# real-time Raag Recognition for Interactive Music

Parag Chordia
Georgia Institute of Technology
Music Technology Group
840 McMillan St
Atlanta, GA
ppc@gatech.edu

Alex Rae
Georgia Institute of Technology
Music Technology Group
840 McMillan St
Atlanta, GA
arae3@gatech.edu

## ABSTRACT

We describe a system that can listen to a performance of Indian music and recognize the *raag*, the fundamental melodic framework that Indian classical musicians improvise within. In addition to determining the most likely *raag* being performed, the system displays the estimated the likelihood of each of the other possible *raags*, visualizing the changes over time. The system computes the pitch-class distribution and uses a Bayesian decision rule to classify the resulting twelve dimensional feature vector, where each feature represents the relative use of each pitch class. We show that the system achieves high performance on a variety of sources, making it a viable tool for interactive performance.

## Keywords

*raag*, *raga*, Indian music, automatic recognition

## 1. BACKGROUND AND MOTIVATION

*Raag* classification has been a central preoccupation of Indian music theorists for centuries [1], reflecting the importance of the concept in the musical system. The ability to recognize *raags* is an essential skill for musicians and listeners. The *raag* provides a framework for improvisation, and thus generates specific melodic expectations for listeners, crucially impacting their experience of the music [5]. Recently, MIR researchers have attempted to create systems that can accurately classify short *raag* excerpts [3, 2]

*Raag* 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 the essential qualities of a *raag*. Most common descriptions of *raags* are not sufficient to distinguish them. For example, many *raags* share the same notes, and even similar characteristic phrases. While ascending and descending scales are sometimes used as fairly unique descriptors, they are highly abstracted from the real melodic sequences found in performances. It takes a performer lengthy practice to fully internalize the *raag* and be able reproduce it. This is despite the fact that humans are highly adept at pattern recognition and have little problem with pitch recognition, even in harmonically and timbrally dense settings. Clearly, there are several difficulties for an automatic system.

The most obvious method for automatic *raag* recognition would be to attempt some kind of discrete note transcription and then look for known phrases [6]. Unfortunately such an approach is highly sensitive to inaccuracies in pitch tracking that lead to the insertion or deletion of notes. Pitch tracking in real contexts is difficult because of accompanying instruments and the prevalence of fast, continuous motion between notes. For plucked instruments such as the sitar and sarod, strumming patterns on drone strings are interspersed with the main melodic line. All these factors require either a much more sophisticated pitch tracking algorithm or the use of more advanced machine learning techniques.

Previous work by Chordia [3] showed that pitch class distributions (PCDs) and to a lesser extent pitch-class dyad distributions are relatively robust and easily computed features that can be used for accurate *raag* classification. Having established the effectiveness of such techniques, we wanted to adapt this work for real-time situations involving the interplay between live performers and computer systems.

The benefits of adopting a real-time approach are severalfold. First, there are greater opportunities for visualizing the activity of the system, as one can watch classification decisions evolve in response to audio input, whether live or prerecorded. This in turn provides an opportunity to better understand the behavior of the algorithm, particularly temporal dynamics. In previous work, decisions were made based on isolated segments of fixed length. Observing a continuously updated output allows us to see what events perturb the system, and to understand when ambiguity and clarity accurately reflect the source material or result from inherent limitations of the classification system. Finally, a real-time implementation opens up a wide realm of interactive music applications, in which the system can flexibly respond to a performer at a level that is highly abstracted but nonetheless musically relevant.

## 2. METHOD

Figure 2 shows the block diagram for the system. A sound source is selected and the system begins to listen and pitch track the performance. The pitch estimates are aggregated, weighted by peak amplitude, and counted by pitch-class to create the PCD. This is continuously updated (at the frame rate) and sent to the classifier, which returns the posterior probability of each *raag*. The strength of the posteriors is visualized using the graph shown in Figure 2. Depending on the performance context, the information about the strength of activation of the different *raag* classes would be used to control the musical output of the system.

The system is coded in Java and is currently implemented as a Java external for Max/MSP. The Max environment was chosen primarily for convenience in handling the au-


*NIME08,* Genova, Italy

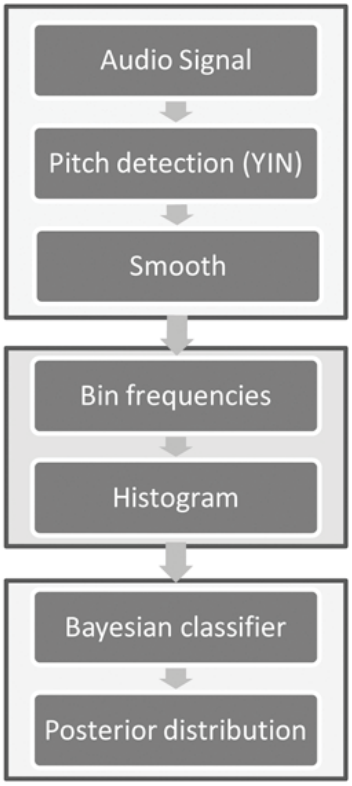


**Figure 1: Block diagram of *raag* recognition system**

dio environment (selecting various sound sources, tuning an oscillator, etc.). With few modifications, a stand-alone application will be developed.

