Challenging Uncertainty in Query by Humming Systems: A Fingerprinting Approach

Erdem Unal, *Student Member, IEEE*, Elaine Chew, *Member, IEEE*, Panayiotis G. Georgiou, *Member, IEEE*, and Shrikanth S. Narayanan, *Senior Member, IEEE*

***Abstract*—Robust data retrieval in the presence of uncertainty is a challenging problem in multimedia information retrieval. In query-by-humming (QBH) systems, uncertainty can arise in query formulation due to user-dependent variability, such as incorrectly hummed notes, and in query transcription due to machine-based errors, such as insertions and deletions. We propose a fingerprinting (FP) algorithm for representing salient melodic information so as to better compare potentially noisy voice queries with target melodies in a database. The FP technique is employed in the QBH system back end; a hidden Markov model (HMM) front end segments and transcribes the hummed audio input into a symbolic representation. The performance of the FP search algorithm is compared to the conventional edit distance (ED) technique. Our retrieval database is built on 1500 MIDI files and evaluated using 400 hummed samples from 80 people with different musical backgrounds. A melody retrieval accuracy of 88% is demonstrated for humming samples from musically trained subjects, and 70% for samples from untrained subjects, for the FP algorithm. In contrast, the widely used ED method achieves 86% and 62% accuracy rates, respectively, for the same samples, thus suggesting that the proposed FP technique is more robust under uncertainty, particularly for queries by musically untrained users.**

***Index Terms*—Database searching, information retrieval, music, uncertainty.**

I. Introduction

CONTENT-BASED multimedia information retrieval (MIR) is gaining widespread attention and becoming increasingly important. The growing capacity of web servers parallels the explosion of information generated worldwide. The need for efficient and natural access to these databases cannot be overstated. Digital music and its associated information are prime examples of such complex information that can be stored in a variety of formats, such as MP3, MIDI, wav, scores, etc. These data can also be accessed in multiple ways. If the user is familiar with the name of the song or the band, and the source material is annotated with metadata, retrieval can be straightforward. However, if one does not know the lyrics, title, or the performer, alternative retrieval methods are necessary, such as through singing, humming, or playing a sample of the piece, as a query to the database. Enabling such kinds of natural human interactions with large databases has thus become an essential component of effective and flexible MIR systems.

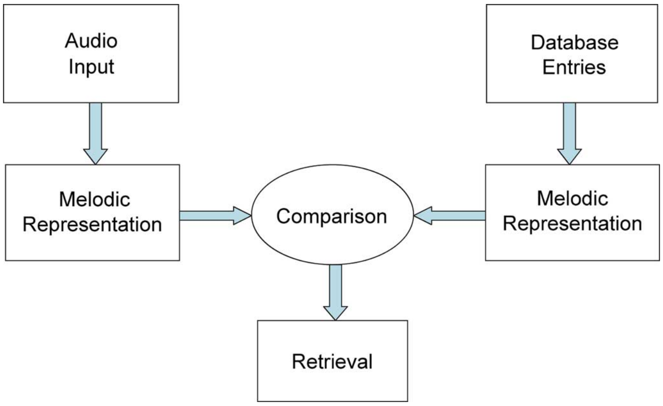


Fig. 1. Typical QBH system compares existing musical representations with new audio input to return the appropriate match.

This paper focuses on a statistical approach to the uncertainty problem in the retrieval process of a query-by-humming (QBH) System. Fig. 1 shows the components of a typical QBH system. The humming audio and the database elements should be represented in such a way as to enable meaningful comparisons during search and retrieval. The input query is transcribed to a melodic representation in the front-end of the system. The database entries are converted to this same melodic representation. The query is compared to the database entries in the retrieval part, and the system returns the most appropriate match. Search and retrieval form the back-end of the system. One reason uncertainty arises is because human queries are subject to user-dependent variability, and the hummed input may not accurately match the original melody (the expected form) in the database. Therefore, identifying features that are more robust in the presence of user- and algorithmic-specific variability and errors are proposed. These small packets of underlying characteristic information are called fingerprints (FPs) abstracted from the musical piece. The melodic representations of the query and the database are compared through the corresponding FPs of the pieces.

A hidden Markov model (HMM)-based note segmentation system, much like a statistical pattern recognition-based speech recognizer, is used for mapping audio to meaningful melodic symbols in the front-end. The HMM-based statistical recognizer is combined with postprocessing using pitch and energy

Manuscript received December 15, 2006; revised October 28, 2007. This work was supported in part by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, under Cooperative Agreement EEC-9529152, in part by the National Science Foundation Information Technology under Research Grant NSF ITR 53-4533-2720, and in part by the ALi Microelectronics Corporation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors, and do not necessarily reflect those of the National Science Foundation or ALi Microelectronics Corporation. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Masataka Goto.

The authors are with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: unal@usc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2007.912373

information for improved segmentation. Our HMM model construction differs from earlier statistical algorithms for note segmentation. Our contribution to the note segmentation part of the QBH problem is the grouping of humming syllables with respect to linguistic characteristics of the humming sounds produced and, informed by analysis of humming patterns, the training of the HMM segmentation tool using speech data. This grouping strategy enhances the segmentation performance in the front end, compared to the building of a single universal model. Details are provided in Section IV. Postprocessing refines this segmentation by using standard pitch and energy tracking methods to locate and correct insertion and deletion errors. The segmented information is employed in the extracting of pitch and duration information for use in audio-to-symbol transcription. System-dependent transcription errors may occur, such as note insertions and deletions, in audio-to-symbol conversion at the system's front-end. The final representation is in the form of relative pitch differences, in semitones, and of duration ratios for consecutive note transitions.

For search and retrieval, a fingerprinting (FP) algorithm is proposed that extracts salient fixed length sections of the symbolic contour representation. FPs cover rare melodic segments in the query, where minimum and maximum pitch and duration changes occur. Unlike monotonic subsequences, where the pitch or duration contour tends to be stable, FPs are hypothesized to carry distinct information about the query. The extracted features, namely the FPs, are then compared statistically to the database entries, in order to find the closest match. The use of FPs was first proposed in Unal *et al.* [1]. This paper expands on our earlier effort by evaluating the algorithm on a larger database, and improving the algorithm by incorporating statistical distance measures. Our contribution to the retrieval problem, explained in Section V, includes not only the consideration of important parts of the melody for comparison, but also the statistical alignment of the selected melodic features to the database transcriptions to decrease computation time.

FPs represent melodic information that more strongly characterizes the hummed query. When humming a melody, a person would likely strive to capture its essence by recognizably reproducing these characteristic segments. Another advantage of the FP approach is that global music syntax, and the exact sequence of the melody, become less important. This is important because users will often hum only a specific part of the melody that they remember best. Search algorithms based on sequence alignment may be less able to handle these types of uncertainty. Since FPs are designed to locate distinctive melodic segments in the query, calculating similarity measures with respect to these characteristic features is proposed to be more efficient in finding the desired match. These hypotheses are tested in this paper by comparing the performance of FP-based retrieval with that of an edit distance (ED) method.

A. Related Work

Various researchers have proposed different representations for hummed audio input; based on their chosen representation, different search and retrieval techniques have been put forward in the QBH literature.

Initial efforts on QBH systems used note-level decoding of humming input for pitch and duration contour representation [2]–[4]. Autocorrelation-based pitch tracking algorithms and energy tracking algorithms were used to determine note boundaries. Other than note-level decoding, fixed-size audio frame representation was also used, which may avoid note transcription errors, but can be computationally expensive during the matching process [5]. For retrieval, dynamic programming (DP)-based ideas were evoked to find the closest match in a database. dynamic time warping (DTW), a specific instance of DP for time series data, is one of the most popular techniques in the MIR literature that has been used extensively in different forms [5]–[10].

Pardo *et al.* [11] and Jang *et al.* [8] used sequential statistical modeling techniques, such as HMMs and CHMMs, for representing audio elements in the database. In such approaches, each of the melodies in the database generates a sequential statistical model, and a database melody is judged similar to the query if its HMM representation has a high likelihood of producing the symbolic query. Statistical modeling presents powerful means for representing variability, but appropriate training of such systems tend to be computationally expensive. Rather than modeling individual songs, as these researchers did, HMMs are used in the front-end segmentation part of our system, for statistical segmentation of the humming note instances in the audio, so as to enable accurate audio-to-symbol transcription.

