TrAP: An Interactive System to Generate Valid Raga Phrases from Sound-Tracings

Udit Roy
IIIT, Hyderabad
Gachibowli, Hyderabad
AP, India 500032
udit.roy@research.iiit.ac.in

Tejaswinee Kelkar
IIIT, Hyderabad
Gachibowli, Hyderabad
AP, India 500032
tejaswinee.k@research.iiit.ac.in

Bipin Indurkhya
IIIT, Hyderabad
Gachibowli, Hyderabad
AP, India 500032
bipin@iiit.ac.in

ABSTRACT

We propose a new musical interface, TrAP (TRace-A-Phrase) for generating phrases of Hindustani Classical Music (HCM). In this system the user traces melodic phrases on a tablet interface to create phrases in a raga. We begin by analyzing tracings drawn by 28 participants, and train a classifier to categorize them into one of four melodic categories from the theory of Hindustani Music. Then we create a model based on note transitions from the raga grammar for the notes used in the singable octaves in HCM. Upon being given a new tracing, the system segments the tracing and computes a final phrase that best approximates the tracing.

Keywords

Interface, Sound Tracing, Drawing, Motion and Gesture, Indian Music

1. INTRODUCTION

Computer controlled interfaces and instruments have been explored in the area of sound creation and manipulation. Such interfaces allow a user not only to control musical parameters but also to express through them. New interfaces can dissociate the expressive action from the sound in a variety of ways, making the mapping between action and sound complex and indeterminate [1].

Research in new musical interfaces predominantly concentrates on generating idiosyncratic grammars for novel instruments, although little has been done to create interfaces which accomodate existing grammars that musicians are already accustomed to. This limits users from having virtuosic control over the instrument [2].

The first generative theory of tonal music [3] formalized some concepts of composition and presented hierarchical systems that shape musical intuitions. [4] [5] explain some methods of generating computer music in the Western classical style. These methods also learn and generate signature motifs used by particular composers. However, they cannot be used in Hindustani Classical Music (HCM), because the parallels between the concepts of western tonal theory and HCM are limited to tuning and modal constructs.

The probabilistic generation of HCM was first proposed in [6]. Music generation in [7] explores the probabilistic modeling of raga grammars. Similar strategies have been used in music information retrieval systems for HCM [8] [9] [10] to help identify ragas as well. Although the methods to learn raga grammars are largely based on machine learning techniques [11], generativity in Hindustani music has not been studied explicitly. Strategies for creating valid phrases in HCM have not been investigated either, because extracting information from live recording and mapping it to analytical concepts from music theory were found challenging.

Mapping motion to music is a known pedagogical technique used both in Western and Indian musical systems. Input device technology that considers human motion has also been explored in the form of gestural and motion capture based systems [12]. More recent studies have focused on the relationships between sounds and natural movements through tracings of sound objects [13] [14]. Data from human motion has also been mapped to a scheme of generative music [15] while [12] focuses on free musical textures in improvisation. Sound-tracing studies to distinguish between trained and untrained musicians have also been conducted in [16] to quantify shapes as natural visual representations.

Due to the language-like nature of phrase mappings in HCM, phrase level grammars are a known compositional technique [17] [18]. We explore the application of these grammars through the medium of visual representations and propose a system for computer-aided composition which dynamically generates valid phrases in ragas from a tracing. Our goal is to use sound-tracings as an interface to generate raga phrases. In the next section we describe our initial data collection process to learn a model for melodic mapping. The interface design is then discussed in section 3.

NIME'14, June 30 – July 03, 2014, Goldsmiths, University of London, UK.

2. SOUND TRACING CLASSIFICATION

Formal theory of melody in Indian music appears in many texts [18] [17]. All possible melodic phrases are divided into four categories or *varnas*:

1. *Aarohi* / Ascending (A)- global ascending contour for pitches in the melody
2. *Avarohi* / Descending (D)- global descending contour for pitches in the melody
3. *Sthyayi* / Stationary (S)- hover around a stationary point in the melodic frame
4. *Sanchari* / Random (R)- do not fit any of the above three categories

To begin with, we conduct an experiment to collect sound tracings as a response to the aforementioned phrase categories and explore their classifiability to build a corpus.

