Designing Automatic Note Transcription System for Hindustani Classical Music

Prasenjit Dhara, Pradeep Rengaswamy, K. Sreenivasa Rao
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur, India
Email: {prasenjitdhara.pd,pradeeprengaswamy}@gmail.com, ksrao@iitkgp.ac.in

Abstract—**Hindustani music is heterophonic with lead voice accompanied by instruments. A trained Hindustani musician is capable of perceiving the notes based on the lead voice but a novice person is unable to decode the notes. This necessitates the development of an automated note transcription system. The automatic system detects and generates the notes present in the music file. In this work, the melody contour is extracted from the audio file using salience based method. The extracted melody values are normalized in cent scale. The notes are having a fixed melody value in cent scale. Each of the melody values from the extracted melody contour is compared with the melody value of a note. If the extracted melody value matches within a given range of tolerance of melody value of a note then this note is assigned. The successive similar notes for each melody value of the contour are merged together to form a single note preserving the start and end time. In the method, some notes are also detected during transition of melody contour from one note to another. These notes are termed as transitions notes which are not perceived and not desired. The durations of these notes are very less. These notes are eliminated by using duration thresholding. So, the tolerance of melody value and threshold duration of the note plays an important role in the accuracy of the transcription system. These parameters are optimized to maximize the accuracy of the system. The performance of the system is evaluated using two metrics. The results show the note transcription system performs satisfactorily.**

***K*eywords —** Hindustani classical music, heterophonic music, vocal, notes, melody, salience, transcription

I. Introduction

Hindustani classical music is in *raga* and *tala* format. The term *raga* describes the set of permitted *swaras* or notes and characteristic phrases, and *tala* defines the rhythm. The combination of notes and its ornamentation characterizes *raga* [1]. It is melodic in nature as notes are played in succession, unlike in Western Music which is harmonic in nature, where different notes are played simultaneously [2].The melodic form of Hindustani classical music seems to be simple but, it is highly evolved and sophisticated [3].

The seven *swaras* or notes are Shadja(Sa), Rishabh(Re), Gandhãr(Ga), Madhyam(Ma), Panchama(Pa), Dhaivat(Dha) and Nishãd(Ni). Out of these seven notes, Shadja and Panchama have no variation and are called *achala* or immovable note. However, the rest of the notes have microtonal variation, also called *vikrit* form. Rishabh, Gandhãr, Dhaivat and Nishãd have *komal* or flat version, moved below their natural place and only Madhyam has *tĩvra* or sharp version, higher than the natural one [2]. Taking these variations into account, there are twelve notes in Hindustani classical music. The frequency values of these notes are not fixed rather relative to tonic, chosen according to singer's comfort [4]. But, in Western music the frequency values of the notes are fixed.

During rendition, the notes are played in succession. A listener having a background in music can perceive the notes and other ornamentation, but the laymen cannot [4]. Therefore, transcription of these files can prove to be very helpful for interested laypersons. Indian classical music has a huge database. Manual transcription of this huge database is very cumbersome. So, we need to develop a system which transcribes the notes of a music file to help the naive person. This is the motivation of our work. The ornamentation like *meend*, *andolan*, *kaun-swara*, etc present in Indian classical music makes transcription of music files very difficult compared to Western music. Our challenge is to overcome this inherent complexity in Indian classical music and build an automatic transcription system for Hindustani classical music.

A. Swaras And Their Relation

The tonic (the note 'sa') is the base of Indian classical music and all other notes are defined based on this tonic, i.e relative to tonic. The ratio between different notes to the tonic is given in the following table I [2].

TABLE I
Swaras and Their Ratio

Symbol	Ratio to tonic	Hindustani Name
sa	1	Shadja
re	256/243	Komal Rishabh
Re	9/8	Shuddha Rishabh
ga	32/27	Komal Gandhãr
Ga	5/4	Shuddha Gandhãr
ma	4/3	Madhyam
Ma	45/32	Tĩvra Madhyam
Pa	3/2	Pancham
dha	128/81	Komal Dhaivat
Dha	5/3	Shuddha Dhaivat
ni	16/9	Komal Nishãd
Ni	15/8	Shuddha Nishãd
Ṡa	2	Shadja