Dannenberg *et al.* [12] created a test-bed for comparing the retrieval accuracy and computation time of different retrieval systems. They compared the results of search algorithms that use: note-interval matching with DP, fixed-frame melodic contour matching with DTW, and an HMM. They reported that contour matching with DTW and the HMM approach, while extremely slow, outperforms all conventional and faster methods. On the other hand, the note interval system was able to deliver similar results and was faster than the other two methods. Ito *et al.* [13] compared different selected feature sets for DP-based search algorithms. They introduced three different similarity measures: distance, quantization, and fuzzy quantization-based similarity measures. They concluded that the distance-based similarity, based on absolute difference between two compared pitches, performed best for retrieval experiments. Parker [14] presented a retrieval system using binary classification and adaptive boosting. After several iterations with boosting, the proposed system was able to outperform DTW algorithms with negligible increase in time.

Similar types of sequence alignment and comparisons can be seen in genetic applications [15] employing statistical string matching techniques such as n-grams. The typical query input in QBH is rarely an exact substring of the original melody in the database because of the imperfect production of humming. Thus, a distance measure needs to be incorporated alongside the sequential alignment techniques. More recently, polyphonic music retrieval ideas are being incorporated into QBH research [16]–[21]; the core of the polyphonic retrieval problem is very similar to that of QBH, even though representations may differ. We intend to focus on polyphonic retrieval in future work.

With regards to QBH front ends, Pauws [22], in his Cuby–Hum system, designed a new transcription technique that processes the input in terms of energy and pitch to detect note onsets

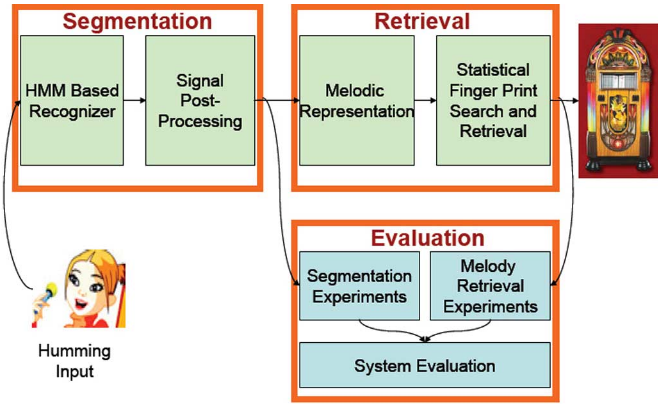


Fig. 2. Proposed QBH system.

and locations, and quantizes the input frequencies to semitone representations. Pauws also used an ED approach for retrieval. Clarisse *et al.* [23] first evaluated several existing transcription systems (such as Meldex, Pollastri, Autoscore, etc.) and, observing that they are not adequate for human level performance, constructed an auditory model-based transcription system for their QBH system front end. Inspired by the lack of robust humming audio transcription systems, Shih *et al.* [24], [25] used HMMs in the front end of their QBH system to statistically capture the time location of notes in the input. Energy and Mel frequency cepstral coefficient (MFCC) features were used to train the statistical models; for a fixed syllable input case (humming with/DA/), 95% note segmentation accuracy was reported.

Few other researchers have focused on accurate audio-to-symbol transcription in designing QBH systems; more have focused on retrieval techniques. In this paper, both the audio-to-symbol mapping problem and the retrieval problem are addressed and discussed in detail, and experimental results are reported. Our contribution to the audio-to-symbol transcription is the use of a task-specific segmentation algorithm with HMMs for accurate note boundary detection, as in state-of-the-art automatic speech recognition techniques for encoding speech audio into phonemes/syllables, and, on the retrieval side, the use of statistical distance measures to align salient parts (FPs) of the input query to the database entries for fast and robust search under uncertainty in the audio input.

II. Proposed QBH Approach

In order to design a robust QBH system, knowledge derived from signal processing and music theory should be combined and used advantageously. The proposed system has been divided into the Segmentation, Retrieval and Evaluation subsystems, with respect to their functionality as shown in Fig. 2. The hummed audio input is processed in the segmentation part of the system using signal processing techniques. The retrieval part of the system first converts the segmented audio into meaningful music symbols of pitch and duration contours, exploiting music knowledge, and performs a robust search guided by statistical findings derived from analysis of our previously collected human-generated humming database. The evaluation part tests the performance of both the segmentation and the retrieval modules.

III. Data

Two sets of databases were developed for this study: a human-generated humming database (which we shall call the Humming Database), and a MIDI database of melodies (termed the Experimental Melody Dataset).

Our interest was to achieve retrieval through humming instantiations of melodies, and as such, the database we collected supports this goal. While our present work is restricted to hummed queries, we hope that in the future we can generalize our techniques to the retrieval of sung instantiations, and employ alternative databases, such as that of the Query-by-Singing/Humming(QBS/H) 2006 Music Information Retrieval eXchange(MIREX) competition [26].

A. Humming Database

One of the important components of the proposed QBH system is the Humming Database. This data is essential for the design of the recognition and retrieval subsystems. It contains information about the levels of variability one can expect in the humming audio input. Statistical information gathered from the data is used to estimate important model parameters. A large database is necessary so that the system can learn statistical rules that robustly guide the search and retrieval function. To satisfy these needs, extensive work was carried out in the collection of a humming database.

A list of 22 well-known melodies was chosen for their melodic structure. The 22 target melodies covered a mixed variety of genres and musical structures. Some of the melodies that are included in the database are: “I am a Little Teapot,” “London Bridge is Falling Down,” “If you are Happy,” “American Anthem,” “Hey Jude,” “Take Me Out to the Ball Game,” and “Ten Little Indians.” Since our melodic representation focuses on relative pitch and duration information with respect to consecutive notes, the list was selected so that the melodies would cover a wide range of intervals up to a full octave (12 semitones), ascending and/or descending, and represent a variety of metric structures (2/4, 3/4, and 4/4). The numbers of each interval covered in the sample melody list is shown in Fig. 3. Only one interval (major 7th, M7) was not represented, and all time signatures listed were accounted for.

One hundred persons with widely varying musical backgrounds participated in our data collection experiment. The data collected were categorized into two main groups with respect to the hummer’s musical background: musically trained, and nontrained. The musically trained persons’ training ranged from two years of amateur-level instrument training to 25+ years of professional training. Nontrained persons, on the other hand, had no musical training at all.

Each person selected ten melodies from the melody list based on their familiarity with the melodies, and hummed each of the melodies twice. Each participant was also asked to hum three other melodies of their own choice, in order to increase the variability of the database. A detailed description of the database can be found in [27].

In this paper, we created an evaluation corpus of 400 humming samples, 100 samples each of four target songs, “Happy Birthday,” “London Bridge is Falling Down,” “Take Me Out to

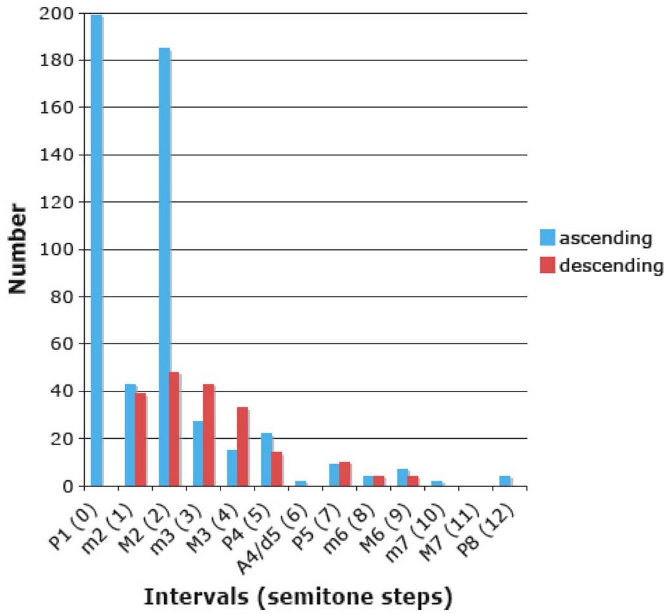
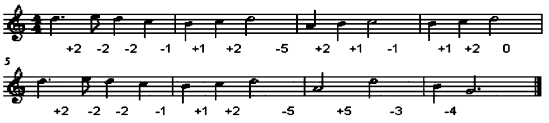


Fig. 3. Interval coverage in melody list. (Symbols describing intervals: A—augmented, M—major, m—minor, d—diminished.)



