Joint Recognition of Raag and Tonic in North Indian Music

Parag Chordia* and Sertan Şentürk†
*Smule
577 College Avenue
Suite 200
Palo Alto, California, 94306 USA
parag@smule.com
www.paragchordia.com
†Music Technology Group
Universitat Pompeu Fabra
Roc Boronat, 138
08018 Barcelona, Spain
sertan.senturk@upf.edu
www.sertansenturk.com

Abstract: In many non-Western musical traditions, such as North Indian classical music (NICM), melodies do not conform to the major and minor modes, and they commonly use tunings that have no fixed reference (e.g., A = 440 Hz). We present a novel method for joint tonic and raag recognition in NICM from audio, based on pitch distributions. We systematically compare the accuracy of several methods using these tonal features when combined with instance-based (nearest-neighbor) and Bayesian classifiers. We find that, when compared with a standard twelve-dimensional pitch class distribution that estimates the relative frequency of each of the chromatic pitches, smoother and more continuous tonal representations offer significant performance advantages, particularly when combined with appropriate classification techniques. Best results are obtained using a kernel-density pitch distribution along with a nearest-neighbor classifier using Bhattacharyya distance, attaining a tonic error rate of 4.2 percent and raag error rate of 10.3 percent (with 21 different raag categories). These experiments suggest that tonal features based on pitch distributions are robust, reliable features that can be applied to complex melodic music.

Introduction

We begin by discussing the motivation and musical context for raag recognition. This is followed by a discussion of the significance of pitch distributions in our work, and an introduction to raag.

Motivation

Pitch is fundamental to most of the world's musical traditions, and humans utilize a rich set of cognitive representations for pitch processing. For centuries, music theorists have noted that the musical effect of a pitch is in large part determined not by its absolute frequency but rather by its relationships to other pitches, typically those nearby in both in pitch space and in time (Aldwell and Schachter 2002). Over the past three decades, psychological researchers have confirmed that pitches derive their meaning in large part from their relationship to a tonal center (tonic), the most fundamental pitch, and from their relative frequency of occurrence within a piece of music. Moreover, high-level responses, such as emotion, can be related to the pitch properties of the music (Juslin and Sloboda 2001). Thus, tonic and mode, a set of pitches defined by their interval from the tonic pitch, are fundamental properties of many types of music and have real psychological relevance. For automated systems whose goal is to understand aspects of music in order to, for example, provide recommendations to listeners or interact with human performers, such information is critical. In this article, we describe a system that is capable of automatic tonic and raag recognition in North Indian classical music (NICM). Raag is the fundamental melodic form in NICM; as with the Western concept of mode, a raag defines a set of pitch-intervals, or scale, relative to the tonic, but raag also encapsulates a more specific manner of playing, including characteristic phrases and articulations, such that two distinct raags may share the same set of pitches. We describe raag in more detail in the Introduction to Raag section.

Tonal Hierarchy

It has been shown through a variety of experimental paradigms that listeners are sensitive to the relative

Computer Music Journal, 37:3, pp. 82–98, Fall 2013
doi:10.1162/COMJ_a_00194

distribution of pitches in music. The probe-tone method pioneered by Krumhansl and Shepard (1979) demonstrated that judgments of "fittingness," in response to a short excerpt that clearly defined a key, show a hierarchical response, with tones more commonly used in that key being judged as better fitting; probe-tone profiles for a given key are similar to the distribution of pitches of common-practice music in the relevant key. This observation led to automatic key-finding algorithms based on pitch-class distributions (PCDs) (Krumhansl and Kessler 1982; Gomez 2006; Temperley and Marvin 2008). These techniques have been highly successful for identifying the key of polyphonic music from symbolic data (Huron and Parncutt 1993). Subsequently, they have been applied to the closely related task of symbolic chord recognition (Temperley 2001).

Because it is still not possible to reliably estimate pitches in polyphonic audio, the application of PCDs to audio is based on a proxy for the PCD called the chroma feature (Fujishima 1999; Gomez 2006). This is computed by "folding" the spectral data into one octave and summing the energy in twelve bins, usually centered around the equally tempered chromatic pitches given a reference tuning (Pauws 2004). The chroma feature is based on the idea that the first several partials of pitched tones that are exact or nearly exact multiples of the fundamental will "fold" back to the fundamental and the fifth and third scale degrees above it, forming a distinctive pattern. Obviously, such a method is susceptible to inharmonic partials, which will spread energy between adjacent bins, as well as strong third partials, which will lead to confusion between the fundamental and the fifth. This ad hoc feature, however, has proved surprisingly useful for algorithms recognizing keys and chords in polyphonic music from audio (Fujishima 1999; Bartsch and Wakefield 2001; Pauws 2004; Bello and Pickens 2005; İzmirli 2005; Lee 2006). These algorithms typically utilize nearest neighbor (NN) classifiers (Fujishima 1999), or explicit statistical modeling of the chroma vectors (Lee 2006). This approach has also been used with some success in early work on raag recognition (Chordia 2004, 2006).

In monophonic music, it is possible to compute a PCD directly from audio that has been pitch-tracked. For music with continuous pitch motions, however, the PCD will be noisy because of time spent between notes of the scale. In previous work (Chordia and Rae 2007, 2008) we demonstrated that these techniques could be applied to recognizing a large set of modal types (raags), even in cases where the underlying scales (i.e., pitch sets) were the same. In that work, the true tonic was given and input and was not estimated as part of the recognition problem. Recently, Gedik and Bozkurt applied similar techniques to the recognition of modal types (*makams*) in Turkish music (Bozkurt 2008; Gedik and Bozkurt 2009, 2010). They used a template-matching algorithm using a fine-grained pitch distribution (FPD; cf. Akkoç 2002), with bins whose width was 1/3 Holdrian comma, i.e., about 7.5 cents (hundredths of a semitone), resulting in a 159-dimensional feature-vector, compared with the standard twelve-dimensional PCD. Additionally, the tonic was automatically detected by cross-correlating FPD with a *makam* template; the lag corresponding to the maximum was taken as the tonic. During the tonic detection phase, it was assumed that the *makam* was known. FPDs have also been used to infer the scale, in a collection of African music, by identifying peaks in the FPD (Moelants, Cornelis, and Leman 2009). The current work is also closely related to recent work in the area of automatic vocal accompaniment undertaken by Cao and Chordia, in which an FPD with template matching was used for key detection (Cao 2009).

Introduction to Raag

Almost all North Indian classical music is organized around the melodic abstraction known as raag (sometimes seen in other transliterations, such as "raga"). A raag is most easily explained as a collection of melodic gestures, along with techniques for developing them. The gestures are sequences of notes that are often inflected with various micro-pitch alterations and articulated with an expressive sense of timing. Longer phrases are built by joining these melodic atoms together.

Although there is considerable continuous pitch motion, due to the way notes are connected and ornamented, it is nevertheless accurate to consider raag melodies to be composed from a discrete set of pitches. These notes are generally drawn from a chromatic scale of twelve pitches tuned in just intonation (see Jairazbhoy 1971). In some cases, however, the scale may more closely resemble equal temperament and, rarely, a raag may contain notes that do not lie within either of these tunings. There are almost no raags in which stable, held tones fall outside of this chromatic scale. Micro-pitch structure, however, is often essential, and the same nominal note may take on a different character depending on how it is articulated. In some raags, there are consistent pitch-time trajectories that are essential to the character of the raag.

The frequency of the tonic pitch is set by the performer based on the constraints of the instrument or voice, or simply on the preference of the performer. The tonic pitch is not varied over the course of the performance and is often sounded continuously by a drone instrument. The primary pitch representation is scale degrees; notes are referred to by their north Indian solfege syllables. There are seven such syllables, some of which may be raised or lowered to cover the chromatic scale.

