AUTOMATICALLY IDENTIFYING VOCAL EXPRESSIONS FOR MUSIC TRANSCRIPTION

Sai Sumanth Miryala Microsoft Research India mssumanth99@gmail.com

Kalika Bali Microsoft Research India kalikab@microsoft.com

Ranjita Bhagwan Microsoft Research India bhagwan@microsoft.com

Monojit Choudhury Microsoft Research India monojitc@microsoft.com

ABSTRACT

Music transcription has many uses ranging from music information retrieval to better education tools. An important component of automated transcription is the identification and labeling of different kinds of vocal expressions such as vibrato, glides, and riffs. In Indian Classical Music such expressions are particularly important since a *raga* is often established and identified by the correct use of these expressions. It is not only important to classify *what* the expression is, but also *when* it starts and ends in a vocal rendition. Some examples of such expressions that are key to Indian music are *Meend* (vocal glides) and *Andolan* (very slow vibrato).

In this paper, we present an algorithm for the automatic transcription and expression identification of vocal renditions with specific application to North Indian Classical Music. Using expert human annotation as the ground truth, we evaluate this algorithm and compare it with two machine-learning approaches. Our results show that we correctly identify the expressions and transcribe vocal music with 85% accuracy. As a part of this effort, we have created a corpus of 35 voice recordings, of which 12 recordings are annotated by experts. The corpus is available for download [1] .

1. INTRODUCTION

Vocal expressions, such as glides, licks, and vibrato are an intrinsic part of vocal music of any genre. The use of suitable vocal expressions establishes the characteristic mood of a song and enhances its emotional appeal. In western classical music, the appropriate use of vibrato and tremolo while singing can drastically change the appeal of a given piece.

Similarly, in North Indian Classical Music (NICM), not only do vocal expressions enhance or characterize a song's mood, they also establish the correctness of a *raga*'s [2] rendition. For instance, the nature of the glide between two notes is a key difference between two *ragas*, *Bhupali* and *Shuddha Kalyan*, that are based on the same melodic scale (see Figure **??**). Whether the glide from the third to the fifth note is quick or slow differentiates them (see Table **??** for note definitions). Also, whether a note is steady or oscillating can also depend on the *raga* (see Figure 2).

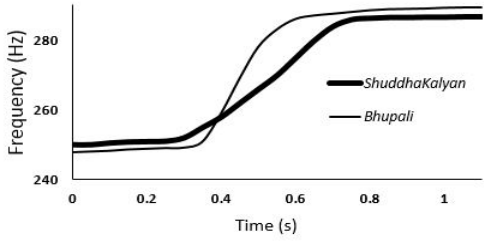

Figure 1. The glide from the third to the fifth note for *raga Bhupali* is quick, but in *Shuddha Kalyan*, it is slower and perceivably touches the fourth's microtones.

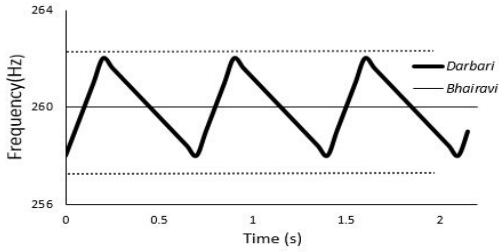

Figure 2. The third note is steady in *raga Bhairavi*, but oscillates across the note's microtones in *raga Darbari*

Hence, automatically identifying and labeling such vocal expressions is very important for accurate music transcription. In addition, identifying vocal expressions is a basic requirement towards building tools for music education. Such an educational tool can process a student's rendition, identify the vocal expression and provide feedback, visual or auditory, on the correctness.

In this paper, we propose an algorithm for automatically identifying vocal expressions and can therefore be a building-block for various music transcription and education tools. The proposed algorithm (a) estimates the pitch curve, (b) identifies the singing voice frames, (c) processes the pitch envelope to obtain a canonical representation and

[1] www.google.com

[2] A *raga* is based on an ascending and descending scale, but is characterized using many other features and evades a formal definition. See [**?**] for a detailed exposition.

(d) uses *templates* to identify each expression and finally create a transcription of the audio signal.