2.1 Stimuli Design

We create a set of 64 stimuli in various combinations. The duration of each stimulus is between 4 and 6 seconds. All stimuli were sung by one singer. There were three independent variables as follows,

1. Phrase Type: This includes the four classes for melodic categories as described above.

2. Articulation style: These are divided into two types: *Gamak* and Flat notes. *Gamak* notes are with continuous slides between two notes. Flat notes do not have this kind of representation.

3. Thaat group: We divide the ten *thaats* in HCM into four groups on the basis of their affinity and general nature.

2.2 Experimental Setup

In the first part of the study, we obtain data for each of the four phrase categories from 28 participants (median age = 24.96) mixed across both genders. The stimulus set was divided into eight playlists. We analyze the tracings drawn by 28 participants and compare them according to the phrase types used. We find strong correlation between natural movement accompanying auditory perception of musical phrase and some features of the music itself: pitch positions, dynamic stresses and accents. Participants were asked to trace 32 musical phrases on a WACOM Bamboo digital tablet.

After obtaining a dataset containing 32 $\times$ 28 tracings equally distributed over all classes, we approximate our tracings as described in section 3.1 and group them into melodic phrase categories using a K Nearest Neighbor(KNN) classifier. The classification results as discussed in the section 3.2 as a part of the interface design. In Fig. 2, we plot a characteristic example of the first three phrase categories along with all the user tracings in the second column.

3. INTERFACE DESIGN

TrAP interprets the users' tracings and generates a valid musical phrase based on a selection of ragas. TrAP relies on various rules specific to ragas while generating a phrase. The components of TrAP are shown in the schematic diagram of Fig. 3.

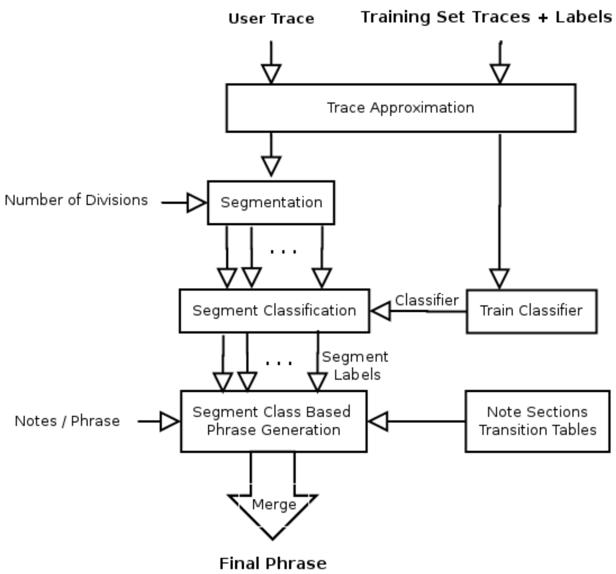

Figure 1: Structure of TrAP

3.1 Tracing Approximation

Given a tracing $T = (x, y)$, where x is the time axis and y is the pitch axis, we normalize its length, and center the y values across its mean. We now interpolate the normalized tracing using a cubic spline function with 8 knots. This approximation is required to remove aberrations and discontinuity from the tracing, thereby capturing the global contour of the tracing.

