Global Journal of Computer Science and Technology
Volume 11 Issue 22 Version 1.0 December 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Automatic Raaga Identification System for Carnatic Music Using Hidden Markov Model

By Prasad Reddy P.V.G.D, B. Tarakeswara Rao, Dr. K.R Sudha, Hari CH.V.M.K.

Andhra University, Visakhapatnam, India

Abstract - As for as the Human Computer Interactions (HCI) is concerned, there is broad range of applications in the area of research in respective of Automatic Melakarta Raaga Identification in music. The pattern of identification is the main object for which, the basic mathematical tool is utilized. On verification, it is observed that no model is proved consistently and effectively to be predicted in its classification. This paper is, therefore, introduces a procedure for Raaga Identification with the help of Hidden Markov Models (HMM) which is rather an appropriate approach in identifying Melakarta Raagas. This proposed approach is based on the standard speech recognition technology by using Hidden continuous Markov Model. Data is collected from the existing data base for training and testing of the method with due design process relating to Melakarta Raagas. Similarly, to solve the problem of automatic identification of raagas, a suitable approach from the existing database is presented. The system, particularly, this model is based on a Hidden Markov Model enhanced with Pakad string matching algorithm. The entire system is built on top of an automatic note transcriptor. At the end, detailed elucidations of the experiments are given. It clearly indicates the effectiveness and applicability of this method with its intrinsic value and significance.

Keywords : Melakarta Raaga, Raaga Recognition, Hidden Markov Models (HMM) classifier, Pakad.

GJCST Classification : G.3, I.2.6

Automatic Raaga Identification System for Carnatic Music Using Hidden Markov Model

Prasad Reddy P.V.G.D [α], B. Tarakeswara Rao [Ω], Dr. K.R Sudha [β], Hari CH.V.M.K. [ψ]

Abstract - As for as the Human Computer Interactions (HCI) is concerned, there is broad range of applications in the area of research in respective of Automatic Melakarta Raaga Identification in music. The pattern of identification is the main object for which, the basic mathematical tool is utilized. On verification, it is observed that no model is proved consistently and effectively to be predicted in its classification.

This paper is, therefore, introduces a procedure for Raaga Identification with the help of Hidden Markov Models (HMM) which is rather an appropriate approach in identifying Melakarta Raagas. This proposed approach is based on the standard speech recognition technology by using Hidden continuous Markov Model. Data is collected from the existing data base for training and testing of the method with due design process relating to Melakarta Raagas. Similarly, to solve the problem of automatic identification of raagas, a suitable approach from the existing database is presented.

The system, particularly, this model is based on a Hidden Markov Model enhanced with Pakad string matching algorithm. The entire system is built on top of an automatic note transcriptor. At the end, detailed elucidations of the experiments are given. It clearly indicates the effectiveness and applicability of this method with its intrinsic value and significance.

Keywords : Melakarta Raaga, Raaga Recognition, Hidden Markov Models (HMM) classifier, Pakad.

I. Introduction and Problem Definition

Raaga classification has been a central pre-occupation of Indian music theorists for centuries [1], reflecting the importance of the concept in the musical system. The ability to recognize raagas is an essential skill for musicians and listeners. The raaga provides a framework for improvisation, and thus, generates specific melodic expectations for listeners, crucially impacting their experience of the music. Recently, MIR researchers have attempted to create systems that can accurately classify short raaga excerpts [1, 2]

Raaga recognition is a difficult problem for humans, and it takes years for listeners to acquire these skills for a large corpus. It can be very difficult to precisely explain precisely the essential qualities of a raaga. Most common descriptions of raagas are not sufficient to distinguish them. For example, many raagas share the same notes, and even similar characteristic phrases. While ascending and descending scales are sometimes used absolutely in an artistic manner. They are highly abstracted from the real melodic sequences found in performances. It takes a performer lengthy practice to fully internalize the raaga and be able reproduces it. This is despite the fact that humans are highly adept at pattern recognition, they have little problem with pitch recognition, even in harmonically and timbrally dense settings. Clearly, there are several other difficulties for an automatic Melakarta raaga recognition system.

The special feature of the paper is the study of musical raagas through major contributions. In first place, our solutions based primarily on techniques from speech processing and pattern matching, which shows that techniques from other domains can be purposefully extended to solve problems in computational musical raagas. Secondly, the two note transcription methods presented are novel ways to extract notes from sample raagas of Indian classical music. This approach has given very encouraging results. Hence these methods could be extended to solve similar problems in musical raagas and other domains.

The remaining parts of the paper is organized as follows. Section 2 highlights some of the useful and related previous research work in the area. The solution strategy is discussed in detail in Section 3. The test procedures and experimental results are presented in Section 4... Finally, Section 5 lists out the conclusions.