Song ID:007
Song Name:London Bridge is Falling Down
RPD: 0 2 -2 -2 -1 1 2 -5 2 1 -1 1 2 0 2 -2 -2 -1 1 2 -5 5 -3 -4
RDD: 1 0.33 2 1 1 1 2 0.5 1 2 0.5 1 2 0.75 0.33 2 1 1 1 2 1 1 0.5 1.5

Fig. 4. Score representation of "London Bridge is Falling Down" with sample semitone transitions and database representation of the same melody, with relative pitch difference (RPD) in semitones and relative duration difference (RDD) in time ratios.

the Ball Game" and "Hey Jude." The samples are equally distributed between musically trained and nontrained subjects.

B. *Experimental Melody Dataset*

Currently, our melody database consists of 1500 preprocessed MIDI files, stored in the proposed notation format, representing relative pitch and duration transitions, in terms of semitones and duration ratios, respectively. The melody database includes, in addition to the 22 preselected melodies, 200 songs by the Beatles, and classical melodies by composers such as Bach, Beethoven, Haydn, Mozart, etc., downloaded from Musedata [28] as single channel MIDI scores. For each MIDI file in the database, the pitch and duration transition information is stored in the form of symbol sequences similar to that shown in Fig. 4.

In order to perform one-to-one mappings between the input and original melodies, the database elements should be in the same format as the query, and thus have to be preprocessed. Preprocessing includes converting MIDI channels into streams of pitch and duration transitions, with manual extraction of the melodies when necessary. The song "London Bridge is Falling Down" and its database representation are shown in Fig. 4.

TABLE I
LEXICON: BOOTSTRAP MODELS WITH THEIR PHONEME STRUCTURES

Model	Consonant	Phoneme
DA (voiced stops)	D D D ...	IX AE IH ...
TA (unvoiced stops)	T T ...	IX AE ...
RA (Liquids)	R R ...	IX AE ...
LA	L L ...	IX AE ...
NA (Nasals)	N N N ...	IX AE IH ...

IV. NOTE SEGMENTATION AND POSTPROCESSING

In order to accurately capture the way humans perceive and reproduce music in a QBH system, the front end needs to be able to robustly extract pitch and duration representations of the hummed notes. This requires accurate segmentation of the hummed notes in the query audio. In this paper, an HMM-based speech recognizer, in conjunction with targeted postprocessing of its output, is used for note segmentation. The recognizer output is ultimately converted to meaningful musical symbols using the same representation as that of the original melody database entries.

A. *Automatic Segmentation Setup*

One way of segmenting hummed notes is to exploit our knowledge of how users are likely to hum. Users often use syllables, such as/LA/to hum a melody. To take advantage of this knowledge, we built an HMM-based speech recognition system for note segmentation using MFCCs [29]. The system is built on the ARPA Wall Street Journal task [30] of continuous English speech (containing a 50-phoneme set). Based on this speech recognition system, we defined four generic syllable model types, denoted by/DA/,/TA/,/NA/,/RA/in the lexicon, where each model represents a single type of consonant that is expected at the beginning of a hummed note [/DA/: voiced stops (b, c, d, g),/TA/: unvoiced stops (p, t, k),/NA/: nasals (m, n), and/RA/: liquids (l, r)].The lexicon also includes the different types of vowels (AA, AH, IH, AE) that are expected to follow the consonants to form the hummed syllable. The four generic syllable models provided in the dictionary aim to account for all different types of musical syllables that can be used by the human subjects, regardless of the consonant at the beginning of the humming syllable, or the vowel following. The lexicon supplied to the recognizer is summarized in Table I.

Fig. 5 shows a typical humming input, with the recognizer's output labeled on the waveform as note boundaries. As seen in the output, each note is labeled with one of the note models supplied in the lexicon.

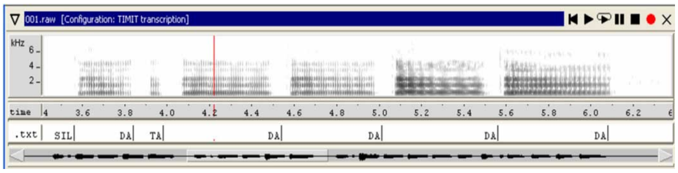


Fig. 5. Visualization of the recognizer output: detected notes are labeled under the spectrogram as humming syllable models—{/DA/,/TA/,/NA/,/RA/, or/SIL/}.

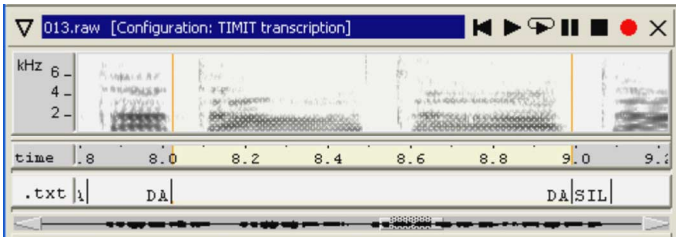


Fig. 6. Deletion error in the recognizer's output: missing boundary between two notes in the center (between the vertical lines).

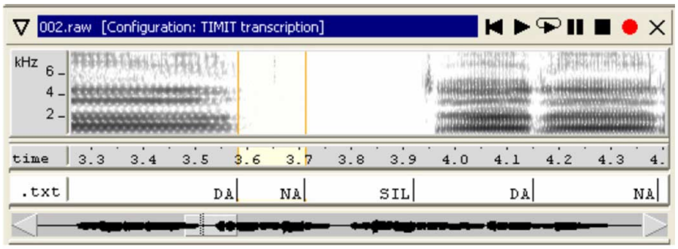


Fig. 7. Insertion error in the recognizer's output: extra note /NA/ is detected after the first /DA/, should be /SIL/.

B. Segmentation Errors and Postprocessing

The output of the speech recognition based front-end segmentation can contain errors, such as insertions and deletions of notes. Since duration and pitch information is derived from the segmented notes, the accuracy of the extracted note onsets and offsets is important. Fig. 6 shows an example of a note deletion error, where two notes between the vertical lines in the spectrogram are detected as a single note. Fig. 7 shows an insertion error at the end of the first detected /DA/.

Note segmentation accuracy directly affects the performance of the retrieval system. If the recognizer fails to segment two or more hummed notes one from another, the feature extraction mechanism will consider these nonsegmented units as one single note, leading to erroneous extraction of note pitch and duration. These insertions and deletions will, in turn, cause retrieval mismatches.

In order to detect common segmentation errors, and to correct them whenever possible, the output of the recognizer is postprocessed using heuristics based on energy and pitch information. As mentioned earlier, the recognizer uses MFCCs as its feature set. Incorporating additional energy and pitch information as a new feature set at the postprocessing stage can be useful for correcting such segmentation errors introduced by the recognizer, whose MFCC-based models have been trained with speech data. Since standard energy extraction algorithms are more accurate than pitch extraction algorithms, energy analysis is applied first.

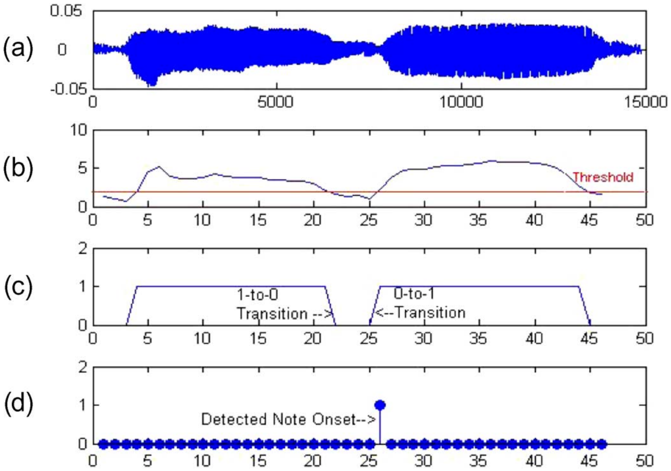


Fig. 8. Example of deletion correction using energy-based threshold filtering. (a) Two humming syllables which the recognizer considered to be one. (b) Corresponding energy sequence and local energy threshold. (c) Quantization with respect to the threshold value. (d) Correct detected note onset.