We concentrate on two main expressions predominantly used in NICM: *Meend* or a slow glide between notes, and *Andolan* or slow microtonal oscillations within the same note. We limit our discussion in this paper to these because they are two of the most important vocal expressions that determine the correctness of a *raga*. While we do not claim that our specific algorithm can capture every kind of vocal expression across all genres, we believe that the overall approach we have proposed can be used to identify different styles of vocal expressions across different genres.

The ground-truth is created by manual annotation of the recordings by experts. We compare the results of our work with two different machine learning techniques: decision trees and conditional random fields. Our findings show that we can achieve up to 85% accuracy across all vocal expressions identified, an improvement of 7% over a machine learning-based competitive baseline.

2. RELATED WORK

The automatic transcription of polyphonic music needs to deal with source separation and note detection in the primary source. Rao et al. [**?**] proposed a system for extracting vocal melody from a polyphonic music signal in NICM using a spectral harmonic-matching pitch detection algorithm. The voice signal extracted using these methods can then be input to our algorithm for identifying expressions. However this method does not apply to identifying fine vocal expressions.

Typically, audio alignment for note detection, specifically onset, steady period and offset of a note, employ signal processing methods like Dynamic Time Warping (DTW) in conjunction with graphical models like Hidden Markov Models (HMM). Devaney et al. [**?**] used an HMM model with acoustic features like power and aperiodicity along with DTW priors to align as well as identify the transient as well as steady portions of a note. Our technique does not use onset detection for reasons outlined in Section **??**.

Classification of *ragas* based on the concept of Pitch Class Distribution Dyads (PCDD) [**?**] uses spectrum based pitch extraction and onset-detection to create Pitch Class Distribution (PCD) and PCDD. A classifier is then used on the PCD and PCDDs to identify *raga* labels. Ross et al. [**?**] detects melodic motifs to identify repetition of phrases in a *raga* rendition. While all these methods are successful to a certain extent, they do not take into account the vocal expressions that may be specific to the composition or introduced by the singer for a more rich rendition, or in the music education scenarios, mistakes by a learner.

The biggest challenge for an automatic transcription of NICM is the absence of a written score that makes any top-down processing using the score as a knowledge source practically impossible. A number of approaches for the transcription of Western music [**?**] have made use of the availability of a score as one of the knowledge sources in their models. Klapuri [**?**], has an extensive discussion on the use of a "musicological model" as a part of a transcription system. While the system developed in [?] primarily uses the sinusoidal properties of the polyphonic music signal, the concluding discussion clearly points to the use of an existing score for improvement. In a later version of the polyphonic music transcription [**?**], they make use of a reference database for note event modeling, as well as a musicological model for the transcription system. The work reported in this paper makes no assumptions about the availability of a musical score or a knowledge model and aims to identify and label the different vocal expressions from the signal using signal processing and machine learning techniques.

Sa	*Re*	*Ga*	*Ma*	*Pa*	*Dha*	*Ni*
Do	*Re*	*Mi*	*Fa*	*So*	*La*	*Ti*
1^{st}	2^{nd}	3^{rd}	4^{rd}	5^{rd}	6^{rd}	7^{rd}

Table 1. Relative note names in Indian (top) and Western (bottom) traditions .

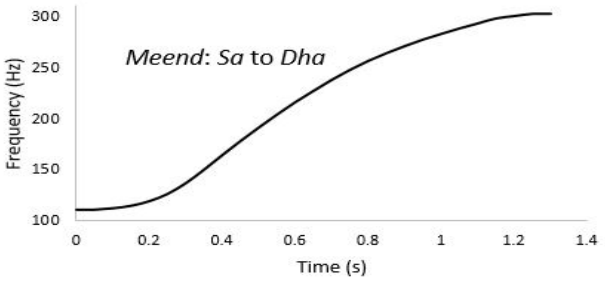

Figure 3. Pitch Envelope of a *meend* from *Sa* to *Dha* in *raga Darbari*, *Sa* is 220Hz.