The YIN algorithm is used for pitch tracking [4]. This method is based on finding periodicities in the time-domain signal through a method similar to autocorrelation. It differs in that instead of multiplying time-lagged versions of the signal with the original, the YIN algorithm is built on the squared difference function, along with several other optimizations that have been shown to collectively increase the accuracy of the estimate [4].

As the signal enters the system it is divided into frames, typically of size 1024 samples given a sample rate of 44.1 kHz. A hop parameter determines how frequently new pitch estimates are generated, usually overlapping frames by 50%. The raw pitch track is smoothed by disallowing large jumps that immediately return to the approximate previous pitch value. The smoothing uses a configurable threshold for what is considered an unacceptably large jump, expressed as a ratio. The peak amplitude of the frame is used to weight the pitch estimate in the next step.

To create the PCD, pitch estimates are fed into a block that converts frequency values to pitch-classes. In this context, pitch classes are considered as scale degrees, since the tonic can vary from performance to performance. Thus it is necessary to provide the frequency of the tonic to the system. This is done manually, or alternatively the performer can simply play a note to the system that will be pitch tracked and used as the root. The positions of the scale degrees are derived from a just-intoned scale, chosen over equal temperment due to prevailing practice and lack of harmonic modulation in Indian music. It should be noted that there are twelve fixed positions within the octave in NICM, despite microtonal inflections.

The bin boundaries are set by a configurable percentage so that frequencies within some distance of the center frequency for a given scale degree are aggregated (e.g. a value of .5 creates contiguous bins, .25 quarter-tone bins, and so on). The frequency values for the bin boundaries are calculated in the log domain. Each incoming frequency value is adjusted so it falls within a central pitch range by multiplying or dividing by a power of two. Then the binned frequency values give a twelve-dimensional vector representing the weighted frequency with which each pitch-class is used. Because it is calculated at the frame-level, duration is automatically taken into account; a held tone will occur in many frames, giving it more weight.

The classification is performed using a Bayesian classifier. WEKA, a commonly used general-purpose machine learning package, was used for this purpose [8]. PCD vectors from each *raag* are used to learn the multivariate Gaussian class-conditional probability distributions as well as the prior probabilities. Training was performed on a database of *raags* containing thirty-one different targets totaling 20 hours of recorded material. The PCDs had been previously calculated using a non-real-time method with a slightly different pitch detection algorithm [7].

When a new case is presented, as through playing a soundfile, the stored models are used to classify the given instance. In our case, we were interested primarily in the posterior distribution, and not just the most likely *raag*, which was the reason for using a Bayesian classifier rather than a Support Vector Machine that gave very high accuracy rates in previous work. The posterior distribution was displayed as a bar graph, and was updated at the hop rate, providing a smoothly changing picture of the classifiers best estimates of the target *raag*. The PCD was displayed in a similar manner within the same window, allowing for immediate comparison.

## 3. RESULTS

In previous work we formally evaluated the system using a cross-validation procedure and established that classification accuracy for PCDs using thirty-one targets on thirty second frames was 76%. In this work, we were more interested in qualitative evaluation of the system, particularly its suitability for live performance settings. This led us to examine how stable the estimates were, how much they fluctuated over time, as well as the conditions under which the system failed.

We evaluated these questions in two parts. One involved the first author, a trained Indian classical musician, performing twelve of the thirty-one *raags* for several minutes each on sarod. A second evaluation used eight vocal recordings, of an average length of ten minutes, from the *raag* database. In both cases the system was presented with unaccompanied performance, although in the case of the sarod, as noted above, drone and sympathetic strings also sounded.

The majority of *raags* presented in the first test were detected quickly and accurately, usually finding the correct label within fifteen seconds and becoming stable within thirty to forty seconds. There were a number confusions, mostly of two distinct types. *Raags* with similar scale types frequently led to the posterior distribution not converging completely onto one choice; often the probabability mass, rather than being evenly distributed, would fluctuate between different choices. The other main type of confusion involved *raags* that were quite dissimilar in terms of scale type, but shared one or two prominent notes, what one might call the dominant note effect. To illustrate, we describe three examples.