978-1-5090-2029-4/16/$31.00 @2016 IEEE

II. Related Works

The transcription of music is defined as the sequence of notes present in the music file in textual form. It is a very challenging task for Indian classical music due to its melodic variation. Pandey et al. [5] developed a system called "Tansen". This system identifies *pakad* to determine *raga* using HMM classifier and proposed two heuristics, namely, Hill Peak Heuristic and Note Duration Heuristic for *swara* detection, but did not give any quantitative measure of the method and used two melody contours of different window lengths. Shetty et al. [6] developed a neural network based system that detects the note pattern to identify the *raga* of the monophonic music signal. Dighe et al. [7] used the *vadi* or strong note concept of Indian Classical music and computed *swara* histograms and used these histograms as features in random forest classifier to identify the *raga*. Sridhar et al. [8] identified segments based on *tala* of the Carnatic music and applied autocorrelation method for identification of the *swara*, Sa. We also propose some metrics to quantify the performance of our note transcription system.

III. Automatic Note Transcription

A. Database and Labeling

The audio file corpus is collected from professional artists. The corpus contains twenty-seven audio files with eight different *ragas* considered for this study. The duration of each audio file is approximately $50 - 60$ sec. These audio files are heterophonic with one singing voice, and the commonly accompanying instruments are tanpura, harmonium, and tabla, etc. For preprocessing, the stereo audio signals are converted into 44100 Hz mono signal at 32 bits/sample. The notes and other characteristics of raga are manually labeled by a professional musician and then cross-validated by another musician. The individual note boundaries are marked based on human perception. At times during ornamentation of a *raga*, the individual note boundary marking is quite tedious because of the duration it sustains. Those notes are marked together as a phrase.

B. Melody Extraction

Indian classical music is heterophonic in nature [9]. The singer's voice dominates the other accompanying pitched instruments like tanpura, and percussion instruments like tabla. A superposition of all these gives us a composite signal. We extract the singer's melody(pitch) values from this composite signal. The definition of melody is proposed by [10] "... *the single (monophonic) pitch sequence that a listener might produce if asked to whistle or hum a piece of polyphonic music, and that is listener would recognize as being the 'essence' of that music when heard in comparison*". Melody is extracted using state of the art salience-based predominant melody extraction method. These methods exploit the local salience of the melodic pitch as well as smoothness and continuity over time to provide a pitch estimate and voicing decision. In this method, the melody values are computed for every 46.4 ms window with a frame shift of 2.9 ms [11] .

C. Melody Contour Smoothing and Normalization

The extracted melody contour has some spurious peaks and random variation due to the melody extraction algorithm. The melody contour is passed through a seven point averaging filter to eliminate these types of perturbations.

Each octave is separated by 1200 cents in the cent scale. So, the smoothed melody contour is mapped into cent scale for better resolution using the following relation:

$$f_{cent} = 1200 * log_2 \left(\frac{f}{f_0} \right) \tag{1}$$

where f is the extracted melody and f_0 is the reference melody value. In this case, it is the tonic of the music and given manually. Here zero and twelve hundred cent values correspond to tonic and higher $\dot{S}a$ and each note is separated by almost 100 cents. The value of the notes in cent scale is given below.

TABLE II
Detail of the notes (swaras) with note melody values in Cent Scale

Symbol	Ratio to tonic	Notes Melody value (in Cent)	Hindustani Name
sa	1	0	Shadja
re	256/243	111.7312	Komal Rishabh
Re	9/8	203.91	Shuddha Rishabh
ga	32/27	315.64	Komal Gandhãr
Ga	5/4	386.31	Shuddha Gandhãr
ma	4/3	498.04	Madhyam
Ma	45/32	609.77	Tĩvra Mad-hyam
Pa	3/2	701.955	Pancham
dha	128/81	813.68	Komal Dhaivat
Dha	5/3	884.35	Shuddha Dhaivat
ni	16/9	996.08	Komal Nishãd
Ni	15/8	1088.26	Shuddha Nishãd
Ṡa	2	1200	Shadja

D. Note Detection

A singer may or may not sing exactly at particular frequency of a note but sings at frequencies around the designated frequency of the note. This deviation of frequency values from the ideal value makes the note identification problem all the more challenging.