The presentation of raag typically proceeds in several sections. In the first section, the main melodic instrument, accompanied only by the drone, slowly develops the melodic framework. In later sections, the emphasis shifts to faster sequences of notes, leaving behind most of the subtleties of pitch articulation, and the soloist is usually accompanied by tabla (the pair of hand drums that together constitute the main percussion instrument of NICM). In both cases, the characteristic phrases of the raag are often repeated with variations. The notes used in these phrases therefore determine the relative prevalence of various scale degrees in the piece, or in some local context. This hierarchy among scale degrees is heightened by pausing on phrase-ending notes, as well as through repetition. Abstractly, this is similar to the creation of tonal hierarchies in Western music through the use of chords and repeated melodic tones.

The idea of a level of representation more abstract than the phrase level is an old concept in NICM. The collection of tones used in all the phrases that make up the raag constitute the scale. Raags have been categorized by scales (called *melas* and later *thaats*) for several centuries. A hierarchy of tones has also been described: The most stressed note is called the *vadi* and the second most stressed, traditionally a fifth or fourth away, is called the *samvadi*. There are also less commonly used terms for tones on which phrases begin and end. A typical summary of a raag includes its scale type, *vadi*, and *samvadi*. To capture some further nuance, ascending and descending scales are often given, capturing the typical upward and downward motions that the phrases define. (See Jairazbhoy 1971 for a more detailed presentation of this summary.)

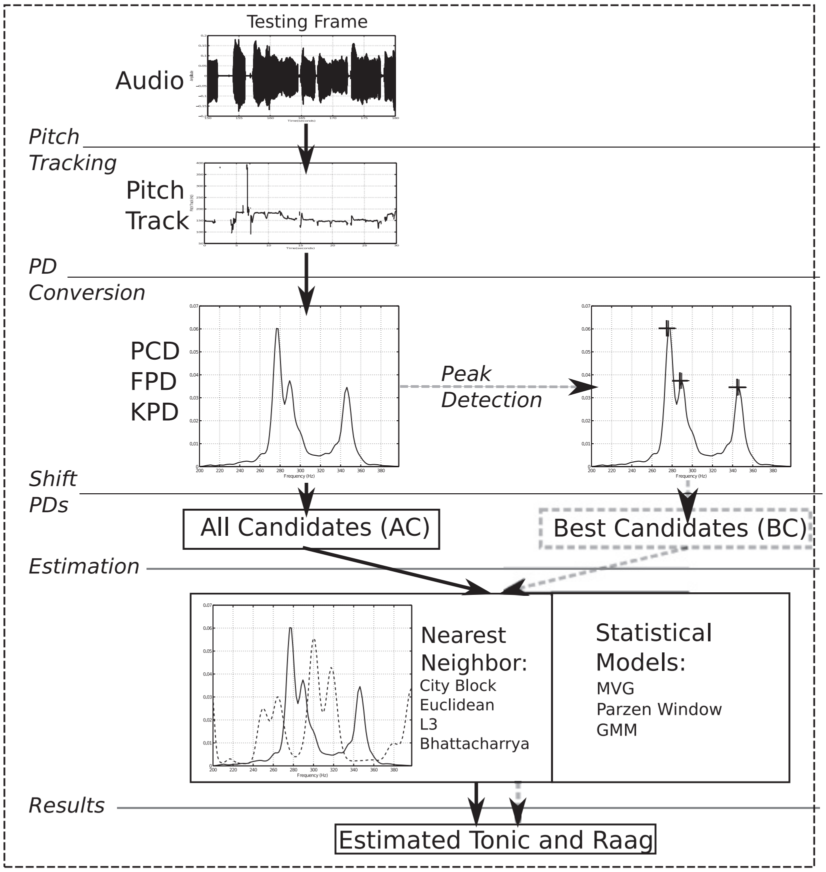
To some extent, these traditional concepts can be viewed as anticipating a modern representation of the tonal hierarchy, namely, the pitch-class distribution (PCD), which gives the relative frequency of each scale degree, possibly weighted by duration or loudness.

Analysis of North Indian Classical Music

NICM thus presents several challenges for automatic analysis: (1) diversity and complexity of melodic types (raags); (2) prevalence of continuous pitch motions; and (3) arbitrarily tuned tonic pitch. The current research seeks to address these issues through the use of tonal representations based on pitch distributions, and particularly pitch distributions that are more continuous than the standard PCD. The main contributions of this work are (1) modeling the pitch distribution of a frame non-parametrically, resulting in a more robust feature for classification; (2) a joint method for tonic and raag recognition; and (3) systematic testing on a large, real-world database to establish robustness of the proposed approach. To the best of our knowledge, this is the first example of a system that is able to jointly recognize tonic and raag in a fully automatic way.

Thus, the novelty of this work, compared with template-based key finding algorithms (İzmirli

Figure 1. Block diagram of the algorithm for joint tonic and raag detection. PCD, HPD, and KPD stand for pitch class distribution, high-dimensional pitch distribution, and kernel-density pitch distribution, respectively. Dotted lines in the bottom graph illustrate a candidate pitch distribution from the database, which is being compared to the analyzed audio's pitch distribution (solid lines).



2005), is the use of the high-dimensional pitch feature, and, specifically, the kernel-based method discussed herein, which avoids certain limitations of a simple histogram approach. Further, compared with detection of Western key and mode, raag recognition is more difficult because there are many more raags than modes. Indeed, many raags use the same set of notes. Compared with the work of Gedik and Bozkurt described earlier, the current work not only adds its novel kernel-based pitch feature, but also considers a greater variety of classification methods, such as statistical classifiers in addition to instance-based classifiers, and it jointly estimates the raag and tonic.

Finally, our use of maximum a posteriori techniques to jointly find the tonic and raag has not, to the best of our knowledge, been utilized before.

Method

The goal of the system is to automatically determine the raag and tonic pitch, given a short audio excerpt of around 30 seconds. In this work, we restrict ourselves to monophonic recordings in which accompanying instruments have been removed from the original multitrack recordings. Figure 1 shows a block diagram of the system. Each audio file is first pitch-tracked, resulting in a pitch-versus-time graph (described in the section on Pitch Recognition). Next, the pitch information is mapped to one octave and summarized by a pitch distribution. This becomes the tonal feature vector used for tonic and raag recognition (described in the section on Tonal Features). The pitch distribution is then compared with samples in the training database (nearest-neighbor) or with distributions summarizing each of the raags (statistical classifier). The comparison is made for various hypothetical candidates by a circular shift of the pitch-distribution vector. The tonic and raag are jointly labeled based on the best overall match.

It is worth clarifying what we mean by "joint" estimation. In our case, we are maximizing the joint probability of a certain tonic and raag, given the PCD. An alternative approach would be to first estimate the tonic and then the raag. By contrast, our approach simultaneously computes the tonic and raag. From the performer's standpoint these are unrelated, because any raag can be performed at any tonic. The PCD we observe, however, depends on the the interaction of the tonic and the raag. To be concrete, the local peaks in the PCD can be thought of as defining the intervals in the scale. But we don't know which peak is the true tonic, and the raag will differ based on this. Thus, there are often several tonic and raag combinations that explain the data. Rather than maximizing the probability for the tonic and the raag independently, the information in the pitch distribution can be better utilized by finding the tonic and raag combination that jointly maximize the probability of the observed PCD.