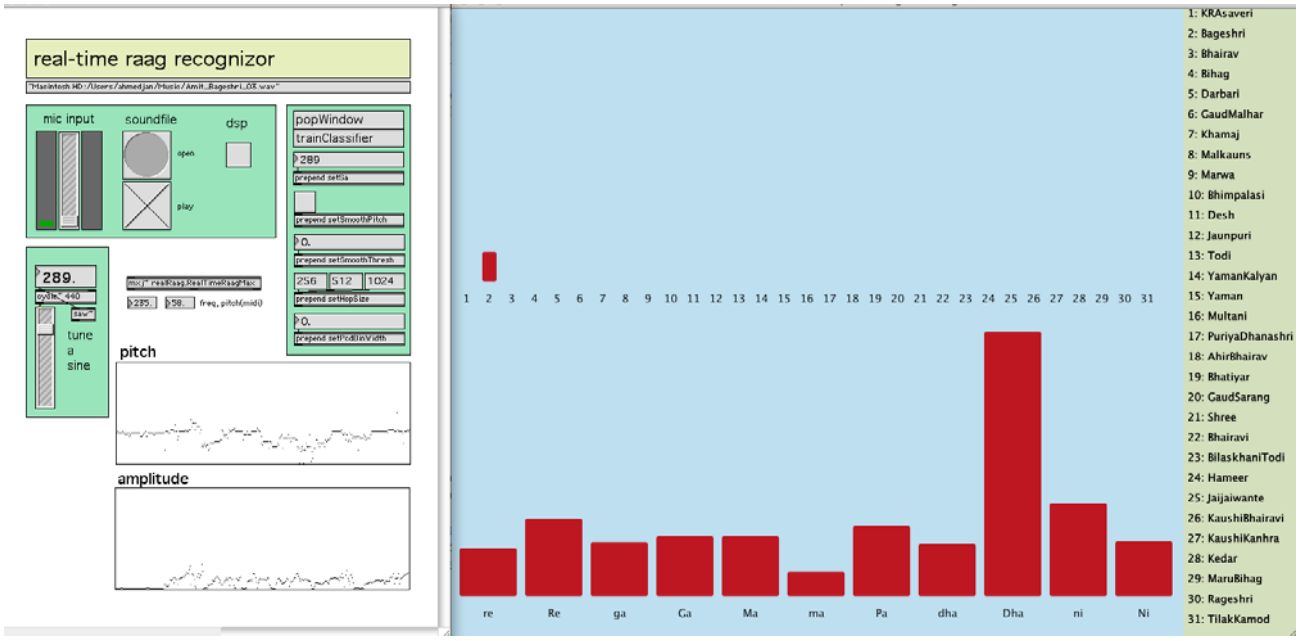


**Figure 2: The left panel shows a screen shot of the java external embedded in the Max/MSP environment. The user is given access to various parameters that control the algorithm. The pitch track and amplitude of the signal are displayed here. The right panel shows the display of the posterior distribution over the *raag* targets and the pitch class distribution.**

The system correctly identified *raag Desh* as performed by the author in less than fifteen seconds, and the posterior became locked within thirty. There was some slight confusion with Jaijaiwante, which is similar in both scale and phraseology. In the first few seconds, when the performer had played only two or three notes, with a prominent major seventh scale degree, *raag Yaman* had the highest posterior, however with the introduction of more material, the estimate quickly shifted to *Desh*. As a special case we were interested to see whether the introduction of *raag Jaijaiwantes* signature phrase, which includes the minor third scale degree not present in *raag Desh*, would be immediately detected. In fact, the change in the PCD brought on by this phrase did indeed produce a nearly immediate shift.

*Raag Kedar* (also performed live) provided a challenge: it is a *raag* that is primarily distinguished by its zig-zagging phrases, and has a major scale similar to several other *raags*. Not surprisingly, the system took longer to converge, and for the first two to three minutes *Kedar* was confused with *Maru Bihag*, *Jaijaiwante*, and *Darbari*. The first two were unsurprising due to their similar scales, however *Darbari* contains a minor third and minor sixth, neither of which are present in *Kedar*. We observed that this confusion occurred when the fifth scale degree was the most prominent, an example of the dominant note effect.

*Raag Darbari*, as sung by Amit Mukherjee, provided an interesting example of tetrachordal ambiguity being resolved. Often a performer will focus on half of the scale for a period of time. In this case, the singer lingered on the upper four notes of the scale for nearly forty seconds before including the other notes. During this time, *Malkauns* had by far the highest posterior, an unsurprising confusion given the two *raags*' similarity in the upper tetrachord. However, the system immediately and unambiguously switched to *Darbari* upon hearing notes from the lower tetrachord.