Pitch analysis is used as a final step for correcting segmentation errors. This order of postprocessing proves to perform best when tested against other sequential orders.

1) Short-Term Energy Analysis: We design our energy analysis postprocessing step specifically for correcting deletion errors. We compare the extracted energy feature with a threshold value, derived from a small held-out set, to introduce new note boundaries undetected by the recognizer. For each segmented humming note in the recognizer's output, the signal is windowed into frames of 20 ms, with a shift of 10 ms, which creates a 50% overlap between consecutive analysis frames. For each frame k the short-term energy is calculated as

$$E_k = \sum_{m=1}^{N} y(m)^2 \tag{1}$$

where N is the number of samples in an analysis frame (sampling rate × 0.02). For the energy sequence E, an adaptive threshold value τ_e is defined as the product of the median value of E' (nonzero elements of the energy sequence, E), and a constant, α, that is calculated from a development set. Values in E greater than the threshold are quantized to 1, and values smaller than the threshold are quantized to 0. A note onset is detected if a 1-to-0 transition is followed by a 0-to-1 transition. The onset is positioned at the 0-to-1 transition point. The offset of the first note is given by the time of the 1-to-0 transition. The procedure can be seen in Fig. 8.

2) Pitch Analysis: The pitch vector, as yet unused by both the recognizer and the energy analysis, can also be helpful in finding deletion errors. Since each segmented audio chunk corresponds to a humming note, where the pitch vector tends to stay the same, the analysis of pitch vector extracted for consecutive frames can be used to detect immediate changes within the corresponding note.

Pitch tracking for the hummed signal is performed using a standard pitch detection algorithm with the PRAAT software, which employs an autocorrelation method. The extracted pitch was stored in a sequence P. For each segment, the gradient of P

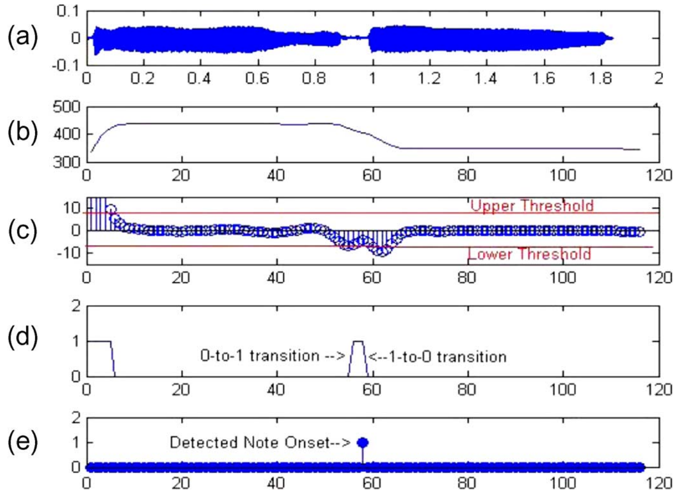


Fig. 9. Example of deletion correction using pitch threshold filtering. (a) Two humming syllables which the recognizer considered to be one. (b) Corresponding pitch sequence P. (c) Gradient vector, and threshold region τ_p. (d) Pitch change quantization. (e) Onset detection.

was calculated, and its absolute value compared to a threshold value τ_p, which was estimated using a development set. Values that fall between the thresholds, τ_p and $-\tau_p$, are quantized to 0, and the remaining values are set to 1. An onset is recognized when a 0-to-1 transition is followed by a 1-to-0 transition. The onset is positioned at the time of the 1-to-0 transition. The procedure is illustrated in Fig. 9.

3) Segmentation Evaluation: Reference transcriptions for the hummed samples are created manually by an experienced music student. Each manual transcription is compared to the automatic transcription of the proposed front-end recognizer to evaluate the recognizer's performance. Standard precision (PRC) and recall (RCL) measures are used to evaluate the segmentation performance of the system. These measures are defined as follows:

$$\text{PRC} = \frac{\text{Number of Correctly found boundaries}}{\text{Number of Hypothesized Boundaries}} \quad (1)$$

$$\text{RCL} = \frac{\text{Number of Correctly found boundaries}}{\text{Total Number of Boundaries}}. \quad (2)$$

A hypothesized boundary at time t is correct if it lies within the time interval $t_0 - \Delta T \leq t \leq t_0 + \Delta T$, where t_0 is the correct boundary, and ΔT is the threshold value. In this study, ΔT is empirically chosen to be 75 ms, which corresponds to approximately 10% of the average note length in our test dataset or three to four frames of audio. A 10% duration difference is assumed to be negligible in our experiments.

The F measure [31] is also calculated to provide a single performance metric using PRC and RCL:

$$F = \frac{2 \times \text{PRC} \times \text{RCL}}{\text{PRC} + \text{RCL}}. \quad (4)$$

Segmentation tests are repeated ten times over 200 humming samples of "Happy Birthday." At each iterations, 10% of the data was randomly selected, and the values of τ_p and τ_e (defined in Sections IV-B1 and IV-B2) that maximize the F measure are calculated. These values of τ_p and τ_e are then used in the tests on the remaining 90% of the data, and an F measure for that particular dataset calculated. After the ten rounds are completed, results are averaged. For a 75-ms tolerance window, the performance of the recognizer and the improvement with the pitch and energy postprocessing are shown in Table II.

TABLE II
F MEASURE RESULTS FOR SEGMENTATION PERFORMANCE

	F	1-F (Error)	Error Improvement	
Pitch & Energy	0.67	0.33	-	-
Speech Recognizer	0.79	0.21	36%	-
Speech Rec. + Pitch & Energy Post-Processing	0.84	0.16	51%	23%

From Table II, one can see that, on average, 21% of the recognizer's output sequences contain insertion and deletion errors. Postprocessing decreases the recognition error rate by a relative 23%. The recognizer with postprocessing achieves an accuracy (F measure) of 84%. The performance improvement is confirmed to be statistically significant, with $\text{p} \leq 0.01$, using McNemar's test for proportion comparison.

V. Search and Retrieval

The first task of the retrieval back-end part of the system is to transcribe the segmented humming audio to musical symbols with pitch and duration information. The ultimate goal is to compare the transcribed audio to the database entries and to find the closest melodic matches. Because of the aforementioned challenges such as user-dependent uncertainty and system-dependent segmentation errors, the query sequence may contain high levels of uncertainty. Such uncertainty affects the query calculations and can result in matching errors. For this reason, robust retrieval must handle and accommodate such inherent uncertainty in the queries.

A. *Transcription*

Conventional music notation provides a comprehensive way to represent a melody. The goal in our studies is to utilize an approach that is both comprehensive, and interconnects the query formulation in the front end with the database entries in the back end. As indicated in Fig. 1, both the audio input and the database entries have to be represented in a consistent way so that meaningful symbol-to-symbol comparisons can be applied for search and retrieval. Inspired by the QBH studies of other researchers [2]–[4], [6]–[8], relative pitch and duration information have been selected as the main attributes of a musical note used in our retrieval calculations.

After extracting duration and pitch values for each hummed note, we label the note transitions. The relative pitch difference (RPD), in semitones, is calculated using the formula

$$\text{RPD}(k) = \frac{log f_k - log f_{k-1}}{log \sqrt[12]{2}} \quad (5)$$

TABLE III
SECTION OF TRANSCRIBED OUTPUT FOR A HUMMING SAMPLE OF "HAPPY BIRTHDAY"

RPD	0	1.70	2.44	-1.71	5.19	-0.81	-4.79	0.07	2.44	...
RDD	1	0.78	1.78	1.14	0.94	1.05	0.56	0.77	2.06	...

where f_k is the calculated frequency value of the kth hummed note. Similarly, the relative duration difference (RDD) is calculated as the ratio of the durations of two consecutive notes, and can be represented as

$$\mathrm{RDD}(k) = \frac{t_k}{t_{k-1}} \tag{6}$$

where t_k is the duration of the kth hummed note.

In contrast to common practice in the literature on melodic transcription techniques for QBH, we allow fractional number ratios because it accounts for finer levels of detail for user-centric variability, our main challenge in building a robust system. A human cannot be expected to hum a note transition perfectly, meaning that the pitch difference and duration ratios will not necessarily correspond to integers and simple ratios, respectively. This is one reason why we wish to record the variability as an unconstrained measure of error encountered in the humming query.