Pitch Recognition

For pitch-tracking the database, a sawtooth-waveform-inspired pitch estimator (SWIPE′) algorithm is used (Camacho 2007). Because SWIPE′ is not widely known, we briefly summarize the algorithm here. First, the basic SWIPE algorithm estimates the pitch as the fundamental frequency of the sawtooth waveform whose spectrum best matches the spectrum of the input signal. SWIPE computes the similarity between the square root of the spectrum of the signal and the square root of the spectrum of a sawtooth waveform, using a pitch-dependent optimal window size. SWIPE′ is an improvement on the basic SWIPE algorithm. It uses only the first and second harmonics in computation, which gives a significant improvement by reducing subharmonic errors. Both SWIPE and SWIPE′ were compared against other pitch-tracking algorithms, and were found to outperform better known methods such as YIN (de Cheveigné and Kawahara 2002) and harmonic product spectrum (see Schroeder 1968).

While pitch-tracking a song, it is first divided into 30-second chunks, because of memory considerations. Each chunk is read into MATLAB and, in cases where the solo instrument was recorded in stereo, converted to mono. After processing, SWIPE′ is called, and the pitch of the chunk is estimated every 10 msec. The pitch estimate is kept within the range 73.4–587.2 Hz using a resolution of 48 steps per octave. The spectrum is sampled every 1/20 of an equivalent rectangular bandwidth for that frequency range. A window overlap factor of 50 percent is used. In addition to a pitch estimate, the SWIPE algorithms return an estimate of pitch strength, a value between zero and one. Pitch estimates less than 0.2 are deemed unreliable and are replaced with the floating-point value "not a number". Finally, the pitch tracks of each chunk of the particular song are recombined.

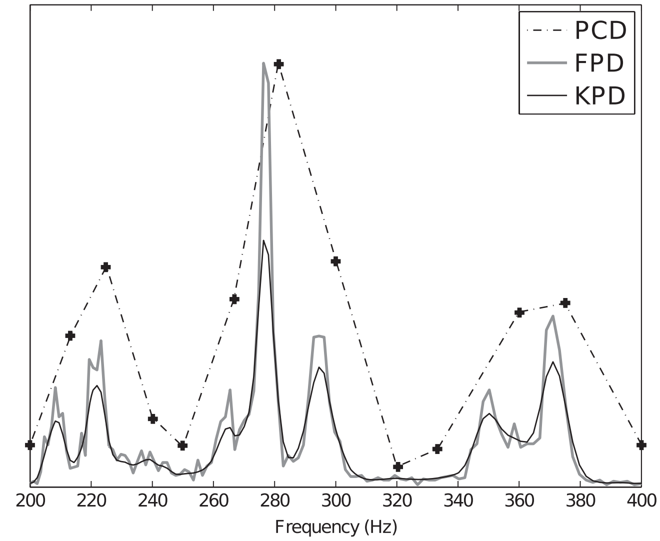
The pitch tracks of the songs are divided into non-overlapping frames. Frame lengths of 30, 60, and 120 seconds were compared in different experiments.

Tonal Features

Pitch tracks are used to obtain pitch distributions (PDs), which, along with the tonic frequency, constitute the main tonal features used in the raag-recognition task. Three different histograms are used, namely, a twelve-dimensional PCD, an FPD, and a kernel-density pitch distribution (KPD), the latter two being high-dimensional pitch distributions (HPDs). The two HPDs are calculated using resolutions of both 5 and 10 cents. Pitch tracks are stripped of unreliable pitch estimates during the PD calculations. A tonic must be known or assumed in order to calculate the bin placement for PDs; this is discussed in the Tonic Recognition section.

The PCD is computed by mapping pitch estimates to one of twelve chromatic pitch classes. Specifically, we first map all pitches into one octave by dividing or multiplying each pitch's frequency by 2^k for the value of k that places it

Figure 2. Different pitch distributions obtained from the same frame using the standard twelve-dimensional pitch class distribution (PCD), a fine-grained pitch distribution (FPD), and a kernel-density pitch distribution (KPD).



within the desired octave. The center of each of the twelve bins is given by the following just-intonation intervals for each chromatic tone: $\{1, 25/24, 9/8, 6/5, 5/4, 4/3, 45/32, 3/2, 8/5, 5/3, 9/5, 15/8\}$. The boundary of each bin is calculated in the log space so that each bin has a width of approximately 100 cents. The PCD is then normalized so that the values sum to 1, making them independent of the frame duration.

The FPD uses the same procedure as the PCD, but instead of 12, the number of bins is increased to 120 or 240, corresponding to bin widths of 10 cents and 5 cents, respectively. Compared with the PCD, the FPD is a much more continuous representation of the pitch distribution (see Figure 2). It can be seen, however, that the FPD contains many local maxima, making it difficult to quickly assess the location of stable tones. This motivated the use of a kernel-based approach, where the bin width could be precisely controlled without edge effects.

Kernel density estimation (KDE), also known as the Parzen window method, is used to compute the KPDs, essentially a continuous version of the pitch histogram. KDE is more typically used to compute probability densities based on observed samples. Instead of assigning a pitch value to a given bin, the kernel approach centers a window function, typically a symmetric, peaked curve such as a Gaussian, on the pitch value. The sum of all such curves gives the overall density. In other words, a kernel is convolved with a series of impulses located on each of the pitch values.

The kernel density is given by

$$\widehat{f_h}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{1}$$

where
K is the kernel with a kernel width of h,
x_i is the value of the ith pitch value, and
n is the total number of pitch values.

If a Gaussian is chosen for K, the whole equation can be rewritten as:

$$\widehat{f_h}(x) = \frac{1}{nh} \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-x_i)^2}{2h^2}} \tag{2}$$

In this work, the MATLAB KDE function `ksdensity` is used (MathWorks 2009). A Gaussian kernel is chosen with widths ranging from 0.01 to 2.2 Hz. The kernel width determines the amount of smoothing, with larger values corresponding to more smoothing. The density is first estimated over the full pitch range and then sampled at either 5-cent or 10-cent intervals, before being folded into one octave. This results in either a 240- or 120-dimensional feature vector, as with the FPD.

One potential advantage of HPDs over PCDs is that the representation is richer, allowing raag-defining characteristics to be expressed more precisely despite the lack of sequential information. For example, if a certain scale-degree tends to be performed with vibrato (*andolan*), this will lead to a wider peak. Similar arguments hold for other common ornaments, such as portamento (*meend*). For example, in raag Darbari it is common to slide from the lowered seventh scale degree to the fifth, which would lead to an increased area under the HPD curve to the left of the seventh scale degree. Because of this, HPDs can capture microtonal pitch structure in a limited way.

Tonic Estimation

The two tonic-estimation techniques that we use are based on calculating the PD for different tonic pitches and finding the one that gives the best match to the database. Because the PD is expressed in relative terms, the *x*-axis is the number of cents above the tonic, and changing the tonic leads to a new curve. We devised two methods for tonic estimation, which we term *all candidates* (AC) and *best candidates* (BC). In the AC method, it is assumed that we have no a priori information with which to pick the tonic. A brute force approach is used. First, the PD is calculated using an arbitrary tonic. This results in a 240- or 120-dimensional vector, depending on whether the bin width is set to 5 or 10 cents. To evaluate all possible candidates, we allow each bin to be the tonic. Formally, this is simply a circular shift of the PD. For example, if our original vector was $x_1, x_2, x_3, \ldots, x_{120}$, then a shift by 1 leads to the sequence $x_{120}, x_1, x_2, \ldots, x_{119}$. This new PD has a tonic that is 10 cents lower than the original. This process is repeated for all possible shifts leading to 120 PDs representing 120 different tonic hypotheses (or 240 in the case of 5-cent bin widths).

