TONIC-INDEPENDENT STROKE TRANSCRIPTION OF THE MRIDANGAM

Akshay Anantapadmanabhan[1], Juan P. Bello[2], Raghav Krishnan[1] Hema A Murthy[1],

[1]*Indian Institute of Technology Madras, Dept. of Computer Science and Engineering, Chennai, Tamil Nadu, 600036, India*

[2]*New York University, Music and Audio Research Lab (MARL), New York, NY, 10012, USA*

Correspondence should be addressed to Akshay Anantapadmanabhan (akshay.anantapadmanabhan@gmail.com)

ABSTRACT

In this paper, we use a data-driven approach for the tonic-independent transcription of strokes of the mridangam, a South Indian hand drum. We obtain feature vectors that encode tonic-invariance by computing the magnitude spectrum of the constant-Q transform of the audio signal. Then we use Non-negative Matrix Factorization (NMF) to obtain a low-dimensional feature space where mridangam strokes are separable. We make the resulting feature sequence event-synchronous using short-term statistics of feature vectors between onsets, before classifying into a predefined set of stroke labels using Support Vector Machines (SVM). The proposed approach is both more accurate and flexible compared to that of tonic-specific approaches.

1. INTRODUCTION

The *mridangam* is the primary percussion accompaniment instrument in Carnatic music, a sub-genre of Indian classical music. It is a pitched percussive instrument – like e.g. the tabla, conga and timpani – with a structure including two loaded, circular membranes, able to produce a variety of sounds often characterized by significant harmonic content [18, 21]. These sounds have been organized into a standard vocabulary of drum strokes that can be used for the transcription of mridangam performances. Transcription is useful for students of Carnatic percussion as the artform is largely focused on being able to discern and reproduce the vocabulary of the instrument (both verbally and while performing the instrument). Furthermore, the ability to transcribe mridangam strokes can provide insight into the instrument's practice for musicians and musicologists of other musical traditions as well.

There have been numerous efforts to analyze and characterize percussion instruments using computational methods. However, the majority of these approaches have been focused on unpitched percussion instruments, in the context of Western music [7, 13, 12, 11, 24, 19]. There have been some works focusing on non-western percussion, specifically on novel approaches to the automatic transcription of tabla performances [8, 5], as well as previous work on the transcription of mridangam strokes [1]. The latter approach spectrally characterizes each of the resonant modes of a given mridangam, and uses the resulting spectra as the basis functions in a non-negative matrix factorization (NMF) of recorded performances of the respective instrument[1].The resulting NMF activations are used as features in an HMM framework to transcribe the audio into a sequence of mridangam stroke labels. Although [1] provides great insight into the strokes of the mridangam and their relation to the modes of the drumhead (as described by [18]), the transcription approach is limited by its dependency on prior knowledge about the specific modes of the instrument. Therefore the method cannot be generalized to other instruments or different tonics [1].

In this paper, we attempt to address the aforementioned problems by proposing a data-driven approach for the instrument- and tonic-independent transcription of the strokes of the mridangam. Tonic independence is achieved by introducing invariance to frequency shifts in the feature representation, using the magnitude spectrum of the constant-Q transform of the audio signal. Then we use NMF to obtain a low-dimensional, discriminative feature space where mridangam strokes are separable. Note that, as opposed to [1], we do not fix but learn the basis functions of the decomposition using an independent dictionary of recordings. Then, we make the re-

[1]Note that the use of NMF for transcribing percussive instruments is not new, as exemplified by [7, 13]

sulting feature sequence event synchronous using simple statistics of basis activations between onsets. Finally, we present these features to an SVM-based classification stage. The resulting approach is both more robust and flexible than previous instrument- and tonic-specific methods.

The rest of this paper is organized as follows: Section 2 briefly introduces the mridangam and describes the strokes that can be produced by the instrument; section 3 provides the details of our transcription approach, and clearly motivates its different parts; section 4 introduces the experimental setup used to validate our technique, while section 5 presents our results and analyses. Finally, the paper is summarized and concluded in section 6.

2. INTRODUCTION TO THE MRIDANGAM

The mridangam has been noted in manuscripts dating as far back as 200 B.C. and has evolved over time to be the most prominent percussion instrument used in South Indian classical music [9]. It has a tube-like structure made from jack fruit tree wood covered on both ends by two different membranes. Unlike most western drums which cannot produce harmonics due to their uniform circular membranes [15], the mridangam is loaded at the center of the treble membrane (*valanthalai*) resulting in significant harmonic properties with all overtones being almost at integer ratios of each other[18, 21]. The bass membrane (*thoppi*) is loaded at the time of performance, increasing the density of the membrane, and helping it propagate its low-frequency sound [18].