3. BACKGROUND AND DEFINITIONS

NICM follows a relative note system, where all the notes are sung with respect to the *tonic* which is called *Sa* (same as the *Do*). The frequency of the tonic depends on the singer. Table **??** shows the Indian names for the seven notes corresponding to the western *Do-Re-Mi*. In this paper, we focus on the *Alap*, which is a meterless form of singing that typically starts any NICM rendition. The *alap* captures the essence of the *raga* being rendered. *Alap* is usually sung with no accompaniment except for a background drone called the *Tanpura*, which provides the singer a reference to the tonic.

3.1 Vocal Expressions

As mentioned earlier, the NICM genre uses various characteristic vocal expressions for enhancing as well as establishing a raga rendition In our work, we concentrate specifically on the *meend* and the *andolan*.

Meend is a smooth glide from one note to another, as shown in Figure **??**, where the singer moves from *Sa* to *Dha*. This is clearly distinct from a steady note, as shown in Figure **??** (top). A very short glide is often termed as a *sparsh*. For the purpose of this work, we will use the term glide to refer to both of these.

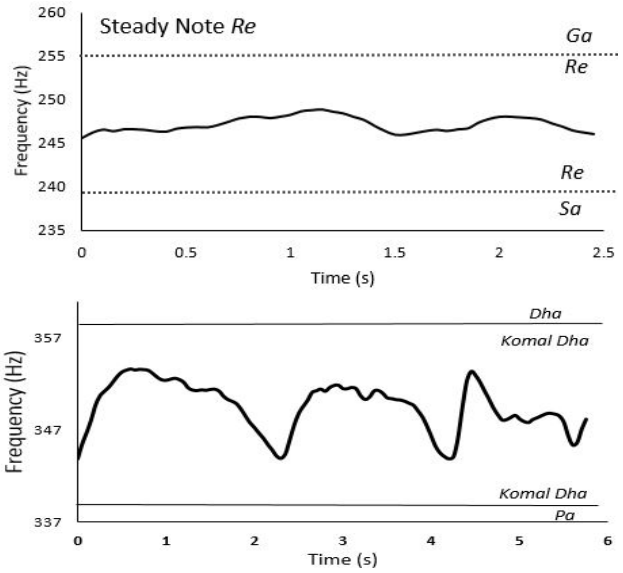

Figure 4. Pitch Envelope of a steady *Re* (top) and *andolan* around *komal Dha* (bottom) in *raga Darbari*.

Andolan is a gentle swing that oscillates between the lower and the higher microtones of a certain note. Fig. **??** (bottom) shows the pitch curve of an *andolan* sung around *komal Dha* or the minor sixth.

3.2 Problem Definition

Given an audio recording of vocal music in NICM, we want to label it with a set of annotations that clearly mark the steady notes as well as the vocal expressions, viz. *meend*, *sparsh* and *andolan*. In the example in Figure **??**. From 0 to 0.6s, a steady *Sa* (or the first note) is sung, followed by a *meend* to the *Re* or the major second, until time 1.2s. This is followed by a steady rendition of *Re* until time 1.6s. After a *meend* to *komal Ga* or the minor third, the singer sings an *andolan* around *komal Ga*. This extends from 2.2s to 3.9s. Given a vocal rendition in NICM, our objective is to output the time-series of such annotations.

3.3 Challenges

The task of identifying these vocal expressions and transcribing NICM faces two primary challenges. First, there is no written score available. Indian classical music is an improvisational art-form, and textual representation of musical pieces, if they exist are very rough and used only as a tentative guide. There are no equivalent notations for vocal expressions like trills, vibrato, or tremolo, that exist quite extensively in western classical music.

Second, in Indian classical music notes generally do not correspond to clear onsets. Hence, conventional transcription methods that rely on onset detection cannot be used. Onset detection algorithms depend on detecting transient regions in the signal, including sudden bursts in energy or changes in the spectrum of the signal etc. These methods fail whenever there is a smooth glide between two notes. In this work, we present a transcription scheme, which relies solely on the pitch envelope.

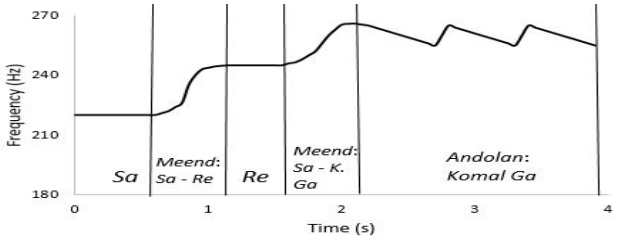

Figure 5. Annotation of an audio recording, using a mock pitch envelope.