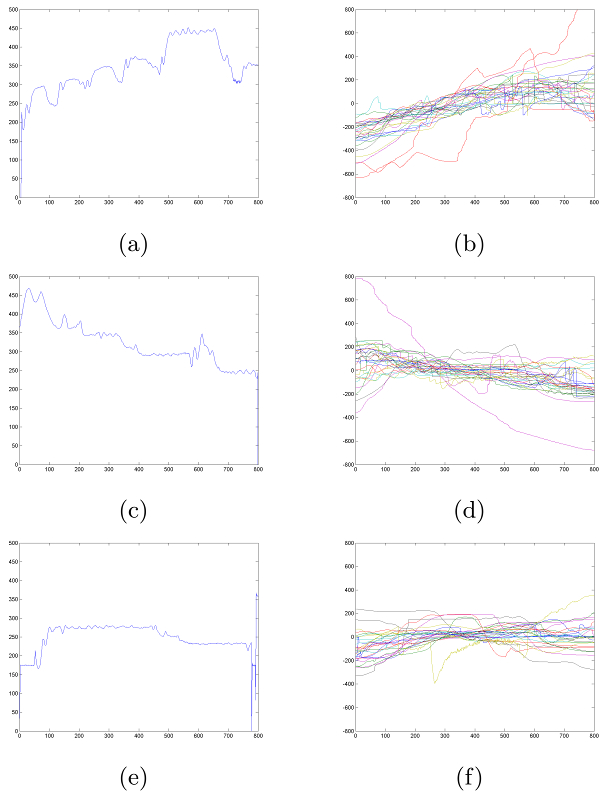

Figure 2: Pitch data and corresponding user tracings for Ascending(a,b), Descending(c,d) and Stationary(c,d) phrases

3.2 Phrase Classification

Using the approximated tracings as raw features for the system, we trained a K-nearest neighbour classifier (KNN) with cityblock distance metric. With the top K results, we selected the mode to be the output class label.

We performed a 10-fold randomized cross validation to check our system's performance. Class accuracies of 81%, 63%, 41%, 28% for ascending, descending, stationary and random phrase types were obtained with an overall accuracy of 54%. We observed better performance for the first three classes as they tend to have a fixed global contour, whereas the random class *(sanchari)* does not.

3.3 Tracing Segmentation

Once a tracing is drawn, it is split into equally-spaced divisions based on a user parameter. After spliting, each sub tracing is classified by the KNN search described above and assigned a class label. The user parameter is to specify the resolution to which information can be utilized from the tracing. The granularity can be exploited to generate a phrase closer to the tracing by selecting a high number of divisions. The subtracings obtained are interpolated to the desired feature vector length and classified using the KNN classifier described above.

3.4 Phrase Generation

Given the subtracings and their respective class labels, we generate subphrases for each class label iteratively. We con-

Data: ;
$L \rightarrow$ Subtracing label ;
$N \rightarrow$ Notes per subtracing ;
$b \rightarrow$ Previous phrase last note ;
$S_n \rightarrow n-$Note sets ;
$T_n \rightarrow n-$Transition sets ;
$F \rightarrow$ Forbidden consecutive notes ;
$R \rightarrow$ Acceptable thresholds per class ;
Result: $P \rightarrow$ Final Phrase
initialization;
$s \rightarrow$ pick initial set from S_n ;
$n \rightarrow 0$;
$P \rightarrow b$ append P;
while $n < N$ **do**
    $r \rightarrow$ pick random note from set s ;
    if *last note* P *and* r *in* F **then**
        continue ;
    end
    $P' \rightarrow P$ append r ;
    if *mean(derivative(*P'*)) in range* R_L **then**
        $n = n + 1$;
    end
    $s \rightarrow$ pick random set number from $T_n(r)$;
end

Algorithm 1: Phrase generation algorithm

catenate them all to get the final phrase in the end. All subphrases generated are built using the grammar specified by the raga explained in section 3.5. Algorithm 1 presents the details of the system component. The raga grammar rules are stored as different sets of notes, transition tables for moving from one set to another and a list of forbidden transitions. All random assignments are based on distribution built from domain knowledge in raga formation in relative scaling. We initially find out the relative occurence of a note after a subjective evaluation which is then quantifed and given a muchness score to produce realistic phrases.

Figure 3: A sample phrase generated by TrAP in Raga Yaman

The classification of an intermediate phrase during its construction is based on the mean of the first-order derivative (referred as μ_d) of MIDI sequence values. The μ_d has bounds for each class of phrase, which serve as a constraint during the iterative process of generation. Since the chosen MIDI sequence values are discrete and a single increment represents a half step, it allows us to objectively define the class thresholds over μ_d and constrain them as needed. Fig. 4 shows an example of three sample phrases generated by the system in raga Yaman.