Mridangams are hand-crafted instruments that are built to be tuned in reference to a specific tonic. The frequency range of the instrument is traditionally limited to one semi-tone above or below the original tonic the instrument is designed for. Hence, professional performers use different instruments to account for these pitch variations. Therefore, it is important that transcription of mridangam strokes is independent of both the instrument and the used tonic. In general, the two membranes of the mridangam produce many different timbres. Many of these sounds have been named, forming a vocabulary of basic strokes and associated timbres, which can be roughly classified into the following sound groups:

1. Ringing string-like tones played on the treble membrane. *Dhin*, *cha* and *bheem*[2] are examples. These tones are characterized by a distinct pitch, sharp attack and long sustain.

2. Flat, closed, crisp sounds. *Thi* (also referred to as *ki* or *ka*), *ta* and *num* are played on the treble membrane and *tha* is played on the bass membrane. These tones are characterized by an indiscernible pitch, sharp attack and almost immediate decay.

3. Resonant strokes are also played on the bass membrane (*thom*). This tone is not associated with a specific pitch and is characterized by a sharp attack and a long sustain.

4. Composite strokes (played with both hands) consist of two strokes played simultaneously: *tham* (*num* + *thom*) and *dheem* (*dhin* + *thom*)

Altogether, the single-handed and composite timbres add to 10 basic strokes. There are a series of advanced strokes which include slides and combinations of strokes using slides, but that is outside the scope of this paper and is not used in the music corpus.

3. APPROACH

The approach we propose for the transcription of basic mridangam strokes is summarized in Figure 1. It consists of four stages: (1) feature extraction, (2) factorization, (3) summarization, and (4) classification. The following explains each of those stages in detail.

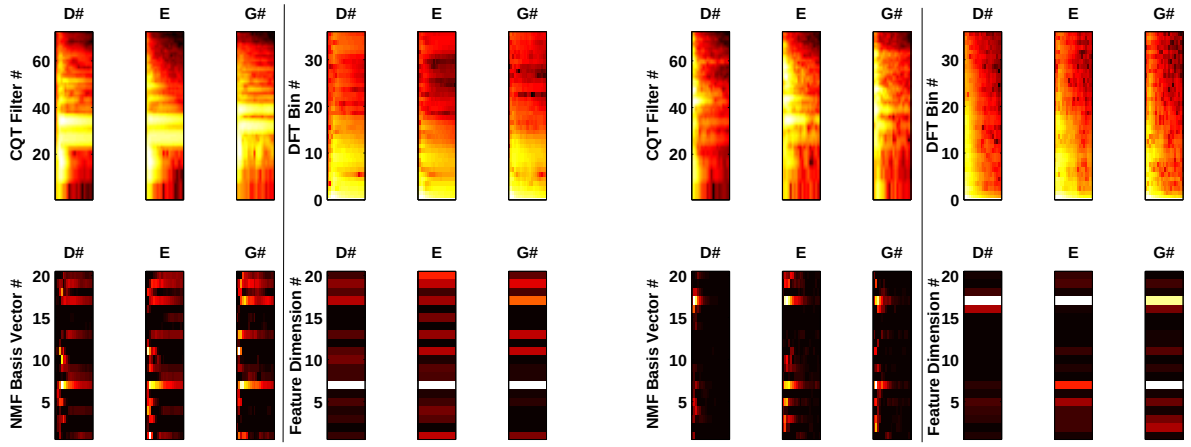
3.1. Feature Extraction

Let us define X as the discrete Fourier transform (DFT) of an N-long segment of the input signal x. We can then compute the constant-Q transform (CQT) of x as [3]:

$$X_{cq} = \frac{1}{\min(N, N_k)} \sum_{v=0}^{N-1} X(v) K_k^*(v) \qquad (1)$$

where K_k is the DFT of the bandpass filter $\mathscr{K}_k(m) = \omega_k(m) e^{-j2\pi f_k m}$, ω_k is a bell-shaped window, $m \in [0, N_k - 1]$, N_k is the filter's length chosen to keep the Q factor constant, $f_k = f_0 \cdot 2^{k/\beta}$ is the filter's center frequency, β is the number of bins per octave, and f_0 is the center frequency of the first filter. Unless otherwise specified, our

[2] The name of this stroke varies between different schools of mridangam.