4. TRANSCRIPTION AND EXPRESSION IDENTIFICATION

The first step towards transcription is the estimation of fundamental frequency. We chose a difference function based method for this purpose. This method was preferred over the frequency-domain methods, because real-time performances are important for music education tools. No significant improvement was observed by using the spectral methods. Any accurate pitch detection method maybe used for this step.

The second step is to detect audio segments with vocal singing as against segments with only the background drone. From the background drone, we detect the tonic, or the absolute frequency of the base-note for the singer. The third step is to obtain a canonical representation of the pitch curve, for which we use a line fitting algorithm to the curve. The final step is to perform vocal expression identification using template representations for each expression. These steps are explained in the following sections.

4.1 Pitch Estimation

In this work the terms pitch and fundamental frequency are used interchangeably. For each frame, the difference function is evaluated by subtracting the original signal with delayed versions of itself. The fundamental frequency corresponds to the absolute minimum of the difference function. The other local minima correspond to the harmonics. We fixed the frame size as 50 ms with 50% overlap for this work. A low-pass filter with a cutoff frequency of 700Hz is applied before pitch estimation to remove high frequency content. Variance of the pitch track is chosen as a measure to eliminate octave errors using the Viterbi algorithm. In this work, source separation or multi-band pitch estimation is not necessary as the singing voice masks the *tanpura* sound well.

4.2 Drone and Tonic Detection

The background drone or the *tanpura* is a stringed instrument that provides a continuous pitch reference to a vocal performer. The drone usually consists of four strings, three of them at the tonic of the scale, and one string tuned to the fifth note. To identify the singing voice frames in the recording, we use resonance properties of the drone. Due to the special form of the bridge fixed to a resonant body, *tanpura* shows remarkably different acoustic properties compared to other stringed instruments [**?**]. The wide

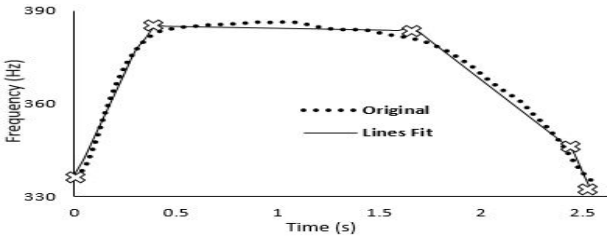

Figure 6. Pitch envelope & Lines fit using the proposed method. Critical points are marked with 'X'

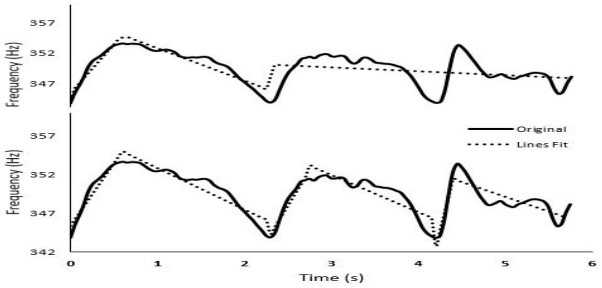

Figure 7. Pitch envelope of the *andolan* note & Lines fit using threshold JND1 (top) & JND1/2 (bottom).

body of the bridge induces a large number of overtones that manifest in the output of the pitch estimation algorithm. In frames that contain voice, these overtones are masked by the voice.

Consequently, the frames with only the drone have higher entropy than the frames that contain voice. We therefore use an entropy based method [**?**] to differentiate the singing voice frames from the drone. For each audio recording, we dynamically estimate an entropy threshold from the histogram of entropy values. Any frame with lower entropy than the threshold is labeled as a singing voice frame, while the frames with higher entropy are labeled as the *tanpura* frames. Tonic is calculated as the mode of the pitch values in the frames where *tanpura* is prominently audible.