For each tonic candidate, the PD is compared with all samples in the training database, and the nearest neighbor is found. The candidate whose nearest neighbor has the minimum distance overall is taken as the tonic. We describe the distance metrics used in the Raag Recognition section. For the statistical classifiers, the tonic candidate that maximizes the posterior probability is taken as the tonic. This is done for each raag category to find the global maximum.

Inspection of the HPDs suggested the BC approach: Stable notes appear as peaks in the HPD. BC greatly reduces computation by considering only this reduced set of tonic candidates. First the HPD is computed using an arbitrary tonic. The seven highest peaks are found and peaks with a normalized height of less than 0.15 are discarded. The corresponding frequencies for the peaks are treated as the "best candidates." Finally, PDs are obtained for only this narrowed set of frequencies. When calculating the PCDs and FPDs, we use candidates obtained from the FPD of the arbitrarily chosen initial tonic; whereas in calculating the KPDs we use candidates from the KPD of the initial tonic. There are a few frames that yield no peaks in the BC method, because they are nearly silent or background noise, and so they are discarded.

Raag Recognition

During tonic estimation, the raag is simultaneously recognized. Depending on the classification technique, it is either simply the label of the overall nearest training sample, or else the raag category that gave the maximum posterior probability.

Nearest-neighbor Classification

Distance between PDs are measured using several metrics: city block, Euclidean, L3 norm, and Bhattacharyya distance.

Bhattacharyya distance is one of the most popular distance metrics for comparing two estimates of probability density. In the discrete case, the Bhattacharyya distance is given by the formula

$$D_B(p, q) = -\ln\left(\sum_{i=1}^{n} \sqrt{p_i q_i}\right) \tag{3}$$

where $p = (p_1, p_2, \ldots, p_n)$ and $q = (q_1, q_2, \ldots, q_n)$.

Statistical Classifiers

In addition to the instance-based classifiers, several Bayesian classifiers were constructed using different techniques to estimate the class-conditional probability density (CCPD). As usual, Bayes's rule (Duda, Hart, and Stork 2001) was used:

$$P(raag_i|x) = \frac{P(x|raag_i)P(raag_i)}{\sum_j P(x|raag_j)P(raag_j)} \tag{4}$$

where x is one of the testing PDs for a frame.

For each raag, the PD feature vectors were used to empirically estimate $P(x|raag_i)$. We used two parametric density models, multi-variate Gaussian (MVG) and Gaussian mixture models (GMM), as well as a non-parametric model estimated using the Parzen windows technique.

For the MVG, feature vectors from each raag are assumed to be samples from an n-dimensional Gaussian distribution. The mean, covariance, and prior probabilities for each category are calculated using a maximum-likelihood approach. The n-dimensional Gaussian distribution can be expressed as:

$$f_X(x) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \tag{5}$$

where
$|\Sigma|$ is the determinant of the covariance matrix,
Σ^{-1} is the inverse of the covariance matrix,
μ is the mean, and
T represents the matrix transpose.

Classification was done using the maximum a posteriori approach. For a test sample, the posterior probability for each raag was computed according to Equation 4, and the label of the highest probability raag was assigned. We used a diagonal covariance matrix, resulting in a naive Bayes classifier; for raags with relatively few examples, it was often not possible to estimate the full covariance matrix. Prior probabilities were calculated according to the relative number of samples of each raag.

The same approach was repeated using a GMM for the CCPD. The primary motivation for such an approach is to have the flexibility to model multimodal distributions. When performing a raag, a performer might focus for a long period of time on a cluster of notes, leading to a very different PD in one section when compared with another, leading to a multi-modal CCPD. A seven-component GMM was fit using a shared, diagonal covariance matrix. The GMM model was applied only to the PCD, because the HPDs were too high-dimensional, resulting in far too many free parameters relative to the training data. Again, classification was done using a MAP approach.

For the final classifier, we used KDE to estimate the CCPD. This is not to be confused with the use of KDE for fitting the one-dimensional pitch distribution for a given frame. Here, a high-dimensional density was estimated, summarizing information for all frames in a given raag. For simplicity, we assumed that the values of the PD were independent. As above, classification was done using a MAP procedure.

Experiments and Results

The following sections describe the set of raag-recognition experiments undertaken and the key results.

Database

In the project, we used the database GTRaagDB, available online at paragchordia.com/data/GTraagDB. The database consists of 127 pieces, encompassing 31 raags. Performances were recorded from 19 different artists and included both vocal and instrumental music (sitar and sarod). The durations of the pieces range from 3 to 60 minutes. Playback time of the entire database is over 20 hours. The pieces were recorded with accompanying instruments on separate audio tracks, but the accompaniment tracks have been removed so that the audio files only contain solo instrument or voice. The artists, instruments, annotated tonic and raag, and pitch tracks of each piece are also presented in the database.

For this study, the following raags were removed because each was represented in only one recording: Jaunpuri, Multani, Puriya Dhanashri, Bhatiyar, Gaud Sarang, and Tilak Kamod. The raag pair of Yaman Kalyan and Yaman were treated as equivalent, as were Kaushi Bhairavi and Kaushi Kanhra.

Tonic Estimation

In all of the following experiments, a tenfold cross-validation scheme was used to assess performance. When 60-sec frames were used, this led to 118 test frames and 1,059 training frames in each fold. Results for the three features (PCD, FPD, KPD) and seven classifiers are summarized in Table 1

Table 1. Average Tonic Error Rates: Features and Tonic-Estimation Methods Against Classifiers

	PCD		*FPD*		*KPD*	
	All Candidates	*Best Candidates*	*All Candidates*	*Best Candidates*	*All Candidates*	*Best Candidates*
City Block	69.67	30.67	13.00	29.33	10.33	14.50
Euclidean	74.50	31.33	13.83	30.50	13.50	17.17
L3	74.50	31.00	19.33	35.00	18.00	20.50
Bhattacharyya	62.17	24.33	7.33	23.83	8.00	12.33
MVG	72.83	33.17	26.33	31.67	28.67	24.67
Parzen	69.50	32.83	24.33	33.83	27.50	22.50
GMM	67.50	28.67				

This table, as well as Tables 2–4 and Figures 3–7, uses 120-sec frames, 10-cent granularity, and 15-cent precision, unless otherwise specified.

and Figure 3. For tonic estimation, the error rate is reported for a given strictness: 15-cent precision means that the estimated tonic was within ±15 cents of the annotated tonic. The KPDs used to obtain the results were calculated with a kernel width of 0.4 Hz unless stated explicitly. The complete results for all the experiments are available online at paragchordia.com/research/raag.html. For all results herein, the term "significant" means that the claim is statistically significant at the 0.01 level as determined by a multiple comparison test using the Tukey-Cramer statistic.

The minimum error rate for 15-cent precision was 4.92 percent, attained using the KPD feature with 5-cent granularity and a nearest-neighbor classifier using the Bhattacharyya distance (NNB) with a kernel width of 0.1 Hz. With the AC method, FPD and KPD features significantly outperformed PCDs, and the error rate typically increased only a few percentage points for KPD. For example, using 120-sec frames and 10-cent estimation granularity, and evaluated using 30-cent precision, the average error rate across all classifiers was 70.1 percent for PCD, 17.4 percent for FPD, and 17.7 percent for KPD, using the AC method.

For all features, the NNB classifier was most effective, with FPD (7.3 percent) and KPD (8.0 percent) again significantly outperforming PCD (62.2 percent) using the AC method. For FPD and KPD using the AC method, NN methods outperformed MVG and Parzen classifiers in every case. Within the NN methods, after Bhattacharyya, the city block distance was most effective, followed by Euclidean and L3.

The experiments compared the effect of using a granularity of 5 or 10 cents for the AC method (see Table 2). For HPDs using a NN classifier, there are typically only very slight performance gains in exchange for significantly longer run time, likely not justifying a doubling of the number of PDs that must be considered.