(a) *dhin* (b) *num*

Fig. 2: For each set of subplots (from left to right), the top first three images show the constant-Q transform and the next three show the magnitude spectrum of each CQT. The bottom first three images correspond to NMF activations and the next three are averaged feature vectors of NMF activations. The plots show the relationship between feature vectors at different stages (extraction, factorization, summarization) of the transcription system across tonics D#, E and G# for pitched stroke *dhin* and unpitched stroke *num*

analysis uses segments of $N = 1024$ samples, with a hop size of N/4, with a CQT filter-bank using $\beta = 12$, $f_0 = 70$ Hz, and a frequency range spanning 6 octaves between 70 - 4.8k Hz. In our implementation we found sufficient to use a simple filter-bank of overlapping triangular filters, as opposed to more elaborate implementations such as the cent filter-bank recently proposed in [20]. This is because better frequency resolution is only of limited use for the analysis of percussive sounds, in contrast to, say, melodic analysis as in the referenced approach. The question of how much resolution is needed is addressed in our experiments.

Unlike the DFT, frequencies in CQT analysis are not linear but log-2 spaced. As a result, pitch transpositions of a given sound result on linear frequency shifts in the CQT representation, as opposed to the change of spacing between spectral peaks that occurs in DFT analysis. This phenomenon can be observed in the top-left three plots in figures 2(a) and (b), which show linear shifts between the magnitude CQT representations of (a) *dhin* and (b) *num* strokes played in three different tonics.

Notably, since we now represent pitch transpositions as linear frequency shifts, taking the DFT of the CQT spectrum encodes those shifts in the phase component. Therefore if we drop phase and keep only the magnitude of the DFT, we obtain a feature representation that is invariant to pitch transpositions, as is clearly shown by the top-right three plots of the same figures. In MIR, applying the magnitude DFT to achieve shift invariance has been previously used in the context of rhythm analysis [17, 16] and cover-song identification [23]. In our implementation we use a hanning window, and an FFT of the same length of the CQT filter-bank.

3.2. Factorization

After taking the magnitude spectrum, we use non-negative matrix factorization (NMF) as a data-driven, feature learning stage that both reduces dimensionality and enhances discrimination. In NMF, a non-negative matrix Z can be decomposed into two non-negative matrix factors, a dictionary B and an activation matrix Y such that:

$$Z \approx BY \tag{2}$$

Given Z is of dimension $m \times t$, then B and Y are of dimensions $m \times n$ and $n \times t$ respectively. Generally $n < t$. This implies that z_i, the i^{th} column of Z, can be represented

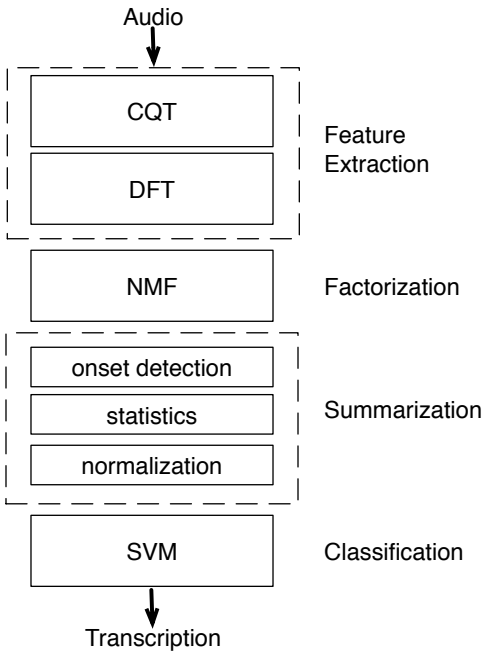


Fig. 1: Block diagram of transcription system

as a linear combination of basis vectors b_j, the columns of B, weighted by the activations y_{ji}, where $j = 1, \cdots, n$. There are numerous algorithms to estimate B and Y, depending on the metric used to quantify the approximation in equation 2. We use the popular Euclidean measure and the multiplicative update rules proposed in [14] to iteratively estimate these parameters. In our experiments, we run 800 iterations during training.

NMF and its variations have been extensively and successfully used for sound and music analysis and transcription [22, 26, 25, 10, 13, 7, 6, 1]. In our approach, we learn the dictionary from a database of five, out-of-sample set of mridangam recordings (details in section 4.1). We then map the magnitude DFT of the signal's CQT onto this dictionary. The dimensionality of the resulting feature vectors is equal to the number of NMF bases, while the length of the vector sequence is equal to the number of frames of the signal's spectrogram used in the computation of the CQT. The 3 bottom-left plots in figures 2(a) and (b) show these intermediate feature sequences for each corresponding stroke, after a factorization using 20 basis functions. It can be seen that the three sequences of feature vectors corresponding to the same stroke carry similar information, despite being performed in different tonics.

3.3. Summarization

After factorization, we make our features event-synchronous by, first, detecting stroke onsets in the signal and, second, summarizing the information between detected onsets using simple, short-term statistics.