3.5 Raga Grammar

In this section we explain a framework for adhering to the raga structure to create valid phrases. First we determine the intonation and note positions of the constituents notes of a raga. We describe some traditional categories of defining a raga. At any point of time in exposition, the constituent notes, the heavily used notes for each raga, and some characteristic phrases, largely determine the melodic movement. We can use this strategy for generating random sequences of valid phrases in HCM if we model the following parameters [19] [20]:

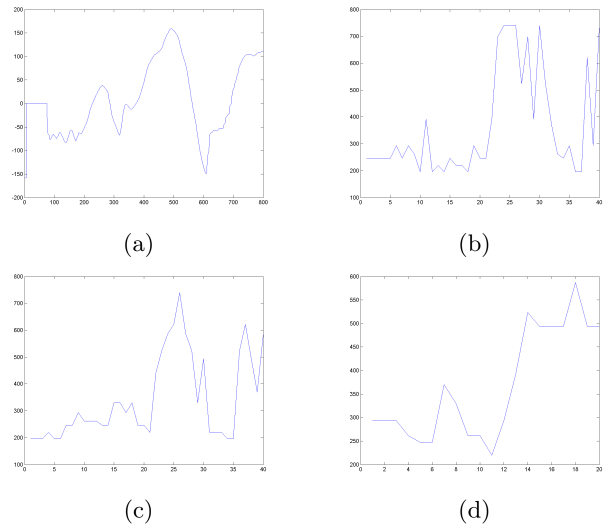

Figure 4: Sample tracing (a) and three TrAP generated melodic phrases in raga Yaman shown in (b), (c), (d)

- *Vadi* - This is the most important note of a raga. It is the note towards which the melody mostly gravitates.
- *Samvadi* - This is the second most important note of the raga.
- *Anuvadi* - This is a note that is used optionally in rendition, to beautify the raga.
- *Bahutva* - This literally means muchness [17] or copious use. There are two types of *Bahutva*: *Abhyas Mulak* and *Alanghan Mulak*, which are two degrees of muchness.
- *Aplatva* - This literally means scant or non-copious use. There are two types of *Alpatva*: *Langhan Mulak* (to be used in transition) and *Abhyas Mulak* (almost not sung at all).
- Nyas - This is the cadential or the final tone to be used at the completion of phrases.
- *Poorvanga / Uttaranga Vadi* - This is determined by whether the vadi or the tenor note of a raga is in the *Poorvanga*(first half), or the *Uttaranga* (second half) of the singable octave. This will help determine the hover point of the phrases, and there would be a higher chance to select phrases from this part of the octave, or for the phrases to move in this part of the octave.
- Characteristic Phrases - These are phrases that occur very often in the rendition of a raga.

We do this by emplyoing the following strategies:

1. Picking an accurate pitch class set for the chosen raga from the singable octave from the corresponding midi notes for a raga.
2. The notes chosen must be around each other to ensure that phrases have continuity. We enforce this by dividing the octave into four parts from which pitches are chosen to restrain the amount of jumps in the melody. A random set is picked to start with each time. There are overlapping notes between adjacent sets, that are pivot points from which the melody can transition. In

Table 1: Pitch Sets in different Ragas

Raga Name	Notes Contained (A)	Notes Contained (D)
Yaman	Ti Re Mi Fi Sol La Ti Do	Do Ti La Sol Fi Mi Re Do
Bhupali	Do Re Mi Sol La Do	Do La Sol Mi Re Do
Bageshri	Te Do Me Fa La Te Do	Do Te La Sol Fa Me Re Do

Table 2, we see the four sets that divide the two octaves that are sung and used, and the respective transition sets, along with their assigned muchness scores based on the above characteristics.

3. Assigning each note a measure of muchness based on the above 8 criteria between level 1 and 5. A non-uniform probability distribution for picking notes from each set which follows the Alpatva / Bahutva paradigm in HCM as explained above, where some notes are sung less frequently than others.

4. Forbidding the note transitions that are disallowed in the raga structure: The example in Fig. 3 is that of a phrase generated in raga yaman. In this raga, it is not possible to go from a Ti to a Re, and we enforce rules like this for note transitions, allowing the phrases to remain valid.