II. Literature Review

On verification of different concepts it is noticed that a that very little work has taken place in the area of applying techniques from computational musicology and artificial intelligence to the realm of Indian classical music. Of special interest to us, is the work done by Gaurav Pandey et al. [17] present an approach to solve the problem of automatic identification of Raagas from audio samples is based on a Hidden Markov Model enhanced with a string matching algorithm, Sahasrabuddhe et al. [7,8,6,14]. In their work, Raagas have been modelled as finite automata which were constructed by using information codified in standard texts on classical music. This approach was used to generate new samples of the Raaga, which were technically correct and were indistinguishable from compositions made by humans.

Author [α] : Department of CS&SE, Andhra University, Visakhapatnam, India.
Author [Ω] : Scohool of Computing, Vignan University, Guntur, India.
E-mail : tarak7199@gmail.com
Author [β] : Department of EEE, Andhra University, Visakhapatnam, India.
Author [ψ]: Department of IT, Gitam University, Visakhapatnam, India.

It is also further noticed that Hidden Markov Models [6,18,19] are now widely used to model signals whose functions however are not known. A Raaga too, can be considered to be a class of signals and can be modeled as an HMM. The advantage of this approach is the similarity and it has with the finite automata formalism suggested above [11].

It is obvious that "Pakad" is a catch-phrase of the Raaga, with each Raaga having a different Pakad[19]. Most people claim that they identify the Raaga being played by identifying the Pakad [19] of the Raaga[17]. However, it is not necessary for a Pakad be sung without any breaks in a Raaga performance. Since the Pakad is a very liberal part of the performance in itself, standard string matching algorithms were not guaranteed to work. Approximate string matching algorithms designed specifically for computer musicology, such as the one by Iliopoulos and Kurokawa [10] for musical melodic recognition with scope for gaps between independent pieces of music, seemed more relevant.

The other connected works which deserve mention here are the ones on Query by Humming [8] and [9], and Music Genre Classification [7]. Although these approaches are not followed, there is a lot of scope for using such low level primitives for Raaga identification, and this might open avenues for future research.

III. Present Work

To solve the problems in speech processing Hidden Markov Models have been traditionally used. One important class of such problems is that involving word recognition. The problem is very closely related to the word recognition problem. This correspondence can be established by the simple observation that raaga compositions.

This correspondence between the word recognition and raaga identification problems is exploited to devise a solution to the latter. This solution is explained below. Also presented is an enhancement to this solution using the Pakad of a raaga, However, both these solutions assume that a note transcriptor is readily available to convert the input audio sample into the sequence of notes used in it. It is generally cited in literature that "Monophonic note transcription is a trivial problem". However, our observations in the field of Indian classical music were counter to this, particularly because of the permitted variability in the duration of use of a particular note. To handle this, two independent heuristic strategies are designed for note transcription from any given audio sample. These strategies are explained in the corresponding pages.

a) Hidden Markov Models

The doubly embedded stochastic process is Hidden Markov Models[17,19] where the underlying stochastic process is not directly observable, and have the capability of effectively modelling statistical variations in spectral features i.e. processes which generate random sequences of outcomes according to certain probabilities. More concretely, an HMM[19] is a finite set of states, each of which is associated with a (generally multidimensional) probability distribution. Transitions among the states are governed by a set of probabilities called transition probabilities. In a particular state, an outcome or observation can be generated, according to the associated probability distribution. It is only the outcome not the state that is visible to an external observer. So states are "hidden" and hence the name hidden Markov model.

In order to define an HMM [16,17,18,19] completely, the following elements are needed [12].

- The number of states of the model, N
- The number of observation symbols in the alphabet, M.
- A set of state transition probabilities

$$A = \{a_{ij}\} \tag{1}$$

$$a_{ij} = p\{q_{t+1} = j | q_t = i\},\ 1 \le i, j \le N \tag{2}$$

where qt denotes the current state.