Overall, error rates decreased significantly as the frame size was increased from 30 to 60 sec (see Table 2). For example, the error rate of KPD using NNB decreased from 18.4 percent to 8.9 percent using the AC method. However, error rates decreased only marginally when the frame size was further increased to 120 sec, with the error rate for KPD using NNB decreasing from 8.9 percent to 8.0 percent. This suggests that, at least for these data, 60-sec frames are sufficient for tonic estimation.

To gain further insight into the best-performing tonic-estimation method (KPD using NNB), we looked at the effect of kernel width (see Figure 4). Using the AC method, for precision levels between 15 and 25 cents, performance was best for kernel widths from 0.04 to 1.4 Hz. When considering more stringent precisions, the range was somewhat narrower: 0.06 to 0.4 Hz. For the BC method, for all but the strictest precision, there was a clear local minimum around 0.6 Hz.

Figure 3. Average tonic error rates comparing features against classifiers for the all-candidates (AC) method.

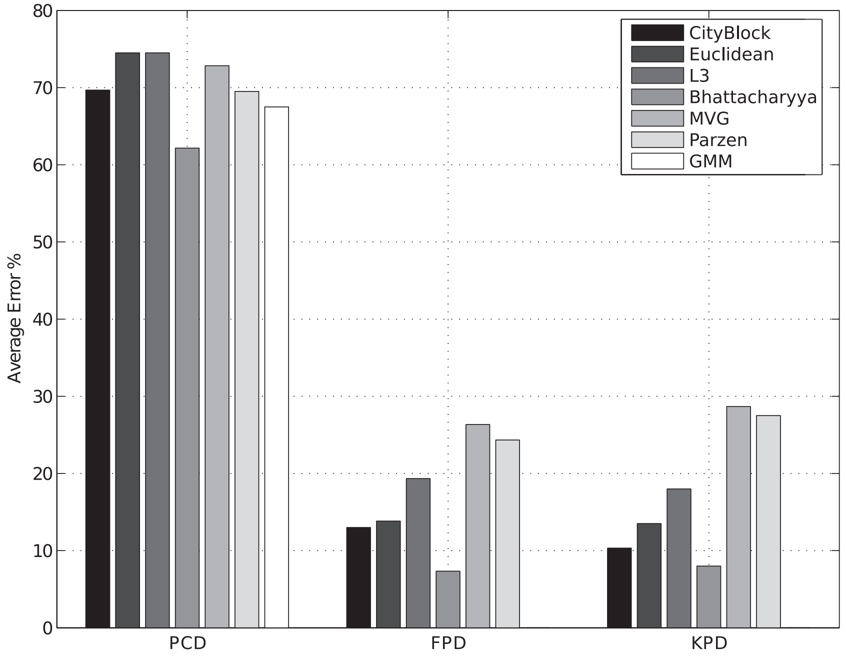


Table 2. Average Tonic Error Rates: Tonic-Estimation Methods and Granularities Against Frame Sizes

	All Candidates (AC)		*Best Candidates (BC)*	
	10 Cents	*5 Cents*	*10 Cents*	*5 Cents*
30-sec frames	18.35	14.74	24.33	22.78
60-sec frames	8.90	7.41	17.03	14.57
120-sec frames	8.00	6.78	12.33	9.49

Values calculated using KPD with nearest-neighbor classifier using Bhattacharyya distance and 15-cent precision.

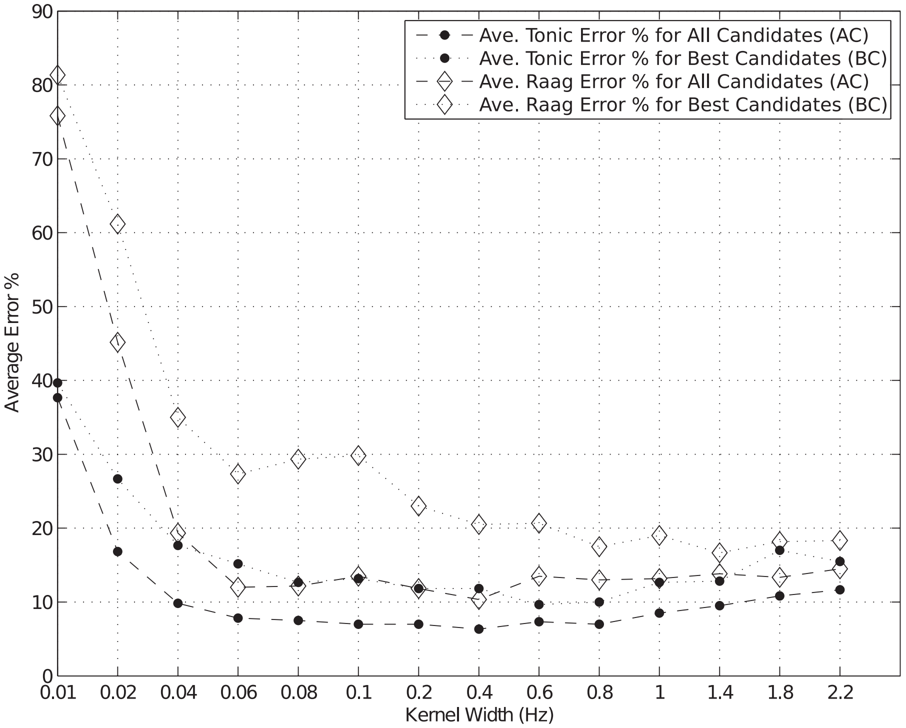
Figure 5 shows the distribution of tonic errors, that is, how often each of the other scale degrees was deemed to be the tonic. In particular, we were interested to know whether the tonic was confused with the fifth or fourth, tonally close areas, or whether they were random or otherwise distributed. In all cases, the most common errors were indeed the fourth and fifth. Compared to FPD and KPD, however, PCDs were more likely to confuse the tones neighboring the tonic: the minor second and major seventh.

Figure 6 shows that the average error rate for PCD (30.3 percent) was similar to FPD (30.7 percent) with the BC method, although both were significantly higher than KPD (18.6 percent). Compared to AC, the BC method dramatically reduced the error rate for PCD by at least 30 percentage points for precision levels between 15 and 25 cents. The results show that the KPD feature was much better than FPD for the BC method. Compared with to the AC method using NNB, the error rate for FPD increased from 7.3 percent to 23.3 percent, while only increasing from 8.0 percent to 12.3 percent for KPD using NNB (120-sec frames, 10-cent granularity, and 15-cent strictness).

Raag Recognition

As with tonic estimation, error rates for raag detection are reported for tenfold cross-validation. To establish a baseline for comparison, rates of raag recognition are given when the true tonic is known, which we refer to as "ground." As with tonic estimation, best performance was attained using KPD with 5-cent granularity, NNB, and the AC method (8.5 percent). This compares with a naive error rate of 88 percent, based on the prior probability

Figure 4. Average error rates versus different kernel widths, by Bhattacharyya distance.



of the most common raag. Across all feature types, NNB (Bhattacharyya) is again the best method (see Table 3). The extent to which NNB outperforms all other methods for raag recognition is striking. The next best classifier for PCD has an error rate 16.0 percentage points greater; the corresponding gaps are 12.0 for FPD and 13.2 for KPD. Unlike tonic estimation, the other NN methods do not consistently outperform the statistical classifiers.

Using the AC method with NNB, raag recognition error is 30.0 percent for PCD, 14.50 percent for FPD, and 12.5 percent for KPD (see Figure 7). In general, when using the AC method, FPD and KPD perform significantly better than PCD. When using the BC method, however, PCD (21.8 percent) outperforms FPD (33.5 percent), with KPD (19.2 percent) providing best results (120-sec frame, 10-cent granularity). Parallel to the results obtained in tonic recognition, AC provides better results compared to the BC method except when using PCDs (see Table 3).