The resulting transcription is a 2 × N sequence, where N is the number of note transitions, with pitch transition information in the first row and duration ratios in the second row. A sample transcription output can be seen in Table III.

Without loss of generality, the first pitch transition is set to 0, and the first duration transition is initialized to 1, since the first hummed note has no prior reference. A positive pitch transition corresponds to a pitch ascent from the previous hummed note, and a duration ratio value larger than 1 refers to an increase in duration from the previous note value. The units for RPD is semitones, and there is no unit for RDD since it is ratio of two time measure.

B. Characteristic Fingerprints

To inform the design of our search and retrieval algorithms, we analyzed the collected data described in [27]. We performed a simple factor analysis of variance (ANOVA) test for different control groups, with respect to different criteria such as musical training, familiarity with the melody, and melodic complexity (perceived difficulty of production of melody). The statistical analysis of the collected data showed that the quality of the humming depends on a person's musical background and their familiarity with the melodies.

The analysis further showed that, regardless of musical training, more errors can be expected in the humming of larger note transitions than small intervals. It is observed that humming major and minor intervals that occur more naturally and frequently in the diatonic scale are easier than humming diminished or augmented intervals. Our plan is to incorporate these statistical findings in the design of our retrieval algorithms.

As mentioned in Section I, different types of retrieval systems have been proposed including the popular ED methods proposed in [5], [10], and [22]. In this section, the performance of the ED algorithm will be compared to that of our proposed statistical retrieval approach. Our proposed approach to retrieval under the effects of production uncertainty is to collect FPs, i.e., the salient portions of a melody, from the transcribed symbol sequence, and to use them in the search for a match in the database. Distinct from our previous efforts on retrieval (see [1]), the system reported in this paper more directly integrates the statistics gathered from detailed analysis of the humming database into the search calculations.

To identify characteristic points in the hummed piece, aspects of composing a melody are considered. It is hypothesized that note transitions in a tune can be considered distinctive if they are rare. For each humming query, the most distinct note transitions are located by searching for the highest levels of change over RPD and RDD. The largest value in RPD captures the most distinct pitch leap, and the largest value in RDD captures the most distinct duration change in the query. The selected FPs should contain these local characteristic points in the hummed input.

Large pitch and durations transitions are prone to error, thus having high uncertainty, a fact that can be explained by Fitt's Law [32]. Fitt's Law predicts that the time required to travel from a starting position to a target point area in space is a function of the distance to the target and the area of the target. This movement can be considered the vertical movement of the larynx along the cervical spine for controlling fundamental pitch (f_0) changes [33]. Vertical larynx movements for large pitch transitions is greater than that required for smaller pitch transitions. Therefore, given a constant time to finish a humming note and start the next one, users tend to make more errors producing large pitch transitions.

In the FPs, minimum changes in RPD and RDD, as well as the maximum changes, are considered. This is to account for the increased importance and descriptiveness of duration ratio information when pitch variability is minimized and similar increased pitch descriptiveness when minimum duration variability occurs. The four generic FPs selected in this work are created using the following indices:

$$(i_1, k_1) = \max(\mathrm{RPD}) \tag{7}$$

$$(i_2, k_2) = \min(\mathrm{RPD}) \tag{8}$$

$$(i_3, k_3) = \max(\mathrm{RDD}) \tag{9}$$

$$(i_4, k_4) = \min(\mathrm{RDD}) \tag{10}$$

where k_j is the position of the jth critical element spotted in the query input, and i_j is the value of the corresponding element $j = 1, \ldots, 4$.

For each $k_j, j = 1, \ldots, 4$, we create corresponding FPs, FP_j, as follows:

$$\mathrm{FP}_j = \begin{pmatrix} \mathrm{RPD}_{(k_j - R:k_j + R)} \\ \mathrm{RDD}_{(k_j - R:k_j + R)} \end{pmatrix}. \tag{11}$$

Each FP is a selected portion of the transcribed humming sequence. It has a radius R, where R is the number of note transitions in the transcribed humming sequence around the critical element k_j to be included as part of the FP_j, $j = 1, \ldots, 4$.

TABLE IV
SAMPLE FPs

RPD	-4.62	-0.33	**10.99**	-2.21	-3.71
RDD	0.48	0.86	2.42	0.94	1.06

FP1: largest pitch transition

RPD	-0.81	-4.79	**0.07**	2.44	-1.92
RDD	1.05	0.56	0.77	2.86	1.18

FP2: smallest pitch transition

RPD	-4.79	0.07	2.44	-1.92	6.54
RDD	0.56	0.77	**2.86**	1.18	0.91

FP3: largest duration transition

RPD	-1.92	6.54	-1.82	-4.62	-0.35
RDD	1.18	0.91	**1.01**	0.48	0.86

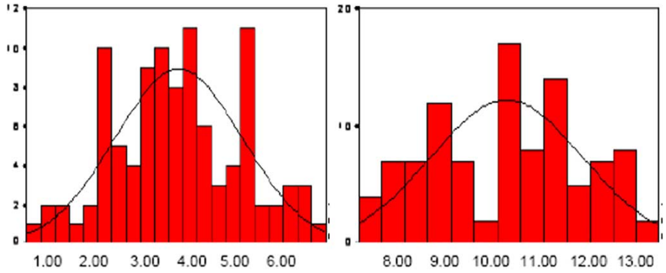
FP4: smallest duration transition

Hence, each FP$_j$ results in a matrix of dimension $2 \times (2R+1)$. For example, if the largest pitch transition is located at the k_1th element of the input sequence, then FP$_1$ is a segment of the original input that spans the note transitions $[k_1 - R, k_1 + R]$. The first row of this substring carries the pitch transcription RPD, and the second row the duration transcription RDD.

The number of FPs selected in this paper is four (FP$_1$ to FP$_4$); note that this number can be extended to capture other characteristic points in the melody. While adding more FPs to the retrieval calculations can be helpful in finding better matches by introducing new characteristic sections of the query, the computation time increases dramatically with the number of FPs since each FP is aligned separately to the database entries. Having fewer FPs may be computationally efficient, but can limit the FPs' ability to carry distinctive melodic information. On average, the coverage of four FPs spans 10-15 note transitions; most of the time, the FPs overlap, and subsequences of such lengths are found to be adequate in our experiments.

Table IV shows some FPs extracted from a hummed sample of "Happy Birthday," for which a partial transcription is presented in Table III. FP$_1$ is centered on the absolute highest pitch transition in the humming sequence and has a radius $R = 2$, resulting in a 2×5 matrix using the neighboring two cells. FP$_2$ is centered on the smallest pitch transition, FP$_3$ is based on the largest duration change, and FP$_4$ on the smallest duration change. In earlier studies [1], we investigated the effect of selecting different R values. When R is too small, the FP does not carry sufficient information for the query calculations to be effective. On the other hand, when R is too large, the FP contains more uncertainty. Retrieval analysis showed that $R = 2$ provided the best performance for our experiment conditions.

C. Prediction Intervals



Transition	4 semitones	12 semitones
Mean Std PCI	3.80 1.08 $1.603 < x < 6.042$	10.53 1.80 $8.316 < x < 12.766$

Fig. 10. Histogram and statistics for two note transitions (one 4-semitone transition, and one 12-semitone transition) each performed 100 times. For the bar graphs, horizontal axes mark the semitones and vertical axes show the occurrences for the corresponding semitone window in the 100 performances.

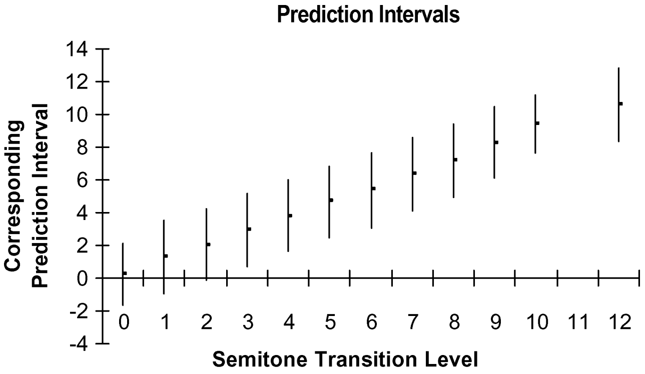
Considering the wide range of musical backgrounds amongst the data collection experiment participants, high levels of uncertainty and variability in the humming samples was observed. There exists a tradeoff between tolerating high levels of variability, and achieving high accuracy in the retrieval system. High tolerance may lead to radical changes in the interpreted melodic structure that could eventually result in erroneous matches in search calculations. Our solution to the handling of this tradeoff is statistical and data driven. For each level of pitch and duration transitions, we build prediction confidence intervals (PCIs) that are used in guiding the search engine as shown in Fig. 10.