4.3 Line Fitting

The third step in the process, is to obtain a canonical representation of the pitch-curve in terms of straight lines and *critical points* of inflection. We use an augmented version of a previously proposed line-fitting algorithm [**?**] for this purpose. We outline our algorithm as follows:

Step 1: Identify the local minima and maxima in the pitch curve. This gives us a set of points representing the curve. To achieve better fit along transient regions, the set of points where the singer is changing notes are added to the list. These points are identified by scaling down the pitch values to one octave and mapping the frequencies to notes. Start a sweep from the left of the curve.

Step 2: Start from the first point on the curve, and connect it to the third point using a straight line. From this line, if the second point lies within the distance specified by Just Noticeable Difference (JND) threshold (equation **??**), the second point is removed. Then, connect the first point to the fourth point and repeat the JND threshold-based check for the third point. Repeat this process until you find a point that lies outside the JND threshold. This point is the starting point for the next iteration.

Step 3: Starting from the new critical point, repeat Step 2 to find the next critical point. Continue this process until the whole pitch curve is represented by a set of critical points and fit lines between these points, by minimizing the squared error between the pitch curve and the lines fit.

$$JND1(F) = 3.13 - \frac{0.13 * F}{100} \quad (1)$$

Figure **??** shows a sample pitch envelope and the final critical points identified. For each pitch curve, we calculate canonical representations by varying the value of the JND threshold. Small variations in pitch due to singing imperfections are eliminated in the canonical representation. These representations are useful in the identification of vocal expressions of certain types, as we shall describe in the next subsection.

4.4 Identifying Vocal Expressions

Given the canonical representation of the pitch curve, we use *templates* for each kind of vocal expression to recognize and classify them. A template for each expression is a loose representation based on some subset of duration, line lengths, slopes, and number of points. In the next subsections, we describe templates for *andolan* and *meend*, and how we detect these expressions.

4.4.1 Andolan

Using the expert manual annotations and by studying the pitch curves ourselves, we have found that an *andolan* has the following template: six to ten straight lines, with consecutive lines having alternating slope signs. All pitch values that the lines touch *should be within the same or adjacent notes*. This template captures the slow oscillations of an *andolan*, that touch the microtones within a single note. This is as opposed to a vibrato, which manifests itself as a much faster oscillation. However, we could use a similar template for vibrato detection as well.

To match this template, we look for such a pattern across the different canonical representations of the pitch curve that are obtained by decreasing the JND threshold value iteratively a maximum of 3 times. In our work, we have used the thresholds $JND1$, $JND1/2$ and $0.4 * JND1$. The threshold needs to be decreased, because the amplitude of the oscillation can vary from very small to quite large. With large JND thresholds, the canonical representation may not capture all the oscillations, as shown in Figure **??**.

However, if the threshold is too low, the oscillatory pattern may be found in steady notes too. So, the threshold should not be decreased too much. If we find such a definite pattern in at least one of the canonical representations, we classify the corresponding segment of the pitch curve as an *andolan*. This matching algorithm is similar to using DTW iteratively to do the template matching.

4.4.2 Glides

The template for a glide is the following: a line between two critical points where the line starts and ends at *different* notes. Any line which satisfies this property is either a *meend* or *sparsh*. If the glide segment is longer than 300 ms, it is labeled a *meend*, else a *sparsh*.

4.4.3 Steady notes

Segments with no expressions are typically represented as horizontal lines, with very low slope values. Hence, we use this simple template to classify segments as steady notes. Steady notes are transcribed using the median of the pitch values between its two end points.

5. EVALUATION & RESULTS

In this Section, we describe our evaluation of the proposed algorithm. First, we describe and characterize the data we collect for evaluation. Next, we compare the accuracy of our technique with that of two machine-learning techniques – the C5.0 Decision Tree Classifier [**?**], and a Conditional Random Field classifier (CRF) [**?**] and present the results.

5.1 Data Collection

We have created a corpus of 35 recordings in 8 *ragas* sung by 6 singers of varying expertise which are publicly available for the purposes of Music Information Retrieval research. In this paper, we use 12 recordings of 3 singers singing *alap* in 4 *ragas* for evaluation of our algorithms. We ensured that all the three singers sang identical pieces, with the same set of notes and same vocal expressions in each *raga*. This is to ensure that we have a balanced dataset across different *ragas* and different singers.