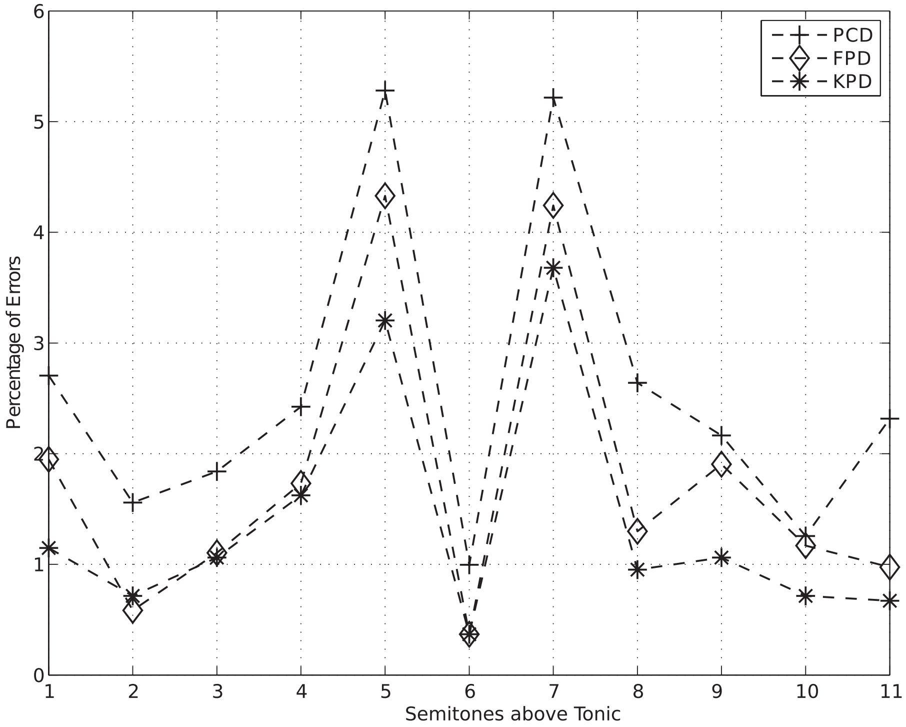
Table 4 shows the effect of frame size on raag error rates. There is a large drop in the error rate from 30 to 60 sec and a smaller decrease from 60 to 120 sec. For example, the KPD error rate using the AC method with 10-cent granularity goes from 26.5 percent to 15.9 percent to 12.5 percent. When the tonic is known, the drop from 60 to 120 sec is smaller: 11.1 percent to 8.5 percent.

Compared with 10 cents, using a granularity of 5 cents leads to performance increases for FPD (11.9 percent vs. 14.5 percent AC). For KPD, however, it makes almost no difference (11.9 percent vs. 12.5 percent AC), and for PCD it decreases performance (33.7 percent vs. 30.0 percent AC).

For KPD, kernel width again influenced performance, with best performance occurring with a width of 0.4 Hz (see Figure 4).

For the most part, confusion occurs between similar raags that share all or most scale degrees. The recall, precision, and F-measure for each raag are available at paragchordia.com/research/raag.html. For example, Khamaj and Gaud Malhar, which both use the major scale degrees with the addition of the minor seventh, are confused. Similarly, Desh is most often confused with Khamaj. Komal Rishabh Asaveri is confused with Darbari and differs from

Figure 5. Distribution of average of errors in the all-candidates and best-candidates methods with respect to tonic, by Bhattacharyya distance.



it with regard to only one scale degree (minor vs. major second).

Table 5 presents an informal performance comparison of the joint estimation, giving average run time across ten trials using the MATLAB "tic toc" timing functions (MathWorks 2009). The BC method yields 15- to 20-fold performance improvements for NN methods averaged across all feature types. This is because the construction of the distance matrix for the NN classifier has complexity of $O(NM)$, where N is the number of PDs we are testing (due to different tonic hypotheses) and M is the number of training examples. The number of tonic hypotheses considered by the BC method is typically less than 5 percent of the total number of possible tonic candidates (i.e., the number of bins in the PD).

Discussion and Conclusion

We have presented a novel method for tonic and raag recognition based on pitch distributions. These experiments provide evidence that it is possible to estimate the tonic accurately in a complex melodic musical genre that makes extensive use of continuous pitch movements and that uses a tremendous diversity of scale types. The more fine-grained pitch distributions, the FPD and KPD, proved to be much more appropriate features for tonic and raag recognition than the more widely used twelve-dimensional PCD. For tonic estimation, NN methods are clearly superior to statistical classifiers, at least for the amount and distribution of training data we have here. For raag recognition, the key point is the marked superiority of the nearest-neighbor classifier using Bhattacharyya distance. For these data, which are likely to contain multi-modal distributions, it is not surprising that instance-based classifiers were broadly superior—there were insufficient data to learn high-dimensional GMMs or non-parametric densities that could model this. But why, among the distance metrics used for NN, did Bhattacharyya distance perform so much better?

Figure 6. Average tonic error rates comparing features against classifiers for the best-candidates method.

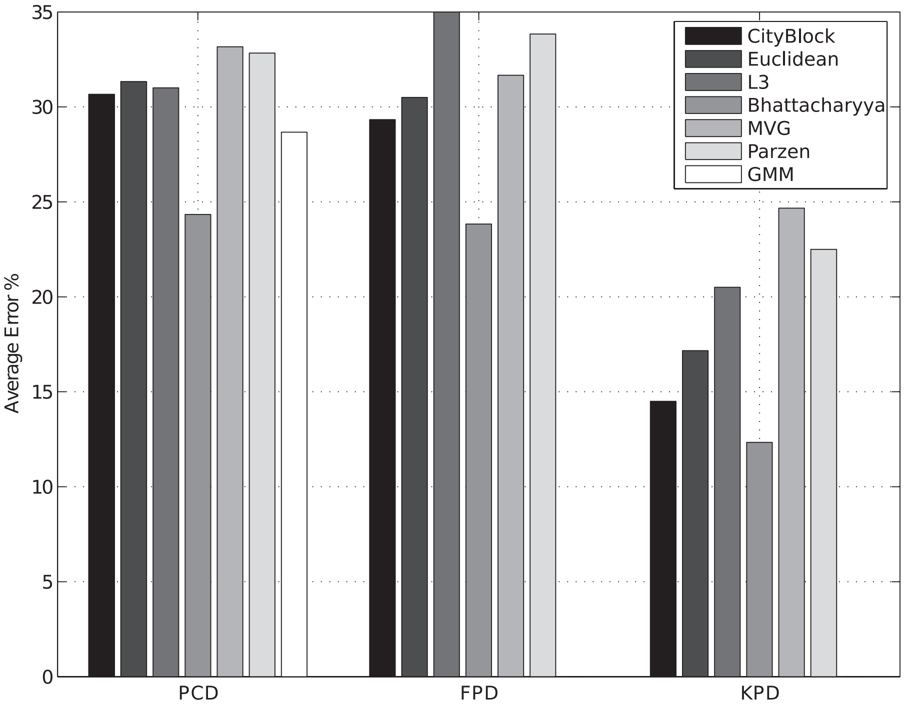


Table 3. Average Raag Error Rates: Features and Tonic-Estimation Methods Against Classifiers

	PCD			*FPD*			*KPD*		
	Ground	*All*	*Best*	*Ground*	*All*	*Best*	*Ground*	*All*	*Best*
City Block	29.17	46.00	39.17	21.00	26.50	44.33	22.67	25.67	36.00
Euclidean	29.33	49.17	41.67	30.00	36.50	62.33	32.33	37.00	50.00
L3	32.50	48.83	43.33	39.50	45.67	71.67	40.17	46.33	58.83
Bhattacharyya	12.50	30.00	21.83	8.67	14.50	33.50	8.50	12.50	19.17
MVG	35.17	57.67	50.83	26.00	37.00	51.83	35.17	44.67	44.17
Parzen	33.50	53.17	46.67	27.50	34.17	53.00	33.67	42.33	43.00
GMM	31.67	43.83	39.50						

"Ground" is the error rate when the tonic is known in advance; "All" and "Best" are the all-candidates and best-candidate methods, respectively.