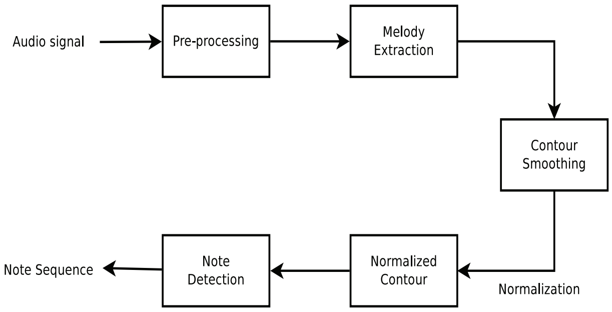

Fig. 1. Framework of the automatic note transcription system

To detect the notes during a rendition, each melody value is compared with the nearest specified value in the table, x and if this deviation is within some threshold value δ i.e., $x \pm \delta$ then it is considered as note otherwise not. We define this range in which we consider a melody value to constitute a note as *tolerance band*. If the extracted melody value falls within this range, we map this melody value to its corresponding note. Melody values which fall outside this range are marked as transitions. In this way, a sub-segmental analysis of the given music segment is performed. The similar notes in consecutive frames are combined to form a single note with starting frame as the *start time* and ending frame as *end time* of the note. When a singer moves from one note to another, some intermediate notes are populated for a very short duration, those regions are considered to be transition periods. These transitions are eliminated by means of duration thresholding based methods. This value we termed as T_d. The appropriate values of δ and T_d are discussed in section V.

IV. Evaluation Metrics

In the previous section, we proposed a method of detecting notes. These detected notes are to be checked against the manual transcription. If the detected note is present within the boundary mentioned in manual transcription, then it is marked as 1, otherwise, it is marked as 0. We have termed the notes marked as 1 as *true note* and the other note as *false note*. We proposed two metrics to measure the performance. Depending upon the thresholding values, the performance measured by these proposed metrics vary. Our aim is to maximize the performance as measured by both these metrics.

A. Performance in Terms of Manual Transcription

We calculated the number of notes correctly detected out of the total number of notes present in the labeled transcription. We proposed a performance metric based on this information using the following relation:

$$MT = \left(\frac{N_{mc}}{N_m}\right) \times 100 \tag{2}$$

where N_m is the total number of labeled notes in the manual transcription and N_{mc} is the number of notes in the manual transcription that are correctly detected by our method.

B. Performance in Terms of Automatic Transcription

In this metric, we calculated the number of notes detected in Automatic Transcription, out of which how many of them correctly appeared in comparison with manual transcription. In equation the performance can be written as

$$AT = \left(\frac{N_{ac}}{N_a}\right) \times 100 \tag{3}$$

where N_a is the total number of notes detected in automatic transcription and N_{ac} is total number of *true notes*. It is important to note that the value of N_{ac} may or may not be same as the value of N_{mc} due to deviation of the melody contour from the designated melody value of a note, where many notes are detected against a single label in manual transcription.

V. Parameter Optimization

In section III-D, *melody tolerance* (δ) and *threshold duration* (T_d) were the parameters of the note transcription system. The significance of these parameters is given below.

A. Melody Tolerance

In the Hindustani classical music, the melody contour fluctuates very much. The singer cannot produce the exact melody value of a note but produces a value in the vicinity(few cents apart) of that note. The value of δ plays an important role in note detection. If the value of δ is very small, then less number of melody values will fall within the range of $x \pm \delta$ where x denotes the value of a note in the cent scale. In effect, less number of notes are detected and sometimes a large number of notes, marked in manual transcription get eliminated as transitions. On the other case if the value of δ is very high, then melody values which are far apart from the value of a note will also populate as note. In effect, a large number of melody values belong to the transition are considered as a note. So, the number of falsely detected notes gets increased. We have to find out the optimum value of δ to get the best balance between missing out notes and detecting false notes.

B. Threshold Duration

In the method, we detect many notes which include both actual notes and as well as notes during transition. The duration thresholding is done to eliminate the notes during transition periods and retain the actual notes. If the value of T_d is low, then a large number of notes are detected out of which many transition notes also get included. In the other case, if T_d is very high, less number of notes get detected, but at the same time, a number of actual notes get flashed away. So, we need to set the threshold value so that transition notes get eliminated and also retain the actual notes.