We asked two experts, one who has been a professional music teacher for 25 years, and the other a serious music learner for 11 years, to manually annotate the vocal expressions on 12 recordings sung by 3 singers in 4 different *ragas*. We had one expert annotate each file first, and the second expert revised and verified these annotations to ensure no expressions were missed. In case of an ambiguity among the two annotators, the more experienced annotator's labels are used. Each file is approximately 2.5 minutes long, and the sum total of the length of all twelve recordings is approximately 30 minutes. The audio was collected in a recording studio and is therefore comparatively noiseless.

The experts used Praat [**?**] to annotate these recordings. Praat allows a user to listen to and annotate audio files, while also displaying the pitch envelope. Using Praat textgrids, the experts annotated each file with note and expression boundaries, and they labeled each segment as either *tanpura*, Steady, *andolan*, *meend*, or *sparsh*. Annotating a 3 minute recording took the two experts 120 minutes on average. Therefore, a total of about 24 hours were required for the manual annotation process.

Feature	No. of Features
Pitch	2n+1
First derivative of Pitch	2n
Pitch (warped to one octave)	2n+1
Entropy	2n+1
Amplitude (Normalized)	2n+1

Table 2. Features for the classifiers.

	Proposed	DT	CRF	Improvement(%)
Drone	0.964	0.966	0.956	-0.18
Steady	0.828	0.825	0.828	5.4
Andolan	0.647	0.449	0.448	44.31
Meend	0.72	0.38	0.314	89.49
Sparsh	0.651	0.295	0.344	89.31

Table 3. F1-scores for each class. The last column shows the percentage improvement that our approach shows over the better classifier

5.2 Evaluation Methodology

The annotations of the algorithms are compared with the ground-truth, frame to frame and the overall accuracy is defined as the percentage of frames labeled correctly. We compare the classification of these frames by our algorithm with that of two stock classifiers: the C5.0 decision tree, and CRF. The reason for trying the CRF is to evaluate a classifier which uses a notion of time or sequences, which seems inherent to expressions such as the *andolan*. The decision tree, on the other hand, does not incorporate time. The weights vector for CRF is initialized randomly and each note is considered as a sequence. These methods are evaluated using the leave one out cross-validation method.

The features used for classification are shown in table **??**. To provide the note change information, pitch is warped down to one octave and fed to the classifier. We collect these features for the current frame, and a fixed number (n) of frames before and after the current frame. The performance of the classifiers is similar across several values of n. In this section, we report the results for $n = 10$.

5.3 Results

The overall accuracy of our technique is 84.7%, whereas with CRF, it is 77.6% and with C5.0, it is 77%. Hence, our proposed method improves the error produced by the better classifier by 31.7%.

Table **??** shows the F1-scores for each class, for each of the evaluated techniques. All the three approaches identify the Drone segments with about 96% accuracy. Our approach shows a 5.4% improvement over the better of the two classifiers for steady notes. For the more complex vocal expressions, our approach shows much higher improve-

Algorithm	Singer 1	Singer 2	Singer 3
Proposed	15.27	13.63	16.8
DT	21.25	21.96	25.44
CRF	18.69	20.72	21.05

Table 4. singer-wise classification errors (in % of frames)

Algorithm	*Bhairav*	*Darbari*	*Janpuri*	*Lalit*
Proposed	14.49	17.02	20.97	10.65
DT	24.77	29.98	20.69	17.45
CRF	20.43	24.82	22.61	12.75

Table 5. *raga*-wise classification errors (in % of frames)

ment: 44.31% for *andolan*, and about 89% for the glides.

Note that this is in spite of the machine learning methods using approximately 90% of the available frames as training data. Our algorithm, on the other hand, does not use any training data. Moreover, we have no tunable thresholds in the core algorithm. We do use fixed thresholds for certain templates, for instance, to differentiate *meends* from short glides (*sparsh*). However, given the nature of these expressions, we feel this is unavoidable.