The note transitions performed by the participants, ranging from 1 to 12 semitones, are each found to be normally distributed with $p \leq 0.05$, according to the Kolmogorov–Smirnov test. We used 100 sample transitions, randomly chosen from our humming database, for training these prediction intervals. Fig. 10 shows the prediction intervals for the cases of 4-semitone and 12-semitone pitch transitions in the humming of the melodies "London Bridge is Falling Down" and "Happy Birthday," respectively.

The PCI for a 4-semitone leap, PCI[4], gives the limits between which a performed transition may fall into the category of a semitone difference of 4. Referring back to FP1 in Table IV, since the center RPD value of 10.99 falls outside the allowable limits for PCI[4], the query engine will not search for a similar pattern around a 4-semitone pitch transition in the database.

Fig. 11 shows the distribution of the minimum, maximum, and mean number of semitone steps allowable for each pitch transition, and a table of the calculated confidence limits. These numbers are obtained from real world humming data and reflect the larger uncertainty introduced by the inclusion of nonmusically trained users. For example, the performance for a unison transition is distributed within the limits [$-$1.62,2.19]. During regular humming performance, the accuracy of humming a pitch transition is largely affected by the previous and upcoming notes and the rhythmic structure of the melody at that particular point. This is why instead of collecting task specific data for certain transitions, transition data within melodies at random locations were collected, which leads to high levels of variance.

The PCI for an 11-semitone transition, corresponding to a major 7th, was not calculated due to a lack of data. For this



Semitones	Number of Samples	Lower Confidence Limit	Upper Confidence Limit
0	100	-1.6152	2.1290
1	100	-.9192	3.5257
2	100	-.1258	4.2316
3	100	.7628	5.2031
4	100	1.6034	6.0422
5	100	2.4437	6.8817
6	18	3.0130	7.6954
7	100	4.1233	8.5615
8	100	4.9626	9.4019
9	100	6.1208	10.4510
10	24	7.6749	11.2312
11	-	-	-
12	100	8.3168	12.7666

Fig. 11. Prediction confidence intervals. Sparse data for 11-semitone transition. Limits are in semitone units. $(p \leq 0.05)$.

case, a virtual distribution is created by linear interpolation of the other distributions in order to detect the sample distribution empirically as needed during error calculations. According to this table, the center value for FP$_1$ (i_1 at the k_1th transcribed interval in the query input, in Table IV), 10.99, may represent a 10-, 11-(most likely), or 12-semitone transition. PCI calculations and such parameter estimations are done using a randomly selected development set, and this development set was withheld from performance evaluation tests.

D. Error Calculations and Similarity Measurement

After all four FPs are extracted, an error value for each of the FPs, against possible matches in the database, is calculated. Continuing with the example of FP$_1$ from Table IV. Having determined that the center pitch transition value of 10.99 (i_1 at the k_1th transcribed interval in the query input) could represent 10, 11, or 12 semitone leaps, the query engine searches through the database for the pitch transition values, {10, 11, and 12}. For each candidate RPD value that is located, C is created, of size $2\times(2R+1)$ in the same way FPs are extracted from the input sequence, considering both the relative pitch difference RPD and the relative duration difference RDD. Next, the numerical difference, the total error (TE$_j$) is assessed between the query FP$_j$ and the candidate matrix C_j.

TABLE V
CANDIDATE SYMBOL SEQUENCES: C VERSUS CM . (a) EXACT CANDIDATE MATRIX C MOST SIMILAR TO FP$_1$. (b) EXPECTED PRODUCTION MATRIX CM CLOSEST TO FP$_1$

RPD	-5	0	12	-3	-4
RDD	0.25	1	2	1	1

(a)

RPD	-4.66	0.25	10.53	-2.98	-3.85
RDD	0.25	1	2	1	1

(b)

In an earlier study [1], the sum of the square differences between the components of the query FP and the original candidate matrix C was used:

$$\text{TE} = \sum_{u=1}^{2} \sum_{v=1}^{2R+1} (\text{FP}_{uv} - C_{uv})^2. \tag{12}$$

This previous approach does not take into account the difficulty in humming large pitch and duration transitions with high accuracy, as was found in our analyses of humming samples [27]. To account for typical errors made in the pitch transitions, rather than comparing the query with the exact pitch transitions, the FPs are matched against the expected production matrix CM instead of the original C:

$$\text{TE} = \alpha_1 \times \sum_{v=1}^{2R+1} (\text{FP}_{1v} - CM_{1v})^2 + \alpha_2 \times \sum_{v=1}^{2R+1} (\text{FP}_{2v} - CM_{2v})^2 \tag{13}$$

where the first term calculates the pitch error, and the second term calculates the duration error. α_1 and α_2 can be used to apply any weighting preference. In our experiments, the weights are selected to be equal. CM is given by mapping each exact transition to its expected value based on its PCI derived from actual humming data. No modifications are made for the duration data. The error value TE$_j$ for each FP, FP$_j$, is calculated.

Consider FP$_1$ from Table IV. Table V(a) shows the most similar original candidate matrix C to the query "Happy Birthday," and Table V(b) the corresponding expected production matrix CM. The error value between FP$_1$ and the closest CM shown is TE $= 1.4391$.

For each melody q in the database, if it contains the local feature within the defined prediction interval, a candidate segment $C(q)$ is extracted and mapped to its expected production matrix $CM(q)$. An error value TE(q) is then calculated between the query FP and $CM(q)$. Next, the candidate melody is assigned a local similarity measure SL by normalizing the TE values to "1" and subtracting the normalized value from 1 as follows:

$$SL(\text{FP}_1) = 1 - \text{norm(TE)} \tag{14}$$

This ensures that melodies with lower TE error values are assigned a higher SL value, and melodies with higher TE error values are assigned lower SL values. Similar steps are applied to the other three remaining FPs for the same query, FP$_2$, FP$_3$, and FP$_4$. A composite similarity measure S is computed as the product of all four SL values for each candidate melody q as follows:

$$S(q) = \prod_{j=1}^{4} SL(\mathrm{FP}_j). \tag{15}$$

In our system, the top f scoring melodies (we chose $f = 5$ in our analysis below) are returned to the user as the query result.

E. ED Approach

A common way to define a distance metric between two symbol sequences is the ED. Several researchers, such as [22], have adopted the ED method in their QBH systems for assessing similarity between transcribed audio and database entries. Since the input query and database elements are concurrent pitch and duration information, the ED approach can then be employed to find the least cost alignment between the two, and the cost can be incorporated into the difference measure.

The ED between the compared sequences can be in the form of insertions, deletions, and transformations. With appropriate selection of the cost function, the ED can also represent the uncertainty that is expected in the humming audio input, which takes user errors into account. Insertion cost covers extra hummed notes, while the deletion cost accounts for skipped notes. The transformation cost penalizes error between a performed transition and the expected reference transition.

Here, a ED algorithm is presented that is based on the earlier efforts in the literature, in order to make performance comparison analysis. The existing algorithms are modified and a version, which is suitable to the way audio-to-symbol transcription is performed, and to the way the melodies in the database are stored in our work, was created.

This algorithm can be implemented by completing an $(M+1)\times(N+1)$ distance matrix $D(I,C)$ that calculates the distance between the $2 \times M$ transcribed audio input I and the $2 \times N$ candidate database entry C.