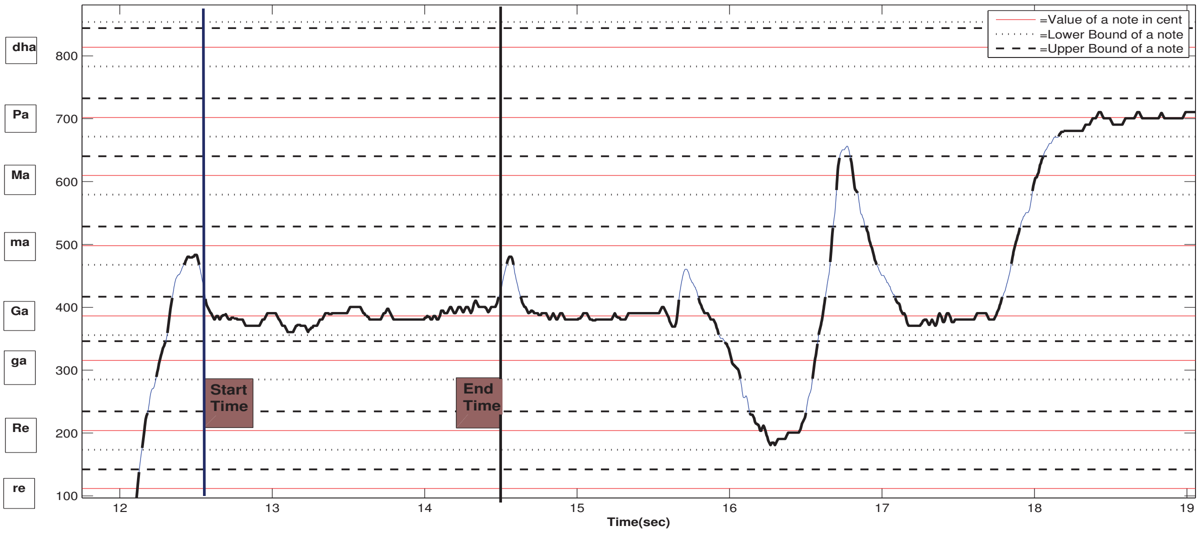

Fig. 2. Melody contour vs. Time plot. The Bold portion of the melody contour denotes a note mentioned along y-axis. *Start time* and *End time* of the note

TABLE III
PERFORMANCE OF THE SYSTEM WITH VARIOUS MELODY TOLERANCE AND THRESHOLD DURATION VALUES

File No.	**Melody Tolerance(δ)**	**Threshold Duration(T_d)**	**MT**	**AT**
1	30	0.02	64.03	64.67
2	20	0.025	65.59	74.80
3	15	0.02	60.27	82.63
4	15	0.045	52.94	83.33
5	40	0.03	65.88	35.83
6	50	0.02	73.95	41.03
7	20	0.025	87.27	82.50
8	15	0.02	61.94	82.16
9	15	0.035	80.76	85.29
10	50	0.02	73.68	36.50
11	50	0.02	77.08	36.21
12	45	0.02	89.47	35.92
13	30	0.045	70.96	66.66
14	15	0.045	30.47	79.22
15	45	0.02	89.47	53.78
16	40	0.02	66.21	50
17	45	0.02	72.60	42.10
18	15	0.06	50	95.45
19	25	0.02	70.12	61.93
20	30	0.02	76.62	68.94

C. Optimized Value

The very high and very low value of both parameters δ and T_d have a great impact on the performance of the note detection system. So, for each music file a set of values of δ and T_d are taken, and combinations of values of these two parameters are used to find out which combination gives the best performance. We choose δ to take values from the set $\{10, 15, 20, 25, 30, 35, 40, 45, 50\}$ and T_d to take values from the set $\{20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80\}$ ms. With these two sets, we can have a total of $13 \times 9 = 117$ combinations of input parameters. We have evaluated the performance with all these combinations using both the metrics proposed above.

The combination of the parameters gives the performance of the system in two metrics. One of metric is described in terms of *Performance in terms of Manual Transcription* (MT) and the other described in terms of *Performance in terms of Automatic Transcription* (AT). We get a total of 117 performance values for each music file. In order to select the best combination of the parameters in terms of performance, we combine the performances obtained using the two metrics for each combination of the parameters. To do this we normalized both MT and AT values by dividing each value by the maximum values of the corresponding metrics. Then, the performance from both of these metrics are linearly combined to obtain a score which denotes a final performance value. In other words this final value gives the overall performance of the system. The following mathematical equation gives the overall performance:

$$P = 0.5 \times MT_i^{normalized} + 0.5 \times AT_i^{normalized} \qquad (4)$$

where $i = 1, 2, 3, ..., 117$. The maximum value of P is selected as best performance, and the corresponding input combination is taken as best input combination for that particular file. This process is repeated for 20 music files. The results are given in table III.