For onset detection we use the well-known spectral flux detection function as defined in [2]:

$$d(n) = \frac{2}{N} \sum_{k=0}^{N/2} [H(|X_k(n)| - |X_k(n-1)|)]^2 \qquad (3)$$

where $H(x) = (x + |x|)/2$. This function is smoothed and thresholded using a moving median window, before peak picking to determine the position of onsets. Any false alarm strokes and missed strokes are excluded from the experiment at this point.

Then, for all frames between detected onset positions, we compute the following statistics independently for each feature:

- weighted mean, where the energy of all vectors (normalized to sum to unity) are used as weights
- standard deviation
- maximum value
- minimum value

These statistics are concatenated into a single feature vector, which we use for classification, resulting in a dimensionality of ($v \times$ *number of NMF bases*), where $v \in 1, 2, 3, 4$. The total number of vectors, or instances, in our dataset, is now the number of automatically detected onsets. Unless otherwise specified, we use only the weighted mean for summarization. Examples of the resulting vector per stroke can be seen in the bottom right plots of figures 2(a) and (b). The dimensionality of these example vectors is 20×1. Again, the similarity between the three *dhin* and the three *num* strokes is apparent despite the change in tonic.

3.4. Classification

For classification we use a set of support vector machines (SVM) with RBF kernels in a one-versus-one configuration. The estimated class per stroke is chosen by majority voting. All SVMs are implemented using the libSVM library [4] in WEKA, and parameterized using $C = 1$, and $\gamma = 0.0625$ where C is the soft margin cost and γ is the RBF kernel spread. It is possible that other classifiers and/or configurations return better results, but since the focus of the current paper is on the combined effect of the used feature extraction, factorization and summarization strategies, we leave such testing for future work.

4. EVALUATION

4.1. Data

The database used for validating our automatic transcription approach consists of six recordings of solo improvisations performed by the first author, who is a professional mridangam player. Each recording corresponds to a different tonic, covering the range of semitones between B and E in the chromatic scale. Recordings for tonics D# and E correspond to the dataset used for evaluation in [1], which we will term $\mathscr{D}_1$ – see reference for recording details. It is important to note that each of these recordings was made with a different drum. The set of recordings for the other four tonics, which we term $\mathscr{D}_2$, was recorded at the Dolan studio of New York University, using two MKH800 microphones with a sampling rate of 96 kHz at 24 bits. The channels were averaged into a single mono track, and resampled to 44.1 kHz at 16 bits to match the specifications of $\mathscr{D}_1$. These recordings were made with a single, more modern, drum that allows for a full tone variation from its nominal tonic in each direction.

Unless otherwise indicated, we use the combined dataset $\mathscr{D} = \mathscr{D}_1 \cup \mathscr{D}_2$ throughout our experiments. It contains a total of 7,170 strokes. The first two columns of Table 2 summarize the number of strokes corresponding to the different tonics. All strokes in the dataset were hand labeled by the first author according to the 10 stroke classes introduced in section 2. The distribution of strokes both by type and tonic, can be seen in Figure 3. It can be observed that strokes *thi* and *tha* occur most frequently, while *bheem* occurs the least across all tonics, which roughly corresponds to common practice.

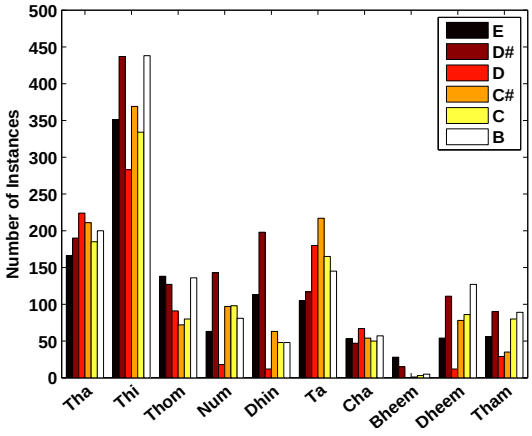


Fig. 3: Histograms of stroke occurrences for instruments tuned to B,C,C#,D,D#,E

Finally, aside from $\mathscr{D}$, we use a separate (unlabeled) training set for data-driven operations such as NMF and PCA[3]. This training set consists of five 1-minute long segments of live concert solos, performed by a variety of professional artists in the tonics of C#, D, G, G#.

4.2. Experimental Setup

For our experiments, the feature extraction, factorization and summarization stages were implemented in MATLAB, while classification and evaluation was done in WEKA (using libSVM). All experiments were evaluated in terms of the classification accuracy, where accuracy is the ratio of correctly classified to total strokes. Unless otherwise specified, we run all experiments using 10-fold cross-validation. For each experimental run, the feature vectors are standardized across the training set. Throughout our experiments, the default system configuration consists of a CQT analysis with $\beta = 12$ across 6 octaves (resulting in 72 log-spaced bins starting from $f_0 = 70$ Hz), NMF with 20 basis functions, and the weighted mean for summarization.