- A probability distribution in each of the states,

$$B = \{b_j(k)\} \tag{3}$$

$$b_j(k) \ge p\{\alpha_t = v_k | q_t = j\},\ 1 \le j \le N,\ 1 \le k \le M \tag{4}$$

where vk denotes the kth observation symbol in the alphabet and αt the current parameter vector. The initial state distribution, $\pi = \{\pi i\}$ where,

$$\pi_i = p\{q_1 = i\},\ 1 \le i \le N \tag{5}$$

Thus, an HMM can be compactly represented as

$$\lambda = (A, B, \pi) \tag{6}$$

The HMM[19] and their derivatives have been widely applied to speech recognition and other pattern recognition problems [1]. Most of these applications have been inspired by the strength of HMMs, ie the possibility to derive understandable rules, with highly accurate predictive power for detecting instances of the system studied, from the generated models. This also makes HMMs the ideal method for solving Raaga identification problems, the details of which are presented in the next subsection.

b) HMM in Raaga Identification

The raaga identification problem falls largely in the set of speech processing problems. This justifies the use of Hidden Markov Models in our solution[19]. Two other important reasons which motivated the use of HMM in the present context are:

1. The sequences of notes for different Raagas are very well defined and a model based on discrete states with transitions between them is the ideal representation for these sequences [4].
2. The notes are small in number, hence making the setup of an HMM easier than other methods.
3. This HMM, which is used to capture the semantics of a Raaga, is the main component of our solution. Construction of the HMM Used The HMM used in our solution is significantly different from that used in, say, word recognition. This HMM, which is called λ from now on that can be specified by considering each of its elements separately.
4. Each note in each octave represents one state in λ. Thus, the number of states, N=12x3=36. Here, the three octaves of Indian classical music are considered, namely the Mandra, Madhya and Tar Saptak, each of which consists of 12 notes.
5. The transition probability A = {aij } represents the probability of note j appearing after note i in a note sequence of the Raaga represented by λ. The initial state probability π = {πi} represents the probability of note i being the first note in a note sequence of the raaga represented by λ.

The outcome probability B = {Bi(j)} is set according to the following formula

$$B_i(j) = \begin{matrix} 0, & i7=j \\ 1, & i=j \end{matrix} \qquad (7)$$

It is Observed, at each state α in λ, the only possible outcome is note α the last condition takes the hidden character away from λ, but it can be argued that this setup success for the representation of Raagas, as our solution distinguishes between performances of distinct Raagas on the basis of the exact order of notes sung in them and not on the basis of the embellishments used. HMM is also a statistical approach, and hence requires large amount of training data for effective estimate of the model parameters.

The system performance degrades when training and testing environments differ.

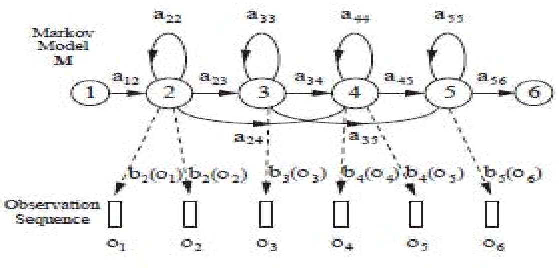

Fig.1 : The Markov Generation Model

By applying the HMMs for Identification One such HMM λI, whose construction is described above, is set up for each Raaga I in the consideration set. Each of these HMMs is trained, i.e. its parameters A and π (B has been pre-defined) is calculated using the note sequences available for the corresponding Raaga with the help of the Baum-Welch learning algorithm.

If all the HMMs have been trained, identifying the closest Raaga in the consideration set on which the input audio sample is based, is a trivial task. The sequence of notes representing the input is passed through each of the HMMs constructed, and the index of the required Raaga can be calculated as under:

$$\text{Index} = \operatorname{argmax}(\log p(O|\lambda_I)),\ 1 \le I \le N \qquad (8)$$

To complete the task, the required raaga can be determined as RaagaIndex. This preliminary solution gave reasonable results in our experiments (Refer to Section 4). However, there was still, a need to improve the performance by incorporating knowledge into the system. This can be done through the Pakad approach, which is discussed in the next section.

c) Pakad Matching

There is general expectation notion in the AI community that the incorporation of knowledge related to the problem being addressed into a system enhances its ability to solve the problem. One such very powerful piece of information about a Raaga is its Pakad[17,19].

Pakad is defined as a condensed version of the characteristic arrangement of notes, peculiar to each Raaga, when repeated in a recital from; it enables a listener to identify the Raaga being played. In other words, Pakad is a string of notes characteristic to a Raaga in which a musician frequently returns while improvising in a performance. The Pakad also serves as a springboard for improvisational ideas; each note in the Pakad can be embellished and improvised around to form new melodic lines. One common example of these embellishments is the splitting of Pakad into several substrings and playing each of them in order in disjoint portions of the composition, with the repetition of these substrings permitted. In spite of such permitted variations, Pakad is a major identifying characteristic of a Raaga and is used even by experts of Indian classical music for identifying the Raaga been played.

The characteristic elements of the Pakad give way to a string matching approach to solve the Raaga identification problem. Pakad matching can be used as reinforcement for an initial estimation of the underlying Raaga in a composition. Two ways of matching the Pakad with the input string of notes are devised in order to strengthen the estimation done. The incorporation of this step makes the final identification process a multi-stage one.