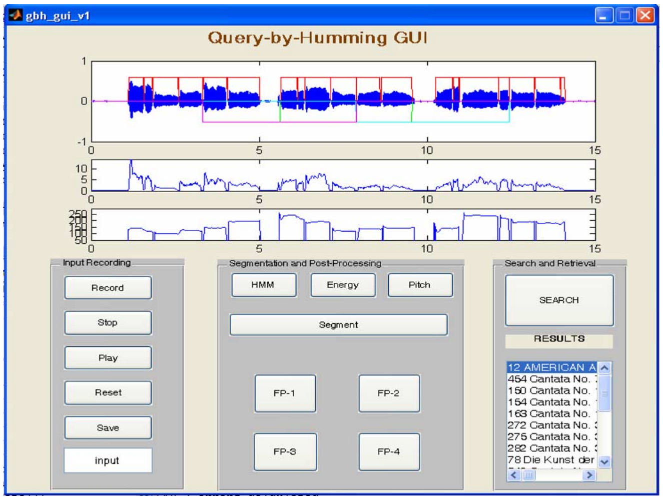
Let $2 \le i \le M+2$ and $2 \le j \le N+2$. The following recursive formula is used to calculate the value in each cell in D:

$$\begin{aligned} D_{i,j} &= \min\{D_{i-1,j}+1, D_{i,j-1}+1, D_{i-1,j-1}+\mathrm{Cost}_{i,j}\} \\ \mathrm{Cost}_{i,j} &= \frac{1}{2}\left\{\left|\frac{I_{1,i-1}-C_{1,j-1}}{12}\right|\right\} \\ &\quad + \frac{1}{2}\left\{\left|1-\frac{\min\{I_{2,i-1}, C_{2,j-1}\}}{\max\{I_{2,i-1}, C_{2,j-1}\}}\right|\right\} \end{aligned}$$

with initial conditions:

$$\begin{aligned} D_{1,1} &= 0, \\ D_{i=2:M+1,1} &= 1,2,3,\ldots,M \\ D_{1,j=2:N+1} &= 1,2,3,\ldots,N. \end{aligned}$$

The recursive formula above defines a constant penalty of 1 for note insertions and deletions, a transposition cost for interval errors (normalized to one octave), and a duration ratio difference cost. The Cost function is the sum of the absolute pitch difference and the absolute duration ratio difference for the particular cell, normalizing to 1 to be consistent with the insertion and deletion penalty. The final ED between two sequences is given by the minimum value appearing in the final row of the distance matrix $D(I,C)$.



12 AMERICAN A
454 Cantata No.
150 Cantata No.
154 Cantata No.
163 Cantata No.
272 Cantata No.
275 Cantata No.
282 Cantata No.
78 Die Kunst der

Fig. 12. Graphical user interface for the proposed QBH system.

F. Graphical User Interface

Fig. 12 shows a screenshot of the Matlab graphical user interface (GUI) designed for the proposed system. The user can record his/her humming through the microphone, or audio samples can be loaded from a file. When the recording is finished or the upload completed, the *STOP* button prompts the GUI panel to display the waveform, and the short term energy and the pitch vector. This input can be played by clicking on the *PLAY* button, or saved to a ".wav" file of name specified by the text block by pressing the *SAVE* button.

The *HMM*, *ENERGY*, and *PITCH* buttons perform the HMM-based segmentation and postprocessing. The *SEGMENT* button draws the segmentation results on the input waveform. The FPs can be extracted by selecting the $FP1$, $FP2$, $FP3$, and $FP4$ buttons; the system displays the corresponding FP package at the bottom of the waveform, and plays that segment of the audio input. The *SEARCH* button performs the necessary error calculations, probability assignments, and similarity assessments, and displays the top f results on the *RESULTS* panel.

VI. Experiments and Results

We evaluate both the FP and the ED algorithms. As mentioned earlier, our evaluation corpus consists of 400 samples selected from our humming database, distributed evenly between musically trained and nontrained participants. The data includes 100 samples each of the melodies "Happy Birthday," "London Bridge is Falling Down," "Take Me Out to the Ball Game," and "Hey Jude." Each humming sample is around 15 to 20 s long, and does not necessarily start from the beginning of the melody. Each target melody is in the database with 1500 more transcribed MIDI files.

Two different success measures are used: A) the first considers a retrieval successful only if the correct melody is returned at the top of the results list; and B) the second measure considers a retrieval successful if the correct melody is in the top f candidates. For A), the percentage accuracy shows, among the 400 melody queries, the percentage that return the correct song at the top of the results list (i.e., $f = 1$); and, for B), the percentage accuracy shows the percentage of the humming samples

TABLE VI
ACCURACY COMPARISON FOR FP ALGORITHM VERSUS ED METHOD. EVALUATION METHODS: A—CORRECT RETRIEVAL TOPS RESULT LIST, B—CORRECT RETRIEVAL WITHIN TOP FIVE. (a) PERCENTAGE ACCURACY RETRIEVAL RESULTS FOR QUERIES BY MUSICALLY TRAINED PARTICIPANTS AND THE RESULTS OF THE SIGNIFICANCE TEST FOR $p \leq 0.05$. (b) PERCENTAGE ACCURACY RETRIEVAL RESULTS FOR QUERIES BY MUSICALLY UNTRAINED PARTICIPANTS AND THE RESULTS OF THE SIGNIFICANCE TEST FOR $p \leq 0.05$

Test Database Size	500		1000		1500	
	A	B	A	B	A	B
FP	91	92	90	92	88	90
ED	90	90	88	89	86	89
Significance	No	No	No	Yes	No	No

(a)

Test Database Size	500		1000		1500	
	A	B	A	B	A	B
FP	73	77	72	74	70	72
ED	72	74	67	69	62	63
Significance	No	Yes	Yes	Yes	Yes	Yes

(b)

that achieved a correct hit within the top five candidate results $(f = 5)$.

Table VI shows the results for the two measures applied to both the FP and ED algorithms, for both trained and nontrained participants. Table VI(a) and VI(b) investigate the system's sensitivity to, or robustness with respect to, increasing reference database (MIDI collection) size.

As expected, retrieval accuracy results for humming queries by nontrained participants using the same test database are consistently lower than those for queries by musically trained subjects. This is because the queries formulated by the nontrained participants contain more uncertainty, or perhaps are more prone to transcription errors.

In general, the FP method achieves higher retrieval accuracy than the ED method. For trained participants' humming queries, as shown in Table VI(a), both the FP and ED approaches performed comparably well. This means that both algorithms work well in the presence of system-based errors and low levels of user-based production errors, and the difference between the performances are statistically insignificant.

In the case of nontrained participants' (noisy) queries, shown in Table VI(b), the FP approach clearly outperforms the ED method. McNemar's test confirms that the performance difference is statistically significant, with $p \leq 0.01$.

The ED distance cost function is also updated with humming statistics by replacing the candidate values of the database elements (C) with the mean and standard deviation values (CM) of the PCIs presented in Fig. 11. The new experiment reports up to 2% absolute improvement for the samples taken from musically trained subjects, and 1% decrease for the samples taken from musically untrained users when compared to the original version of the ED distance experiment. The differences in performance are statistically insignificant for all cases.

The results demonstrate that the data-driven statistical alignment of input to database entries via FPs makes the system more immune to user-dependent uncertainty. The main weakness of the ED algorithm is the introduction of more uncertainty by considering the entire input data stream. A more robust approach should perhaps treat the input audio as a combination of characteristic structures. Our work demonstrates promising results with the introduction of candidate characteristic structures through our FP approach and encourages further examination of the use of robust characteristics of music signals.

VII. Conclusion and Future Work

This paper presented a statistical approach to the problem of retrieval in the presence of uncertainty for QBH systems. Our research was motivated by the way humans perceive and reproduce music. Knowledge-based methods incorporating human understanding of music were combined with data-based models. Our statistical approach required the extraction of FPs from hummed melodies. The search for these FPs was informed by our previous findings on inter-person humming variability, calculated from a database of humming samples by both musically trained and untrained participants. Our results showed that the size of the test database and the subject's musical training are the main factors that determined the success of our approach.

We also implemented an ED approach for retrieval in order to compare the performance of the proposed FP algorithm to a more conventional approach. In our comparison for different test database sizes, the FP approach consistently outperformed the ED approach, and the results showed that the FP algorithm is more robust to user-dependent uncertainty, especially for the more noisy humming queries formulated by nonmusically trained participants.

One important question that one can ask at this point is how the performance of the retrieval system is affected by the accuracy of the front-end recognizer. For instance, for queries formulated by musically trained participants, where we can expect low levels of user-based errors, the dominant factor that impedes system performance are likely the system-based transcription errors, such as note insertions and deletions.

Future work will include further improvements on the front-end segmentation and recognition processes in order to achieve even better retrieval results, especially addressing the issue of mixed hummed and/or sung retrieval queries. We intend to use existing databases [26], our own, and combinations of the two to evaluate such schemes. The development of a scalable front-end segmentation algorithm will be critical to allow for unrestricted human audio input.