Our experiments aim to: compare our approach to previous work in [1]; assess the system's invariance to tonic, and to a lesser extent instrument and recording conditions; find the optimal setting of the approach's parameters, namely CQT resolution, the use of NMF, the num-

[3] As will be discussed later, PCA is used as an alternative strategy to validate the relative importance of NMF in the proposed system

ber of basis functions, and the combination of summarization strategies; and report detailed results for the optimal configuration. The results of those experiments are presented and discussed in the following section.

5. RESULTS AND DISCUSSION

5.1. Comparison to Previous Work

Table 1, presents a comparison between our approach, and previous work for mridangam transcription in [1]. The approaches are identified by classifier type: HMM for prior work and SVM for our approach, although the differences are not limited to the choice of classifier. In this experiment we evaluate only on the $\mathscr{D}_1$ dataset, since it is the only data for which we have the resonant modes of the drum, a necessary condition for the approach in [1] to work. As briefly mentioned in the introduction, this approach utilizes the four resonant modes of the drum as the NMF basis functions. Since the modes are tonic specific, the table separates results when using the modes of the E-drum, and when using the modes of the D♯-drum. These constraints do not apply to our approach, but for fair comparison, we present separate results when the SVMs are trained on E and D♯ strokes from the same recording (the training data is identified in brackets). In both cases the NMF bases are learned in the disjoint dataset mentioned in section 4.1. Finally, the last two columns of the table report classification accuracy separately for the cross-validation performed on E strokes, and on D♯ strokes.

Classifier	NMF Bases (#)	Acc. E	Acc. D♯
HMM [1]	E modes (4)	74.95%	55.15%
HMM [1]	D♯ modes (4)	66.16%	72.61%
SVM (E)	Learned (20)	80.48%	57.49%
SVM (D♯)	Learned (20)	70.36%	82.91%

Table 1: Comparison of Results to Previous Work

The table clearly shows that the SVM approach outperforms the HMM approach in all conditions. In the ideal, but unrealistic, situation that the modes of the instrument being analyzed are known, the HMM-based system returns accuracies as high as 75%. For our approach, accuracies can be as high as 83%, which is encouraging when considering that the process of learning the NMF bases is both tonic and instrument independent. However, from the table it is also clear that a significant drop in accuracy results from classifying strokes in a tonic different from that of the resonant modes and/or the training set. This is likely due to overfitting, and is especially disappointing for our approach, since the results undermine our ability to claim that the representation is indeed tonic invariant. However, there are other variables at play. The strokes in E and D♯ were not only performed in a different tonic, but also with a different drum. Furthermore, as can be seen in Figure 3, the distribution of stroke types varies between these subsets. It is unclear from these results whether the drop in accuracy responds to any or a combination of these variations.

5.2. Tonic-Independence

To partially address these questions we evaluate our approach using the entire $\mathscr{D}$ set. We partition the set into 6 folds, such that each fold corresponds to a different tonic. Table 2 reports the classification accuracies for each fold, when the SVMs are trained using the other 5 partitions. It can be observed that the approach can reach classification accuracies as high as 75%, even when trained only with samples from other tonics (which is too strict a constraint in real-world scenarios). We believe that this is a clear indication that the approach is indeed tonic invariant.

Tonic	Stroke Instances	Accuracies
B	1326	64.40%
C	1129	75.64%
C#	1197	72.60%
D	916	65.28%
D#	1475	56.68%
E	1127	57.14%

Table 2: Number of Stroke Instances per Tonic and 1-Fold Transcription Accuracies Against Remaining Tonics

However, performance is not uniform across the different folds, with better performance for C and C♯, less so for B and D, and worse for D♯ and E. Clearly, classification benefits from having a significant number of samples of the same drum recorded under the same conditions in the training set. When classifying the D♯ and E folds, the training set contains no instances from the same instrument, and roughly $1/5^{th}$ of training instances recorded in the same conditions, thus justifying the relatively poor

performance. Indeed the small performance difference between the two cases could be attributed to the larger number of stroke instances in D♯ (thus counting for more of the training set). This is not the case for the other four tonics, where $3/4^{ths}$ of training instances are of the same drum and recording conditions, positively affecting accuracy. In addition, results seem to indicate that training with strokes of the same drum/recording in tonics one semitone apart in both directions is also beneficial, partly explaining the better performance for C and C♯.