As indicated in the proceeding paragraphs δ-Occurrence with α-Bounded Gaps As mentioned earlier, the Pakad has to appear within the performance of a Raaga. However, it rarely appears in one segment as a whole. It is more common for it to be spread out, with substrings repeated and even other notes inserted in between. This renders simple substring matching algorithms mostly insufficient for this problem. A more appropriate method for matching the Pakad is the δ-occurrence with α-bounded gaps algorithm. The algorithm employs dynamic programming and matches individual notes from the piece to be searched, say t, identifying a note in the complete sample, say p, as belonging to t only if:

1. There is a maximum difference of δ between the current note of p and the next note of t.
2. The position of occurrence of the next note of t in p is displaced from its ideal position by at most α However, this algorithm assumes that a piece(t) can be declared present in a sample(p) only if all notes of the piece(t) are present in the sample(p) within the specified bounds. This may not be true in our case because of the inaccuracy of note transcription (Refer to 3.4). Hence, for each Raaga I in the consideration set, a score γI is maintained as

$$\gamma_I = \frac{mI}{nI}, 1 \le I \le N_{Raagas} \qquad (9)$$

m_I = maximum number of notes of the Pakad of Raaga I identified n_I = number of notes in the Pakad of Raaga I. This score is used in the final determination of the Raaga.

n-gram Matching Another method of capturing the appearance of the Pakad within a Raaga performance is to count the frequencies of appearance of successive n-grams of the Pakad. Successive n-grams of a string are its substrings starting from the beginning and going on till the end of the string is met. Also, to allow for minor gaps between successive notes, each n-gram is searched in a window of size 2n in the parent string. Based on this method, another score is maintained according to the formula,

$$score_I = \Sigma_n \Sigma_j \, freq_{j,n,I} \qquad (10)$$

$freq_{j,n,I}$ = number of times the j^th n-gram of the Pakad of Raaga I is found in the input. This score is also used in the final determination of the underlying Raaga.

Final Determination of the Underlying Raaga Once the above scores have been calculated, the final identification process is a three-step one.

1. The probability of likelihood probl is calculated for the input after passing it through each HMM λI and the values so obtained are sorted in increasing order. After reordering the indices as per the sorting, if

$$\frac{Prob_{N_{Raaga}} - Prob_{N_{Raaga-1}}}{Prob_{N_{Raaga-1}}} > n \quad ----- (11)$$

then,

$$Index = N_{Raagas}$$

2. Otherwise, the values γI are sorted in increasing order and indices set accordingly. After this arrangement, if

$$Prob_{N_{Raaga}} > Prob_{N_{Raaga-1}}, \quad \text{and} \quad \frac{Y_{N_{Raaga}} - Y_{N_{Raaga-1}}}{Prob_{N_{Raaga-1}}} > n$$

then, Index = N_{Raagas}

Otherwise, the final determination is made on the basis of the formula, where K is a predefined constant

$$Index = \text{argmax}\,(\log p\,(o|\lambda_I) + K * score_I),\ 1 \le I \le N \qquad (12)$$

This procedure is used for the identification of the underlying Raaga. The steps enable the system to take into account all probable features for Raaga identification, and thus display good performance. The performance of the final version is discussed in Section 4.

d) Note Transcription

It is based on the assumption that the input audio sample has already been converted into a string of notes[17,19]. However, a few hurdles are faced in our efforts. The main hurdle in this conversion with regard to Indian classical music is the fact that notes are permitted to be spread over time for variable durations in any composition. Here, two heuristics are presented based on the pitch of the audio sample, which are used to derive notes from the input. They are very general and can be used for any similar purpose. From the pitch behavior of various audio clips, two important characteristics of the pitch structure are observed, based on the following two heuristics.

1. input sample.
2. The Hill Peak Heuristic[17,19].

This heuristic identifies notes in an input sample on the basis of hills and peaks occurring in the pitch graph. A sample pitch graph is shown in the figure 2.