The average values of δ and T_d were calculated and used to set the parameter values. The average values of δ and T_d were found to be 30.5 and 0.0275, and used to set the parameter values for testing.

TABLE IV
PERFORMANCE OF THE SYSTEM FOR TEST AUDIO FILES WITH OPTIMIZED TOLERANCE AND DURATION PARAMETER

File Name	MT	AT	Overall Performance
1	54.43	56.09	55.26
2	65.85	51.72	58.78
3	45.45	38.05	41.75
4	60.97	51.92	56.44
5	53.22	49.33	51.27
6	66.66	43.00	54.83
7	54.05	38.98	46.51

VI. Results and Discussion

Seven music files were taken for testing purposes and the results are given in table IV. From the table, it can be inferred that in most of the cases, the system performs well but in some cases, the performance degrades.

Here we consider manual transcription as the benchmark. However, manual transcription may not be 100% accurate. Also, the state of the art melody extraction algorithm used is also not 100% accurate. Furthermore, the files which give low performance have more ornamentation compared to the other files. These might be the reasons of the system not performing well for some test cases. However, in spite of these minor deficiencies, the overall performance of the system is quite satisfactory.

VII. Conclusion and Future Work

We have built an automatic note transcription system for heterophonic music signal. The system takes input as a music file and generates a transcription file. The transcription file consists of a sequence of notes. The notes are detected by the system using the melody values derived from the melody contour. The accuracy of the transcription system is governed by the two parameters. These parameters are the tolerance melody value and threshold duration of the note. We have empirically optimized these parameters to maximize the accuracy of the system. The performance of the system is evaluated by two metrics. The evaluation of the results indicates that the system performs satisfactorily, though it produces some *false notes*. In conclusion, the development of the proposed automatic transcription system performs better in detecting the majority of the notes. If the music file consists of lots of ornamentation, then the automatic note transcription system fails to detect the notes correctly.

In future, we want to analyze the all different kinds of ornamentation present in the Hindustani music. From this analysis, the overall performance of the automatic note transcription system can be improved. Further, the system can be used for detecting the ragas present in the Hindustani music.

References

[1] J. C. Ross and P. Rao, "Detection of raga-characteristic phrases from hindustani classical music audio," in *Serra X, Rao P, Murthy H, Bozkurt B, editors. Proceedings of the 2nd CompMusic Workshop; 2012 Jul 12-13; Istanbul, Turkey. Barcelona: Universitat Pompeu Fabra; 2012. p. 133-138.* Universitat Pompeu Fabra, 2012.

[2] S. Bagchee, *Nād: Understanding rāga music*. Eeshwar, 1998.

[3] P. Rao, J. C. Ross, K. K. Ganguli, V. Pandit, V. Ishwar, A. Bellur, and H. A. Murthy, "Classification of melodic motifs in raga music with time-series matching," *Journal of New Music Research*, vol. 43, no. 1, pp. 115–131, 2014.

[4] G. K. Koduri, S. Gulati, P. Rao, and X. Serra, "Rāga recognition based on pitch distribution methods," *Journal of New Music Research*, vol. 41, no. 4, pp. 337–350, 2012.

[5] G. Pandey, C. Mishra, and P. Ipe, "Tansen: A system for automatic raga identification." in *IICAI*, 2003, pp. 1350–1363.

[6] S. Shetty and K. Achary, "Raga mining of indian music by extracting arohana-avarohana pattern," *International Journal of Recent Trends in Engineering*, vol. 1, no. 1, pp. 362–366, 2009.

[7] P. Dighe, H. Karnick, and B. Raj, "Swara histogram based structural analysis and identification of indian classical ragas," in *ISMIR*, 2013, pp. 35–40.

[8] R. Sridhar and T. Geetha, "Swara indentification for south indian classical music," in *Information Technology, 2006. ICIT'06. 9th International Conference on*. IEEE, 2006, pp. 143–144.

[9] T. Viswanathan and M. H. Allen, "Music in south india," *Oxford University Press*, vol. 166, p. 169, 2004.

[10] G. E. Poliner, D. P. Ellis, A. F. Ehmann, E. Gómez, S. Streich, and B. Ong, "Melody transcription from music audio: Approaches and evaluation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 4, pp. 1247–1256, 2007.

[11] J. Salamon and E. Gómez, "Melody extraction from polyphonic music signals using pitch contour characteristics," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 6, pp. 1759–1770, 2012.