However, for all three evaluated techniques, the F1-scores for identifying the vocal expressions are much lower than those for identifying steady notes. For instance, the F1-score for *andolan* using our approach is 0.647, for *meend* it is 0.72, whereas for Steady is 0.828. One source of error is that the boundaries between the glides and steady notes, as annotated by the experts, do not align exactly with the algorithm's labels. Therefore, some frames in these boundary regions, which the expert has annotated as glides are very often mis-labeled by our approach as steady. Another source of error is the mis-labeling of vocal expressions by the annotators. Some of the short vocal glides are hard to perceive and are labeled 'steady' by the annotators. In case of *andolan*, if the range of oscillation is less, the algorithms would identify it as a steady note and sometimes the pitch estimation algorithm does not pick up the microtonal variation accurately enough. Also, the way in which these expressions are achieved sometimes depends on the *raga* and the singer's expertise.

Table **??** shows the classification error by singer. Singer 1 is a professional artiste with 30 years of training, Singer 2 is a music educationist with 15 years of training, and Singer 3 is a music educationist with 40 years of training. Singer 3 uses much smaller microtonal variations in the rendition of *andolans*, some of which are labeled as steady. Hence, the errors are slightly higher for Singer 3 across all three approaches as compared to Singers 1 and 2.

Table **??** shows the classification error by *raga*. Of the four *ragas*, only *Lalit* does not use *andolans*. *Janpuri* and *Darbari*, on the other hand, use significant amounts of this vocal expression. Hence, the error associated with *Lalit* is a lot less(10.65% using our approach), and that associated with *Jaunpuri* (20.97%) and *Darbari*(17.02%) are higher.

6. CONCLUSIONS

We proposed an algorithm for automatic expression identification in vocal music. The idea is to use templates for each expression and match these templates in the pitch curve. We compared the performance of our algorithm with two machine learning methods. Our algorithm is more accurate than the better classifier by about 7%.

In future work, we intend to apply this technique to more vocal expressions across different genres of music. We also intend to use this algorithm as a building-block in various music education and music transcription applications.

7. REFERENCES

[1] Bret Battey: "Bézier spline modeling of pitch-continuous melodic expression and ornamentation," *Computer Music Journal*, Vol. 28(4), pp. 25-39, 2004.

[2] Bhatkhande V. N, Garg P.K.: *Hindustani Sangit Paddhati*, Sakhi Prakashan, 1990.

[3] Chordia P.: "Automatic raag classification of pitch tracked performances using pitch-class and pitch-class dyad distributions," *Proc. of Intl. Computer Music Conf.*, 2006.

[4] Devaney J. et al.: "Improving MIDI-audio alignment with acoustic features," *In Proc. IEEE WASPAA*, 2009.

[5] Jia C., Bo Xu: "An improved entropy-based endpoint detection algorithm," *International Symposium on Chinese Spoken Language Processing*, 2002.

[6] Klapuri A., Ryynänen, M.P.: "Modeling of note events for singing transcription.", *ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing*, 2004.

[7] Klapuri A., Ryynänen, M.P.: "Transcription of the singing melody in polyphonic music.", *Proc. 7th Intl. Conf. on Music Information Retrieval*, Vol. 15, 2006.

[8] Lafferty J. et al.: "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," *Intl. Conf. on Machine Learning*, 2001.

[9] Martin K.D.: "Automatic transcription of simple polyphonic music: robust front end processing," *in the Third Joint Meeting of the Acoustical Societies of America and Japan*, 1996.

[10] J.R. Quinlan: "Induction of decision trees", *Machine Learning*, Vol. 1.1, pp. 81-106.

[11] Raman C.V.: "On some Indian stringed instruments," *Proceedings of the Indian Association for the Cultivation of Science*, Vol. 7, pp. 29–33, 1921.

[12] Rao V., Rao P.: "Vocal melody extraction in the presence of pitched accompaniment in polyphonic music.", *IEEE Trans. on Audio, Speech, and Language Processing*, Vol 18(8), pp. 2145-2154, 2010.

[13] Ross J.C., Vinutha T.P., Rao P.: "Detecting melodic motifs from audio for Hindustani classical music.", *Proceedings of the 13th Intl. Society for Music Info. Retrieval Conf.*, 2012.

[14] The PRAAT Speech Analysis and Annotation Toolkit: http://www.fon.hum.uva.nl/praat/.