A discreet study of an audio clip and its pitch graph shows that the notes occur at points in the graph where there is a complete reversal in the sign of the slope, and in many cases, also when there isn't a reversal in sign, but a significant change in the value of the slope. Translating this into mathematical terms, given a sample with time points $t_1, t_2, ..., t_{i-1}, t_i, t_i+1, ..., t_n$

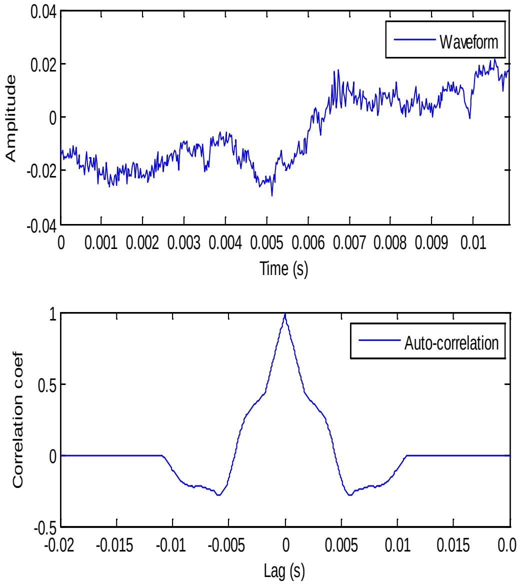

Fig2 : Simple Pitch Graph

Once the point of occurrence of a note has been determined, the note can easily be identified by finding the note with the closest characteristic pitch value. Performing this calculation over the entire duration of the sample gives the string of notes corresponding to it. An important point to note here is that, unless the change in slope between the two consecutive pairs of time points is significant, it is assumed that the last detected note is still in progress, thus allowing for variable durations of notes.

The Note Duration Heuristic is based on the assumption that in a composition a note continues for at least a certain constant span of time, which depends on the kind of music considered. Corresponding notes are calculated for all pitch values available. A history list of the last k notes identified is maintained including the current one (k is a pre-defined constant). The current note is accepted as a note of the sample, only if it is different from the dominant note in the history, i.e. the note which occurs more than m times in the history (m is also a constant). Sample values of k and m are 10 and 8. By making this test, a note can be allowed to extend beyond the time span represented by the history. A pass over the entire set of pitch values, gives the set of notes corresponding to the

IV. Results and Discussion

In conclusion, there are four main modules in this paper. They are the extracted indicators of raaga features, training the features using HMM classifier, training HMM through multiple raaga data and testing the selected raagas by using the features of raaga. The results of classification obtained through both the features are combined to produce more accurate results. The regions commonly identified in both the classification results are now highlighted. In the proposed model, the HMM classifier, Pakad matching, that was implemented to distinguish the various types of raagas is discussed. The implementation is done in MATLAB Speech Tollbox [20].

72 raagas are trained and also tested the said 72 raagas, all of them are recognized. Then, the raagas are tested which are not in the train. After testing the entire process, the obtained results are indicated in the given sample tables with the supporting evidences.

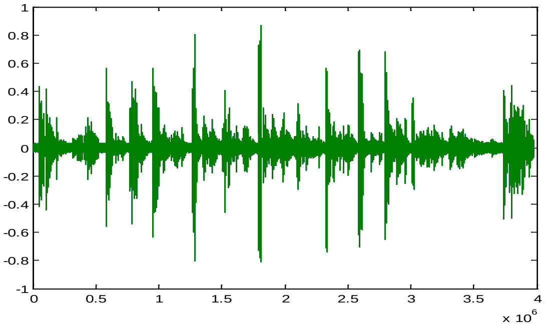

Fig 3 : Plot file for begada raaga (90 sec. Input Raaga)

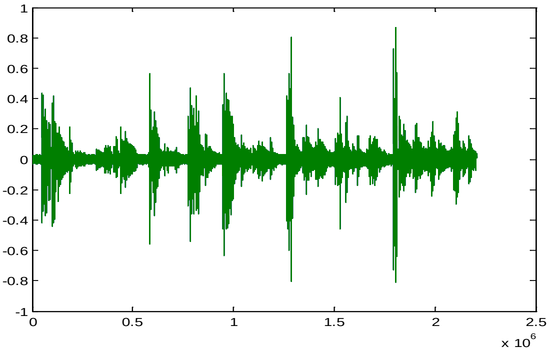

Fig 4 : Plot file for begada raaga (50 sec for Trained raaga)

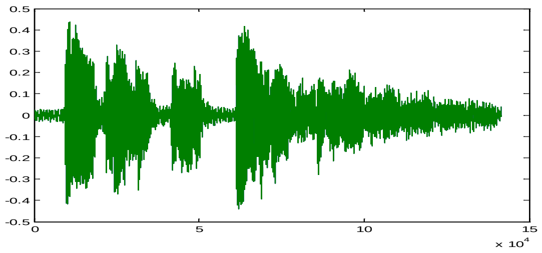

Fig 5 : Plot files for begada raaga (3sec for tested raaga)

Testing Time: First 3 seconds

Table 1 : Confusion matrix indicating that the first 3 sec of each raagas are tested.