In many image-recognition tasks that use NN, such as image retrieval, the Bhattacharyya distance (BD) has been shown to outperform Euclidean distance (ED) (Garcia, Zikos, and Tziritas 2000; Coleman and Andrews 1979). Unlike ED, BD is used to measure the similarity of probability distributions. To gain some insight into why BD performs so much better for raag recognition, it is worth considering several common scenarios. Consider the idealized case where we have two identical PCDs from the same raag. Now imagine that the relative strength of one scale degree is increased or decreased. Such a change will have a greater relative impact on ED than BD. In NICM, however, it would be extremely rare for two different raags to differ in only this regard; such PCDs are

Figure 7. Average raag error rates comparing features against classifiers for the all-candidates (AC) method.

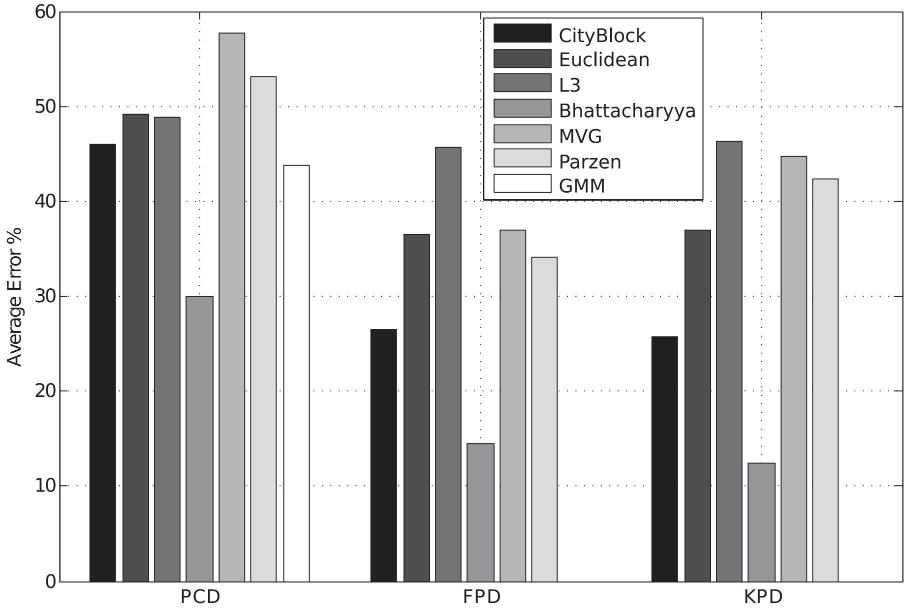


Table 4. Average Raag Error Rates: Estimation Methods and Granularities Against Frame Sizes

	Ground Truth		*All Candidates*		*Best Candidates*	
	10 Cents	*5 Cents*	*10 Cents*	*5 Cents*	*10 Cents*	*5 Cents*
30-sec frames	19.09	18.13	26.45	26.26	37.62	35.70
60-sec frames	11.10	12.33	15.85	15.95	27.80	22.93
120-sec frames	8.50	10.85	12.50	11.86	19.17	18.64

Error rates calculated using KPD with NNB and 15-cent precision.

Table 5. Informal Performance of Estimation Methods

	City Block	*Euclidean*	*L3*	*Bhattacharyya*	*MVG*	*GMM*	*Parzen*
All candidates	6.94	5g89	27.54	26.55	3.89	5.46	76.84
Best candidates	0.45	0.40	1.34	1.41	3.58	0.62	16.71

Performance measured in seconds by average run time across ten trials.

thus most likely from the same raag. Next consider the situation where there is some "pitch leakage", i.e., two PCDs are identical except that for a given scale degree some of the energy is distributed to the adjacent bins. This commonly happens when the tonic or pitch estimation is slightly off. Again, the change in distance will be greater for ED than BD. Finally, consider an idealized binary PCD where each scale degree has a strength of zero or one (scale degree is present or absent). If we have seven tones (i.e., seven of our twelve PCD bins have a value of 1/7), then switching one scale degree from major to minor (or vice versa) will result in a change of 0.2 for ED and 0.15 for BD. If we switch two scale degrees the change is 0.27 for ED and 0.34 for BD, and for three scale degrees the change is 0.33 for ED and 0.56 for BD. In other words, BD is about as sensitive as ED when one scale degree is changed but is more sensitive when several are changed, making it better at detecting changes that are likely to correspond to a different raag. Although the arguments here are highly simplified and do not take into account the complex interactions present in the real data, they do offer some insight into the marked superiority of BD over ED.

If computational performance is an important consideration, such as in real-time systems, computation can be greatly reduced by using the BC method. This leads to a minimal increase in tonic-estimation error, and a moderate increase in raag-recognition error (7%). Compared with other features, KPD provides far superior results when using the BC method. One explanation is that it tends to be smoother and is therefore less likely to have spurious peaks that are common in histograms with small bin widths (see Figure 2). To check this, we calculated the average number of candidates (i.e., peaks) in a frame for FPD (4.9) and KPD (5.5). Contrary to what we expected, the average number of candidates was higher for KPD. To estimate the upper-bound performance of BC, we calculated the percentage of time that the annotated tonic did not appear among the "best candidates." For FPD this was 6.1 percent, for KPD it was 2.0 percent. Although this explains some of the performance gap, it still does not fully explain the difference. In NICM it is common for the performer to spend time in the vicinity of the note due to glides and ornaments. In the FPD these pitches may be distributed to adjacent bins, forming a fork rather than a single peak, leading to peaks that are off-center of the true note. This is due to artifacts from the hard allocation of values to bins in the histogram method, which are visible in Figure 2 around 225 Hz. Overall, KPD using NNB is a robust method for raag and tonic detection.

It is worth noting how tonic-estimation errors affect raag-recognition performance. In most cases, raag recognition is not affected by small errors in the neighborhood of the true tonic. Other tonic errors, however, such as detecting the fifth as the tonic, lead to incorrect raag recognition in the vast majority of cases. This is expected because one will get a totally different set of scale degrees depending on the tonic. In general, the types of tonic errors made by the different approaches were quite similar.

Frame size has a clear effect on classification performance, with longer frames leading to better performance. This is unsurprising, as the PDs become more stable and more representative of the raag with more pitch data. Interestingly, there is a larger drop in error from 30 to 60 sec than from 60 to 120 sec. This is most likely due to the fact that during slow sections, notes are often held for several seconds with long pauses in between, leading to only a few distinct pitch classes in a 30-sec frame. PDs derived from such frames will contain little distinguishing information. In most cases, however, 60 sec is long enough that there are almost always several distinct pitches, leading to more discriminative PDs. Although 120-sec frames lead to even more stable PDs for a given raag, giving a better estimate of the relative strength of the notes, this additional information is usually only necessary when the raag has a close neighbor.