All of this indicates that, while tonic invariant, the proposed approach is susceptible to variations of instrument and recording conditions, thus negatively affecting its usability in real-world scenarios. It is also clear that more annotated data, covering a wider range of recording conditions and instruments (not to mention performers and styles) is needed to develop a more robust system. With these caveats in mind, we now move to systematically test the different parameters of the system and their effect on overall performance.

5.3. Optimal parameterization

The following experiments report classification accuracies on the $\mathscr{D}$ dataset using 10-fold cross-validation with random partitions. In these experiments we use the default system's parameters presented in section 4.2 for all but one parameter, which is varied to study its effect on overall performance.

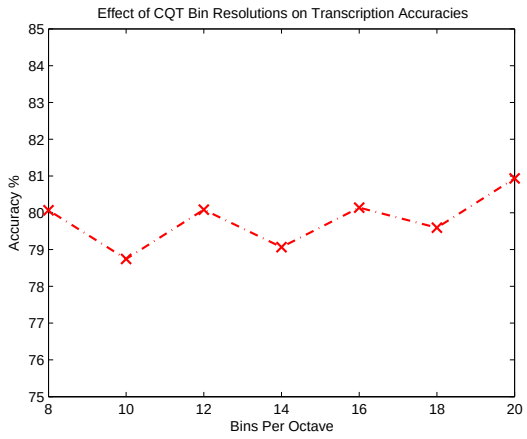


Fig. 4: Transcription accuracies when varying β between 8 and 20 bins per octave.

First, figure 4 explores the effect of varying the resolution of the CQT filter-bank by changing the number of bins per octave β between 8 and 20. It is important to note here that the fluctuation in accuracy in this plot is only about 2.5%. However, the graph does depict a subtle, but upward trend in accuracy values with a maximum value at $\beta = 20$. This β value may not necessarily be optimal but there is indication that more resolution can improve accuracies slightly. The graph shows that performance improves for $\beta \in [12, 16, 20]$. While these results do not show significant variatin in performance, it demonstrates that for mridangam percussive event recognition, relatively low frequecy resolution is sufficient for transcription.

Basis/Cutoff	NMF	PCA	Neither
20	79.47%	55.01%	78.87%

Table 3: Accuracies of Factorization Methods

Second, we validate the choice of NMF for feature learning, comparing with (1) principal component analysis (PCA) as an alternative data-driven method, and (2) bypassing the learning/factorization stage altogether, in which case the input to the SVM are the event-synchronous weighted means of the magnitude DFT. For PCA we set the number of principal components to 20, thus matching the number of NMF basis in the comparison. NMF clearly outperforms PCA ($>$ 20% difference) and slighlty outperforms the non-factorized features (0.6%), suggesting that the factorization stage can add value to the transcription process. Informal, unreported tests also show that this advantage is maintained regardless of the number of bases/components that are used, although the magnitude of the difference changes.

Third, we analyze the sensitivity of classification accuracies to the number of NMF bases, ranging from 5 to 40. Figure 5 shows that accuracy increases as the number of bases increases, with best results for 33 bases. This supports the notion that projecting into an over-complete, sparse representational space enhances discrimination, thus aiding classification. However, while performance is significantly worse when using 10 bases or less, choosing anything above 15 will mostly trend towards small, incremental improvement, hardly justifying the increase in dimensionality. At 25 bases, there is dip in accuracies which is an outlier in the trend. One noticeable characteristic of the plot is how insensitive performance seems to be to the size of the NMF dictionary for most of the range beyond 30 bases.

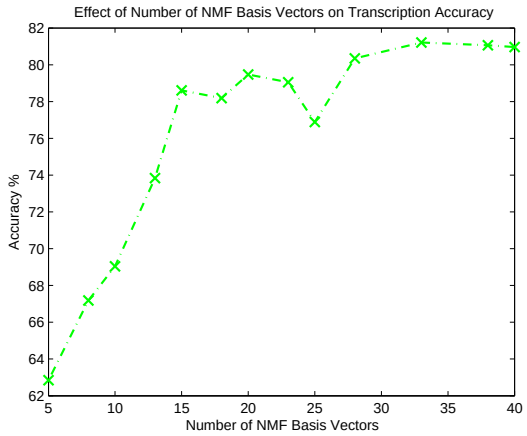


Fig. 5: Effect of number of bases on classification accuracy

Fourth, Figure 6 shows a set of boxplots characterizing the effect of the four summary statistics discussed in section 3.3. In general, it can be observed that the concatenation of statistics does improve performance. The addition of the standard deviation features always seems to improve accuracy. When using multiple features, overfitting is a possibility especially for classes that have little data (stroke *bheem*). The optimal set of features is the combination of mean, maximum, minimum and standard deviation summary statistics because together they have the highest accuracies with minimal spread.