Stimulus	Recognized Raagas (%)						
	Begada	Vanasapathi	sundavinodini	Desh	Kapinarayani	Mayamalavagowla	Kandanakuthuhalam
Begada	**88**	54	58	62	65	65	75
Vanasapathi	54	**88**	63	70	72	75	76
sundavinodini	58	68	**88**	68	70	68	68
Desh	62	70	**88**	68	75	78	72
Kapinarayani	65	72	70	85	**88**	80	82
Mayamalavagowla	65	75	68	78	80	**88**	80
Kandanakuthuhalam	**88**	76	68	72	82	80	85

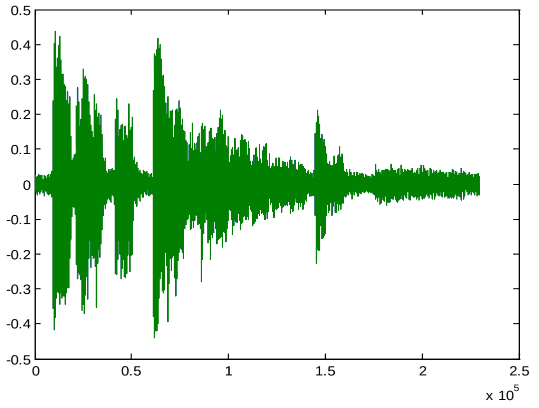

Fig 6 : Plot file for begada raaga (5sec for testing)

Testing Time: First 5 seconds

Table 2 : Confusion matrix indicating that the first 5 sec of each raagas are tested.

Stimulus	Recognized Raagas (%)						
	Begada	Vanasapathi	sundavinodini	Desh	Kapinarayani	Mayamalavagowla	Kandanakuthuhalam
Begada	**90**	58	58	62	65	65	75
Vanasapathi	58	**90**	63	70	72	75	76
sundavinodini	58	68	**90**	68	70	68	68
Desh	62	70	**90**	68	75	78	72
Kapinarayani	65	72	70	85	**90**	80	82
Mayamalavagowla	65	75	68	78	80	**90**	80
Kandanakuthuhalam	**90**	76	68	72	82	80	88

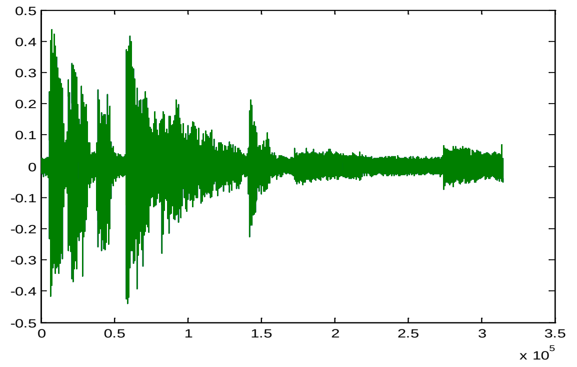

Fig 7 : Plot file for begada raaga (7sec for testing)

Testing Time : First 7 seconds

Table 3 : Confusion matrix indicating that the first 7 sec of each raagas are tested.

Stimulus	Recognized Raagas (%)						
	Begada	Vanasapathi	sundavinodini	Desh	Kapinarayani	Mayamalavagowla	Kandanakuthuhalam
Begada	**92**	58	58	62	65	65	75
Vanasapathi	58	**92**	63	70	72	75	76
sundavinodini	58	68	**92**	68	70	68	68
Desh	62	70	88	**92**	75	78	72
Kapinarayani	65	72	70	85	**92**	80	82
Mayamalavagowla	65	75	68	78	80	**92**	80
Kandanakuthuhalam	**92**	76	68	72	82	80	88

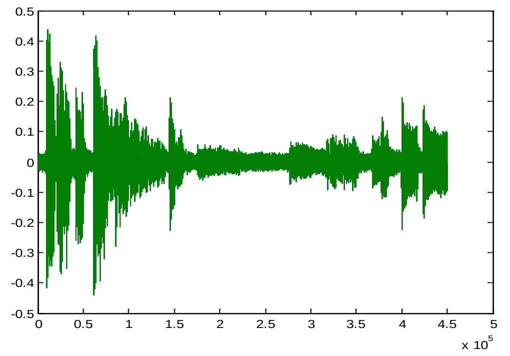

Fig 8 : Plot file for begada raaga (10sec for testing)

Testing Time : First 10 seconds

Table 4 : Confusion matrix indicating that the first 10 sec of each raagas are tested.

