controller. *Journal of New Music Research*, 32(4):351 – 360, 2003.
- [8] K. Martin and Y. M. Kim. 2pmu9. musical instrument identification: A pattern-recognition approach. In *Proceedings of 136th meeting of the Acoustical Society of America*, 1998.
- [9] D. McEnnis, C. McKay, and I. Fujinaga. jaudio: Additions and improvements. In *Proceedings of International Conference on Music Information Retrieval*, 2006.
- [10] J. Platt. Machines using sequential minimal optimization. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.
- [11] V. Sandvold, F. Gouyon, and P. Herrera. Percussion-related semantic descriptors of music audio files. In *Proceedings of the 25th International Audio Engineering Society Conference*, 2004.
- [12] V. Sandvold, F. Gouyon, and P. Herrera. Drum sound classification in polyphonic audio recordings using localized sound models. In *Proceedings of International Conference on Music Information Retrieval*, 2005.
- [13] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.