Future Work

A key result of this paper is that although raags are clearly temporally structured, good classification results can be obtained without using sequential information. An obvious next step would be to model the sequential structure of the melodies. Hidden Markov models are logical candidates. Our initial experiments, however, suggest that a straightforward application—treating pitch values as observations and letting hidden states correspond to stable scale tones—will not work well. The primary difficulty centers around segmentation of the pitch track. The continuous nature of the playing style and of the resulting pitch track make onset-based segmentation of limited use. Despite frequent slurs and portamenti, however, the characteristic melodic patterns of a raag are primarily based on discrete note patterns. Without segmentation, these patterns are obscured; after training, the transition matrix tends to become dominated by self-transitions (i.e., almost diagonal), making it difficult for note transitions to exert much influence on the decoded state sequence. For this reason, we are working on segmentation algorithms that are able to parse the continuous surface into a discrete representation that corresponds to "notes" that a skilled Indian classical musician would hear. Although this is still an active area of research, even a simpler segmentation into stable and transient pitch regions might allow for the successful application of *n*-gram modeling. We are particularly interested in applying variable-length Markov models and multiple-viewpoint models to the problem of raag recognition (Begleiter, El-Yaniv, and Yona 2004; Chordia, Sastry, and Albin 2010; Chordia et al. 2010; Şentürk 2011; Srinivasamurthy and Chordia 2012). Our previous research suggests that *n*-gram modeling could lead to increase in accuracy (Chordia and Rae 2007).

Acknowledgments

The authors would like to thank all the musicians who contributed to the GTraagDB. This material is based upon work supported by the National Science Foundation under grant 0855758.

References

Akkoç, C. 2002. "Non-Deterministic Scales Used in Traditional Turkish Music." *Journal of New Music Research* 31(4):285–293.

Aldwell, E., and C. Schachter. 2002. *Harmony and Voice Leading*. New York: Schirmer Books, 3rd edition.

Bartsch, M. A., and G. H. Wakefield. 2001. "To Catch a Chorus: Using Chroma-Based Representations for Audio Thumbnailing." In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 540–545.

Begleiter, R., R. El-Yaniv, and G. Yona. 2004. "On Prediction Using Variable Order Markov Models." *Journal of Artificial Intelligence Research* 22:385–421.

Bello, J. P., and J. Pickens. 2005. "A Robust Mid-Level Representation for Harmonic Content in Music Signals." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 304–311.

Bozkurt, B. 2008. "An Automatic Pitch Analysis Method for Turkish Maqam Music." *Journal of New Music Research* 37(1):1–13.

Camacho, A. 2007. "SWIPE: A Sawtooth Waveform Inspired Pitch Estimator for Speech and Music." PhD dissertation, University of Florida, Graduate School.

Cao, X. 2009. "Automatic Accompaniment of Vocal Melodies in the Context of Popular Music." Master's thesis, Department of Music, Georgia Institute of Technology.

Chordia, P. 2004. "Automatic Rag Classification Using Spectrally Derived Tone Profiles." In *Proceedings of the International Computer Music Conference*, pp. 83–87.

Chordia, P. 2006. "Automatic Raag Classification of Pitchtracked Performances Using Pitch-Class and Pitch-Class Dyad Distributions." In *Proceedings of the International Computer Music Conference*, pp. 314–321.

Chordia, P., and A. Rae. 2007. "Raag Recognition Using Pitch-Class and Pitch-Class Dyad Distributions." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 431–436.

Chordia, P., and A. Rae. 2008. "Raag Vidya: Real-Time Raag Recognition for Interactive Music." In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 331–334.

Chordia, P., A. Sastry, and A. Albin. 2010. "Evaluating Multiple Viewpoint Models of Tabla Sequences." In *Proceedings of the International Workshop on Machine Learning and Music*, pp. 21–24.

Chordia, P., et al. 2010. "Multiple Viewpoints Modeling of Tabla Sequences." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 381–386.

Coleman, G., and H. Andrews. 1979. "Image segmentation by clustering." *Proceedings of the IEEE* 67(5):773–785.

de Cheveigné, A., and H. Kawahara. 2002. "YIN, a Fundamental Frequency Estimator for Speech and Music." *The Journal of the Acoustical Society of America* 111(4):1917–1930.

Duda, R. O., P. E. Hart, and D. G. Stork. 2001. *Pattern Classification*. New York: Wiley, 2nd edition.

Fujishima, T. 1999. "Realtime Chord Recognition of Musical Sound: A System Using Common Lisp Music." In *Proceedings of the International Computer Music Conference*, pp. 464–467.

Garcia, C., G. Zikos, and G. Tziritas. 2000. "Wavelet Packet Analysis for Face Recognition." *Image and Vision Computing* 18(4):289–297.

Gedik, A. C., and B. Bozkurt. 2009. "Evaluation of the Makam Scale Theory of Arel for Music Information Retrieval on Traditional Turkish Art Music." *Journal of New Music Research* 38(2):103–116.

Gedik, A. C., and B. Bozkurt. 2010. "Pitch-Frequency Histogram-Based Music Information Retrieval for Turkish Music." *Signal Processing* 90(4):1049–1063.

Gomez, E. 2006. "Tonal Description of Polyphonic Audio for Music Content Processing." *INFORMS Journal on Computing* 18(3):294–304.

Huron, D., and R. Parncutt. 1993. "An Improved Model of Tonality Perception Incorporating Pitch Salience and Echoic Memory." *Psychomusicology* 12:154–171.

İzmirli, O. 2005. "Tonal Similarity from Audio Using a Template Based Attractor Model." In *Proceedings of the International Symposium on Music Information Retrieval*, pp. 15–19.

Jairazbhoy, N. A. 1971. *The Rāgs of North Indian Music: Their Structure and Evolution*. London: Faber and Faber.

Juslin, P. N., and J. Sloboda. 2001. *Music and Emotion: Theory and Research*. New York: Oxford University Press.

Krumhansl, C. L., and E. J. Kessler. 1982. "Tracing the Dynamic Changes in Perceived Tonal Organization in a Spatial Representation of Musical Keys." *Psychological Review* 89(4):334–368.

Krumhansl, C. L., and R. N. Shepard. 1979. "Quantification of the Hierarchy of Tonal Functions within a Diatonic Context." *Journal of Experimental Psychology: Human Perception and Performance* 5(4):579–594.

Lee, K. 2006. "Automatic Chord Recognition Using Enhanced Pitch Class Profile." In *Proceedings of International Computer Music Conference*, pp. 304–311.

MathWorks. 2009. "MATLAB." Natick, Massachusetts: MathWorks. Available online at www.mathworks.es/products/matlab/. Accessed April 2013.

Moelants, D., O. Cornelis, and M. Leman. 2009. "Exploring African Tone Scales." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 489–494.

Pauws, S. 2004. "Musical Key Extraction from Audio." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 96–99.

Schroeder, M. R. 1968. "Period Histogram and Product Spectrum: New Methods for Fundamental-Frequency Measurement." *The Journal of the Acoustical Society of America* 43(4):829–834.

Şentürk, S. 2011. "Computational Modeling of Improvisation in Turkish Folk Music Using Variable-Length Markov Models." Master's thesis, Georgia Institute of Technology.

Srinivasamurthy, A., and P. Chordia. 2012. "Multiple Viewpoint Modeling of North Indian Classical Vocal Compositions." In *Proceedings of the International Symposium on Computer Music Modeling and Retrieval*. Available online at mtg.upf.edu/system/files/publications/CMMR2012.pdf. Accessed April 2013.

Temperley, D. 2001. *The Cognition of Basic Musical Structures*. Cambridge, Massachusetts: MIT Press.

Temperley, D., and E. W. Marvin. 2008. "Pitch-Class Distribution and the Identification of Key." *Music Perception* 25(3):193–212.