It should be noted that in some cases the system was unable to converge to the correct *raag*, or only briefly touched upon it before converging elsewhere. For example, *Bhimpalasi* was confused with *Darbari* and *Bageshri*, and at times with *Malkauns*, and was never correctly identified.

## 4. CONCLUSIONS

We have demonstrated that real-time *raag* recognition is possible in realistic performance situations with minimal adjustments needed for different performers. In terms of accuracy we noted that as additional information is collected, initial ambiguities are often resolved leading to correct classification. In some cases, however, as the performer begins to focus on certain phrases that overlap with nearby *raags*, the estimate begins to fluctuate between different possibilities. In the latter case, this error is in part attributable to our feature choice, which does not include any sequence information. This make the system biased towards simply using the most commonly occurring pitches for the classification decisions, without consideration of the melodic context. This suggests that if sequence were taken into account, such as by counting pitch dyads, we would be able to more clearly disambiguate *raags* in the same neighborhood. This is consistent with what we found earlier in our non-real-time experiments where pitch-dyads were also used as features.

Another point that becomes clear is that the algorithms lack a note model makes it very difficult for it to estimate the perceptual salience of a pitch. For example, for a human listener, the fleeting introduction of a new note can dramatically change the tonal landscape even though its effect on the PCD is minimal, perhaps indistinguishable from noise. Another example is the use of gliding approaches to notes that are very common in Indian music. For our system, such glides introduce energy at all the frequency bands between the starting and ending point, and for a plucked instrument that decays quickly, may emphasize the starting point. This leads to a noisier PCD estimate. However, for a human the opposite is often true. Glissandi serve to emphasize the target note. In both these cases, the crucial

difference is that the system has no concept of what constitutes a note. Humans are able to integrate many different types of information to resolve notes from complex time-varying pitch tracks, from the basic perception of vibrato as a modulation of a central frequency to understanding the importance of a pitch within the musical structure.

These insights were suggested concretely by our ability to visualize inner working of the algorithm, both the data presented to it and its hypotheses, while simultaneously listening.

We have created a framework for high-level musical interaction for Indian electroacoustic music. We envision a number of extensions to this system to allow it to be used effectively in a performance setting. Aside from tracking more features to yield different and more robust classification estimates, the momentary maximum posterior can be used for example to introduce phrases from the same or related *raags*. The time-varying posteriors can also be treated as modulating control signal that tracks a very high-level aspect of the music. The time-varying PCDs can be used in a similar fashion. These signals can be used to generate tonally relevant material.

## 5. FUTURE WORK

Improving the pitch estimation algorithm would likely yield better results. The primary difficulties faced are tracking pitches that vary rapidly over time, which is typical of Indian music, eliminating pitches that are not part of the main melodic line, and source separation from accompaniment and resonating strings.

Future systems will attempt to use simple sequential information such as pitch class dyad distributions as well, which would require segmenting the signal into notes using an onset detector. This is particularly difficult in Indian music where notes do not often correspond to clear onsets in the time domain. If solved, this would partially address the lack of a note model referred to above, and would likely lead to a substantial increase in accuracy. However, creating a perceptual note model remains a fundamental and interesting problem.

## 6. ACKNOWLEDGMENTS

The authors would like to acknowledge the following musicians who made substantial contributions to the *raag* database: Prattyush Banerjee (sarod), Nayan Ghosh (sitar), Amit Mukherjee (vocal), Sugato Nag (sitar), Falguni Mitra (vocal), Manilal Nag (sitar).

## 7. REFERENCES

[1] V. Bhatkande. *Hindusthani Sangeet Paddhati*. Sangeet Karyalaya, 1934.

[2] P. Chordia. Automatic raag classification of pitch-tracked performances using pitch-class and pitch-class dyad distributions. In *Proceedings of International Computer Music Conference*, 2006.

[3] P. Chordia and A. Rae. Raag recognition using pitch-class and pitch-class dyad distributions. In *Proceedings of International Conference on Music Information Retrieval*, 2007.

[4] A. de Cheveigne and H. Kawahara. Yin, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America*, 111(4):1917 – 1930, 2002.

[5] D. Huron. *Sweet Anticipation: Music and the Psychology of Expectation*. MIT Press, 2006.

[6] H. Sahasrabuddhe and R. Upadhy. On the computational model of raag music of india. In *Proc. Indian Music and Computers: Can Mindware and Software Meet?*, 1994.

[7] X. Sun. A pitch determination algorithm based on subharmonic-to-harmonic ratio. In *In Proc. of International Conference of Speech and Language Processing*, 2000.

[8] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.