Stimulus	Recognized Raagas (%)						
	Begada	Vanasapathi	sundavinodini	Desh	Kapinarayani	Mayamalavagowla	Kandanakuthuhalam
Begada	**92**	58	58	62	65	65	75
Vanasapathi	58	**92**	63	70	72	75	76
sundavinodini	58	68	**92**	68	70	68	68
Desh	62	70	88	**92**	75	78	72
Kapinarayani	65	72	70	85	**92**	80	82
Maya malavagowla	65	75	68	78	80	**92**	80
Kandanakuthuhalam	**92**	76	68	72	82	80	88

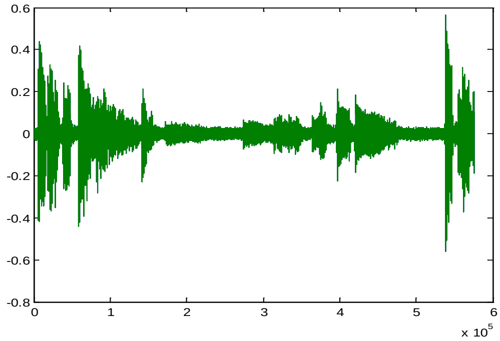

Fig 9 : Plot file for begada raaga (13sec for testing)

Testing Time: First 13 seconds:

Table 5 : Confusion matrix indicating that the first 13 sec of each raagas are tested.

Stimulus	Recognized Raagas (%)						
	Begada	Vanasapathi	sundavinodini	Desh	Kapinarayani	Mayamalavagowla	Kandanakuthuhalam
Begada	**92**	58	58	62	65	65	75
Vanasapathi	58	**92**	63	70	72	75	76
sundavinodini	58	68	**92**	68	70	68	68
Desh	62	70	76	**92**	75	78	72
Kapinarayani	65	72	70	85	**92**	80	82
Mayamalavagowla	65	75	68	78	80	**92**	80
Kandanakuthuhalam	90	76	68	72	82	80	**92**

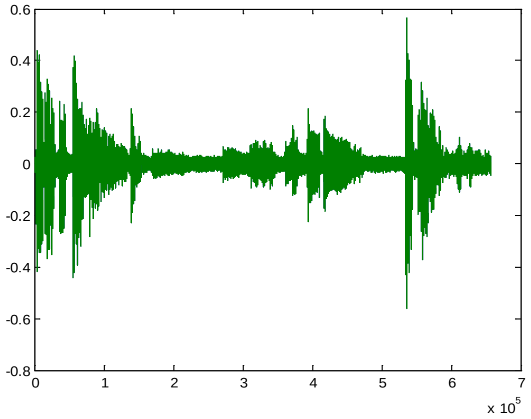

Fig 10 : Plot file for begada raaga (15sec for testing)

Testing Time: First 15 seconds

Table 6 : Confusion matrix indicating that the first 15 sec of each raagas are tested.

Stimulus	Recognized Raagas (%)						
	Begada	Vanasapathi	sundavinodini	Desh	Kapinarayani	Mayamalavagowla	Kandanakuthuhalam
Begada	**92**	58	58	62	65	65	75
Vanasapathi	58	**92**	63	70	72	75	76
sundavinodini	58	68	**92**	68	70	68	68
Desh	62	70	76	**92**	75	78	72
Kapinarayani	65	72	70	85	**92**	80	82
Maya malavagowla	65	75	68	78	80	**92**	80
Kandanakuthuhalam	90	76	68	72	82	80	**92**

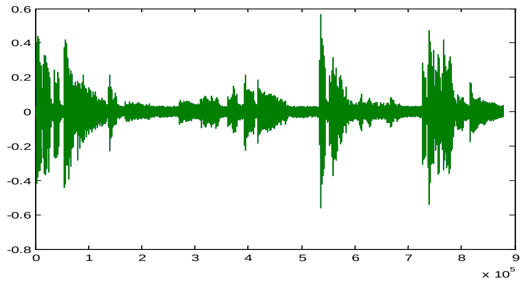

Fig 11 : Plot file for begada raaga (20sec for testing):

Testing Time: First 20 seconds

Table 7 : Confusion matrix indicating that the first 20 sec of each raagas are tested.

Stimulus	Recognized Raagas (%)						
	Begada	Vanasapathi	sundavinodini	Desh	Kapinarayani	Mayamalavagowla	Kandanakuthuhalam
Begada	**92**	58	58	62	65	65	75
Vanasapathi	58	**92**	63	70	72	75	76
sundavinodini	58	68	**92**	68	70	68	68
Desh	62	70	76	**92**	75	78	72
Kapinarayani	65	72	70	85	**92**	80	82
Mayamalavagowla	65	75	68	78	80	**92**	80
Kandanakuthuhalam	90	76	68	72	82	80	**92**

V. Conclusion and Future Work

The system for automatic raaga identification is presented in this paper it is based on Hidden Markov Models (HMM), Mel Frequency Cepstral Coefficients (MFCCs) and string matching. A very important part of the system is its note transcriptor, for which two (heuristic) strategies based on the pitch of sound are proposed. The strategy adopted is significantly different from those adopted for similar problems in Western and Indian classical music. Our problem, however, is different. Hence, probabilistic automata constructed on the basis of the notes of the composition are used to achieve our goal.