Table 2: Notes in the singable octave, divided into sets, with transition set numbers and muchness score

Note	MIDI Value	Set Numbers	P(Transition)	Muchness
Pa	55	1	1, 2	1
Dha	57	1	1	1
Ni	59	1, 2	2, 1	3
Sa	60	1, 2	1, 2, 3	3
Re	62	1, 2	2, 3, 4	2
Ga	64	2	2, 3, 1, 4	5
Ma	66	2, 3	2, 3	2
Pa	67	2, 3	2, 3, 4	3
Dha	69	3	3, 2	1
Ni	71	3, 4	3, 4, 2, 1	4
Sa	72	3, 4	3, 4, 2	3
Re	74	4	4, 3	2
Ga	75	4	4, 3	3
Ma	77	4	4	2
Pa	78	4	4	1

4. CONCLUSIONS

We presented here an interface, TrAP to generate valid melodic phrases in different ragas. The interface has two user-defined parameters to control granularity and length of the final phrase. The system allows the user to also determine the contour of the melodic phrases. Transition tables and pitch class sets can be populated with several other ragas enabling the user to express the tracings in different ragas. Future work can include rhythmic variation and enabling higher order compositional structures through this schema.

5. REFERENCES

[1] Sergi Jorda. Digital lutherie: Crafting musical computers for new musics performance and improvisation. *PhD diss., Universitat Pompeu Fabra, Departament de Tecnologia*, 2005.

[2] Christopher Dobrian and Daniel Koppelman. The 'E' in NIME: Musical Expression with New Computer Interfaces. In *Proceedings of the 2006 conference on New interfaces for musical expression*, pages 277–282. IRCAMâĂŤCentre Pompidou, 2006.

[3] Fred Lerdahl and Ray S Jackendoff. *A generative theory of tonal music.* The MIT Press, 1983.

[4] David Cope. *Virtual music: computer synthesis of musical style.* The MIT Press, 2004.

[5] David Cope. *Computer models of musical creativity.* MIT Press Cambridge, 2005.

[6] Bigamudre Chaitanya Deva. *An introduction to Indian music.* Publications Division, Ministry of Information and Broadcasting, Government of India, 1981.

[7] George Tzanetakis and Perry Cook. *Manipulation, analysis and retrieval systems for audio signals.* Princeton University Princeton, NJ, USA, 2002.

[8] A Srinath Krishna, PV Rajkumar, KP Saishankar, and Mala John. Identification of carnatic raagas using hidden markov models. In *Applied Machine Intelligence and Informatics (SAMI), 2011 IEEE 9th International Symposium on*, pages 107–110. IEEE, 2011.

[9] Zbigniew W Ras and Alicja Wieczorkowska. *Advances in music information retrieval*, volume 274. Springer, 2010.

[10] Ashwin Bellur, Vignesh Ishwar, Xavier Serra, and Hema A Murthy. A knowledge based signal processing approach to tonic identification in indian classical music. 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. 113-118.* Universitat Pompeu Fabra, 2012.

[11] Ajay Srinivasamurthy and Parag Chordia. Multiple viewpoint modeling of north indian classical vocal compositions.

[12] Tomoko Hashida, Yasuaki Kakehi, and Takeshi Naemura. Ensemble system with i-trace. In *Proceedings of the 2004 conference on New interfaces for musical expression*, pages 215–216. National University of Singapore, 2004.

[13] Alexander Refsum Jensenius. Musical gestures. *Musical Gestures: Sound, Movement, and Meaning*, 12, 2009.

[14] Rolf Inge Godøy and Marc Leman. *Musical gestures: Sound, movement, and meaning.* Routledge, 2009.

[15] Jan C Schacher and Angela Stoecklin. Traces–body, motion and sound. In *Proc. of the International Conference on New Interfaces for Musical Expression, Oslo, Norway*, 2011.

[16] Genevieve Noyce. Quantifying shapes: Mathematical techniques for analysing visual representations of sound and music. *Empirical Musicology Review*, 8(2):128–154, 2013.

[17] L. Rowell. Early indian musical speculation and the theory of melody. *Journal of Music Theory*, 25(2).

[18] Manorama Śarmā. *Tradition of Hindustani Music.* APH Publishing, 2006.

[19] Vasant Vasant. Sangeet visharad, 1998.

[20] Vishnu Narayan Bhatkhande. *Kramik pustak malika.* Sangeet Karyalaya, 1993.