Finally, Tables 4 and 7 show, respectively, class-wise classification accuracy and confusions for the optimal system using $\beta = 12$, NMF with 40 basis functions, and the combination of the weighted mean, maximum, minimum and standard deviation for summarization. Overall performance is at 86.65% accuracy. As is to be expected, classification is more robust for those stroke types which are best represented in the dataset, and poorer for the less populated classes. Strokes *thi* and *ta* sound very similar to the trained ear, hence they are the two most confused strokes during classification. *Dheem* has the lowest class-wise accuracy and it oddly does not get confused with its partner stroke *dhin*. However, there is confusion with the partner stroke of the only other composite stroke, *tham*. *Tham* and *dheem* could become hard to decipher when *thom* is played dominantly because its bass sound can overwhelm the composite sound. This can be confirmed because both of these composite strokes have significant confusion with *thom*. In such a scenario, the

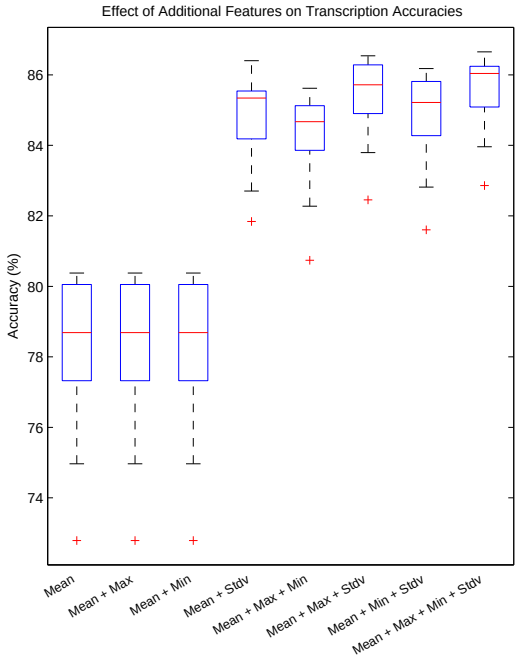


Fig. 6: Effect of summarization strategies, and their combination, on transcription accuracy.

confusion between *dheem*, *tham*, *num* and *thom* is reasonable.

6. CONCLUSION

In this paper we present and validate a tonic-independent approach for mridangam stroke transcription. We describe the transcription system in detail along with several experiments intended to compare to prior art, validate claims of tonic invariance, and assess the relative impact of the different system components on performance. Our results show that our approach indeed improves the state of the art, that features based on the magnitude spectrum of the CQT representation show invariance to pitch transpositions, that feature learning using NMF improves results marginally. Furthermore, an optimal selection of system parameters results in an increase in performance, although at times small (as in the baseline exeperiment in 5.3), at others large (when using multiple features) with accuracies above 86%. We evaluate and demonstrate the effect of most parts of the transcription system (feature extraction, factorization and summarization), but do not analyze the effect of different clas-

	Tha	**Thi**	**Thom**	**Num**	**Dhin**	**Ta**	**Cha**	**Bheem**	**Tham**	**Dheem**
Tha	**1106**	37	7	2	0	17	2	2	1	2
Thi	32	**2024**	19	6	5	107	3	0	12	4
Thom	7	7	**566**	1	0	2	0	0	51	10
Num	0	7	3	**476**	6	2	1	0	1	4
Dhin	0	17	1	6	**449**	0	4	0	4	1
Ta	4	174	3	0	0	**744**	1	0	2	1
Cha	11	79	1	1	7	4	**225**	0	0	0
Bheem	0	2	14	0	0	0	1	**35**	0	0
Tham	0	5	46	1	2	0	0	0	**392**	22
Dheem	1	10	64	17	1	3	2	0	85	**196**

Fig. 7: Stroke Confusion Matrix

Table 4: Classwise Accuracy

Class Name	**Accuracy**
Tha	94.05%
Thi	91.50%
Thom	87.89%
Num	95.20%
Dhin	93.15%
Ta	80.09%
Cha	68.60%
Bheem	67.31%
Tham	83.76%
Dheem	51.72%

sifiers as it is beyond the scope of this paper. Results also show that the system is sensitive to the specific instrument and recording conditions, and that our current dataset is too limited for the development of more robust solutions. Therefore, we are focusing our current efforts on the collection and annotation of a more extensive dataset, including data from multiple performers, styles, instruments, tonics and recording conditions. This data will allow us to investigate alternative feature design, normalization and classification strategies that could potentially overcome the shortcomings of the current system. In the longer term, we plan to extend the problem to the analysis of multi-instrumental, live recordings, scenarios which are closer to the real-world applications that motivate our work.