An attempt has also been made to determine the effectivity of Hidden Makov Model classifier and string matching algorithm is discussed for raaga recognition system that was developed. Raagas from instruments that are under four categories have been considered. The recognized raagas are presented in a confusion matrix table based on samples collected. As detailed in the experimental results, a high degree of accuracy of nearly 92% is achieved for recognized raagas of trained set where as an accuracy of around 70% reported for other raagas from outside sets. The HMM classifier, thus, achieved a better performance by in recognizing raagas from trained inputs, but able to differentiate when it the input parameters are other raagas from outside. although the approach used in the system is very general, there are two important directions for future research. In first place, a major part of the system is based on heuristics. There is a need to build this part on more rigorous theoretical foundations. Secondly, the constraints on the input for this system are quite restrictive. The two most important problems which must be solved are estimation of the base frequency of an audio sample and multiphonic note identification. Solutions to these problems will help improve the performance and scope of the system.

References References Referencias

1. Bhatkande.V (1934), "Hindusthani Sangeet Paddhati.Sangeet Karyalaya", 1934.
2. Chordia.P (2006), Automatic raag classi_cation of pitch-tracked performances using pitch-class and pitch-class dyad distributions", In Proceedings of International Computer Music Conference, 2006.
3. D. Weenink: "Praat: doing phonetics by computer": Institute of Phonetic Sciences, University of Amsterdam (www.praat.org)
4. M. Choudhary and P. R. Ray: "Measuring Similarities Across Musical Composi- tions: An Approach Based on the Raga Paradigm": Proc. International Workshop on Frontiers of Research in Speech and Music, pp. 25- 34.
5. L. R. Rabiner: "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition": Proc. IEEE, Vol. 77, No. 2, pp. 257-286: February 1989.
6. C. S. Iliopoulos and M. Kurokawa: "String Matching with Gaps for Musical Melodic Recognition": Proc. Prague Stringology Conference, pp. 55-64: 2002.
7. H. V. Sahasrabuddhe: "Searching for a Common Language of Ragas": Proc. Indian Music and Computers: Can 'Mindware' and Software Meet?: August 1994.
8. R. Upadhye and H. V. Sahasrabuddhe: "On the Computational Model of Raag Music of India": Workshop on AI and Music: 10th European Conference on AI, Vienna: 1992.
9. G. Tzanetakis, G. Essl and P. Cook: "Automatic Musical Genre Classification of Audio Signals": Proc. International Symposium of Music Information Retrieval,pp. 205-210: October 2001.
10. Ghias, J. Logan, D. Chamberlin and B. C. Smith: "Query by Humming – Musical Information Retrieval in an Audio Database": Proc. ACM Multimedia, pp. 231-236: 1995.
11. H. Deshpande, U. Nam and R. Singh: "MUGEC: Automatic Music Genre Classi- fication": Technical Report, Stanford University: June 2001.
12. S. Dixon: "Multiphonic Note Identification": Proc. 19th Australasian Computer Science Conference: Jan-Feb 2003.
13. W. Chai and B. Vercoe: "Folk Music Classification Using Hidden Markov Models": Proc. Internation Conference on Artificial Intelligence: June 2001.
14. J. Viterbi: "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm": IEEE Transactions on Information Theory, Volume IT-13, pp.260-269: April 1967.

15. L. E. Baum: "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes": Inequalities, Volume 3, pp. 1-8: 1972.
16. L. E. Baum and T. Petrie: "Statistical inference for probabilistic functions of finite state Markov chains": Ann.Math.Stat., Volume 37, pp. 1554-1563: 1966.
17. Gaurav Pandey, Chaitanya Mishra, and Paul Ipe "TANSEN: a system for automatic raga identification"
18. Prasad et al. "Gender Based Emotion Recognition System for Telugu Rural Dialects using Hidden Markov Models" Journal of Computing: An International Journal, Volume2 ,Number 6 June 2010 NY, USA, ISSN: 2151-9617
19. Tarakeswara Rao B et. All "A Novel Process for Melakartha Raaga Recognitionusing Hidden Markov Models (HMM)", International Journal of Research and Reviews in Computer Science (IJRRCS), Volume 2, Number 2,April 2011, ISSN: 2079-2557.
20. L.Arslan, Speech toolbox in MAT LAB, Bogazici University,http://www.busim.ee.boun.edu.tr/ ~arslan/