7. REFERENCES

[1] A. Anantapadmanabhan, A. Bellur, and H. Murthy, *Modal analysis and transcription of strokes of the mridangam using non-negative matrix factorization*, Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (2013).

[2] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, *A tutorial on onset detection in music signals*, IEE Transaction on Speech and Audio Processing **13** (2005), 1035–1047.

[3] J. C. Brown and M. S. Puckette, *An efficient algorithm for the calculation of a constant q transform*, IRCAM (1992), 109–112.

[4] C. Chang and C. Lin, *Libsvm: A library for support vector machines*, ACM Transactions on Intelligent Systems and Technology **2** (2011), 27:1–27:27, Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[5] P. Chordia, *Segmentation and recognition of tabla strokes*, in Proc. of ISMIR (2005).

[6] Battenberg et al, *Toward live drum separation using probabilistic spectral clustering based on the itakura-saito divergence*, AES 45 (2012).

[7] D. Fitzgerald, R Lawlor, and E. Coyle, *Drum transcription using automatic grouping of events and prior subspace analysis*, Digital Media Processing for Multimedia Interactive Services - Proceedings of the 4th European Workshop on Image Analysis for Multimedia Interactive Services (2003), 306–309.

[8] O. K. Gillet and G. Richard, *Automatic labelling of tabla signlas*, in Proc. of ISMIR (2003).

[9] S. Gopal, *Mridangam - an indian classical percussion drum*, B.R. Rhythms, 425, Nimri Colony Ashok Vihar, Phase-IV, New Delhi, 2004.

[10] G. Grindlay and D. P. Ellis, *Multi-voice polyphonic music transcription using eigeninstruments*, in IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (2009).

[11] P. Herrera, A. Dehamel, and F. Gouyon, *Automatic labeling of unpitched percussion sounds*, in Proc. of the Audio Engineering Society, 114th Convention (2003), 1378–1391.

[12] P. Herrera, A. Yeterian, R. Yeterian, and F. Gouyon, *Automatic classification of drum sounds: a comparison of feature selection and classification techniques*, in Proc of Second Int. Conf. on Music and Artificial Intelligence (2002), 69–80.

[13] J.Paulus and T.Virtanen, *Spectral and temporal periodicity representations of rhythm for the automatic classification of music audio signal*, 13th European Signal Processing Conference, 2005.

[14] D. D. LEE and H. S. Seung, *Algorithms for nonnegative matrix factorization*, In Neural Inf. Process. Syst (2001), 556–562.

[15] S. S. Malu and A. Siddharthan, *Acoustics of the Indian drum*, arXiv:math-ph/0001030 (2000).

[16] G. Peeters, *Spectral and temporal periodicity representations of rhythm for the automatic classification of music audio signal*, Audio, Speech, and Language Processing, IEEE Transactions **19** (2011), no. 5, 1242–1252.

[17] T. Pohle, D. Schnitzer, M. Schedl, P. Knees, and G. Widmer, *On rhythm and general music similarity*, Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR09), 2009.

[18] C. V. Raman, *The Indian musical drums*, in Proc. Ind. Acad. Sci. (1934), 179–188.

[19] V. Sandvold, F. Gouyon, and P. Herrera, *Drum sound classification in polyphonic audio recordings using localized sound models*, in Proc. of ISMIR (2004).

[20] P. Sarala and H. Murthy, *Cent filter banks and its relevance to identifying the main song in carnatic music*, Accepted for Publication in CMMR 2013 (2013).

[21] R. Siddharthan, P. Chatterjee, and V. Tripathi, *A study of harmonic overtones produced in Indian drums*, Physics Education (1994), 304–310.

[22] P. Smaragdis and J. C. Brown, *Non-negative matrix factorization of polyphonic music transcription*, in Proc. IEEE Workshop on Applicaion of Signal Processing to Audio and Acoustics (2003).

[23] B. Thierry and D. P. Ellis, *Large-scale cover song recognition using the 2d fourier transform magnitude*, Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR 2012), 2012.

[24] A. Tindale, A. Kapur, G. Tzanetakis, and I. Fujinaga, *Retrieval of percussion gestures using timbre classification techniques*, in Proc. of ISMIR (2004), 541–545.

[25] E. Vincent, N. Berlin, and R. Badeau, *Harmonic and inharmonic nonnegative matrix factorization for polyphonic pitch transcription*, IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP (2008), 109–112.

[26] T. Virtanen and A. Klapuri, *Analysis of polyphonic audio using source-flter model and non-negative matrix factorization*, in Advances in Neural Inf. Process. Syst (2006).