Moreover, better statistical models for the retrieval side of the system can also be developed, such as by extracting repeating patterns that could represent the most likely expected patterns observed in the user's humming input [34]. This, in turn, implies higher penalties for unlikely database elements that will eventually result in improved retrieval calculations. Conversely, we could study the occurrence of rare melodic patterns as signatures to help enable fast retrieval.

REFERENCES

[1] E. Unal, S. Narayanan, and E. Chew, "A statistical approach to retrieval under user-dependent uncertainty in query by humming systems," in *Proc. ACM SIGMM Int. Workshop Multimedia Inf. Retrieval*, New York, Oct. 2004, pp. 113–118.

[2] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith, "Query by humming: Musical information retrieval in an audio database," in *Proc. ACM Int. Conf. Multimedia*, San Francisco, CA, Nov. 1995, pp. 231–236.

[3] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson, and S. J. Cunningham, "Towards the digital music library: Tune retrieval from acoustic input," in *Proc. ACM Int. Conf. Digital Libraries*, Bethesda, MD, Mar. 1996, pp. 11–18.

[4] R. J. McNab, L. A. Smith, I. H. Witten, and C. L. Henderson, "Tune retrieval in multimedia library," *Multimedia Tools Applicat.*, vol. 10, pp. 113–132, Apr. 2000.

[5] N. Kosugi, Y. Nishihara, T. Sakata, M. Yamamuro, and K. Kushima, "Music retrieval by humming—Using similarity retrieval over high dimensional feature vector space," in *Proc. IEEE Pacific Rim Conf. Commun., Comput., Signal Process.*, Victoria, BC, Canada, Aug. 1999, pp. 404–407.

[6] P. Y. Rolland, G. Raskinis, and J. G. Ganascia, "Music content-based retrieval: An overview of Melodiscov approach and systems," in *Proc. ACM Int. Conf. Multimedia*, Orlando, FL, Nov. 1999, pp. 81–84.

[7] S. Blackburn and D. De Roure, "A tool for content based navigation of music," in *Proc. ACM Int. Conf. Multimedia*, Bristol, U.K., Sep. 1998, pp. 361–368.

[8] B. Chen and J.-S. R. Jang, "Query by singing," in *Proc. IPPR Conf. Comput. Vision, Graphics, Image Process.*, Taiwan, R.O.C., Aug. 1998, pp. 529–536.

[9] N. Hu and R. B. Dannenberg, "A comparison of melodic database retrieval techniques," in *Proc. ACM Int. Conf. Digital Libraries*, Portland, OR, Jul. 2002.

[10] Y. Zhu and D. Shasha, "Warping indexes with envelope transforms for query by humming," in *Proc. ACM SIGMOD Int. Conf. Manage.Data*, San Diego, California, June 2003.

[11] B. Pardo, J. Shifrin, and W. Birmingham, "Name that tune: A pilot study in finding a melody from a sung query," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 55, pp. 283–300, Feb. 2004.

[12] R. B. Dannenberg, W. P. Birmingham, G. Tzanetakis, C. Meek, N. Hu, and B. Pardo, "The MUSART testbed for query-by-humming evaluation," in *Proc. ISMIR Int. Conf. Music Inf. Retrieval*, Baltimore, MD, Oct. 2003.

[13] A. Ito, S.-P. Heo, M. Suzuki, and S. Makino, "Comparison of features for DP-matching based query-by-humming systems," in *Proc. ISMIR Int. Conf. Music Inf. Retrieval*, Barcelona, Spain, Oct. 2004, pp. 297–303.

[14] C. Parker, "Applications of binary classification and adaptive boosting to the query-by-humming problem," in *Proc. ISMIR Int. Conf. Music Inf. Retrieval*, London, U.K., Sep. 2005, pp. 245–251.

[15] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison*, Biological Sequence Alignment*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[16] M. Clausen, R. Engelbrecht, D. Meyer, and J. Shmitz, "Proms: A web-based tool for searching in polyphonic music," in *Proc. ISMIR Int. Conf. Music Inf. Retrieval*, Plymouth, MA, Oct. 2000.

[17] G. A. Wiggins, K. Lemstrom, and D. Meredith, "SIA(M)ESE:: An algorithm for transposition invariant, polyphonic content-based music retrieval," in *Proc. ISMIR Int. Conf. Music Inf. Retrieval*, Paris, France, Oct. 2002, pp. 283–284.

[18] R. Typke, P. Giannopoulos, R. C. Veltkamp, F. Wiering, and R. van Oostrum, "Using transposition distances for measuring melodic similarity," in *Proc. ISMIR Int. Conf. Music Inf. Retrieval*, Baltimore, MD, Oct. 2003, pp. 107–114.

[19] E. Ukkonen, K. Lemstrom, and V. Makinen, "Geometric algorithms for transposition invariant content-based music retrieval," in *Proc. ISMIR Int. Conf. Music Inf. Retrieval*, Baltimore, MD, Oct. 2003, pp. 193–199.

[20] A. Lubiw and L. Tanur, "Pattern matching in polyphonic music as a weighted geometric translation problem," in *Proc. ISMIR Int. Conf. Music Inf. Retrieval*, Barcelona, Spain, Oct. 2004, pp. 289–297.

[21] R. Clifford, M. Christodoulakis, T. Crawford, D. Meredith, and G. Wiggins, "A fast, randomized, maximal subset matching algorithm for document-level music retrieval," in *Proc. ISMIR Int. Conf. Music Inf. Retrieval*, Victoria, Canada, Oct. 2006, pp. 150–155.

[22] S. Pauws, "CubyHum: A fully operational query by humming system," in *Proc. ISMIR Int. Conf. Music Inf. Retrieval*, Paris, France, Oct. 2002, pp. 187–196.

[23] L. P. Clarisse, J. P. Martens, M. Lesaffre, B. De Baets, H. De Meyer, and M. Leman, "An auditory model based transcriber of singing sequences," in *Proc. ISMIR Int. Conf. Music Inf. Retrieval*, Paris, France, Oct. 2002, pp. 116–123.

[24] H.-H. Shih, S. Narayanan, and C.-C. J. Kuo, "An HMM-based approach to humming transcription," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Laussanne, Switzerland, Aug. 2002, pp. 337–340.

[25] H.-H. Shih, S. Narayanan, and C.-C. J. Kuo, "Multidimensional humming transcription using phone level hidden Markov models for query by humming systems," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Hong Kong, Apr. 2003, pp. I-61–I-64.

[26] "MIREX 2006 Evaluation Tasks: QBSH (Query-by-Singing/ Humming)." [Online]. Available: http://www.music-ir.org/ mirex2007.

[27] E. Unal, S. Narayanan, E. Chew, H.-H. Shih, and C.-C. J. Kuo, "Creating data resources for designing user-centric front-ends for query by humming systems," *Multimedia Syst. J.*, vol. 10, pp. 475–483, May 2005.

[28] W. Hewlett, "Muse data: An electronic library of classical music scores." [Online]. Available: http://www.musedata.org

[29] E. Unal, S. Narayanan, E. Chew, P. Georgiou, and N. Dahlin, "A dictionary based approach for robust and syllable-independent audio input transcription for query by humming systems," in *Proc. 1st ACM Workshop Audio Music Comput. Multimedia*, Santa Barbara, CA, Oct. 2006, pp. 37–44.

[30] D.-B. Paul and J.-M. Baker, "The design for the wall street journalbased CSR corpus," in *Proc. HLT Workshop Speech Natural Lang.*, Harriman, New York, 1992, pp. 357–362.

[31] C. J. van Rijsbergen*, Information Retrieval*, 2nd ed. Glasgow, U.K.: Dept. Comput. Sci., Univ. Glasgow, 1979.

[32] P. M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement," *J. Experimental Psychol.*, vol. 47, pp. 381–391, 1954.

[33] K. Honda, H. Hirai, S. Masaki, and Y. Shimada, "Role of vertical larynx movement and cervical Lordosis in F0 control," *Lang. Speech*, vol. 42, pp. 401–411, 1999.

[34] H.-H. Shih, S. Narayanan, and C.-C. J. Kuo, "A dictionary approach to repetitive pattern finding in music," in *Proc. IEEE Int. Conf. Multimedio Expo*, Tokyo, Japan, 2001, pp. 281–284.

Erdem Unal (S'07) received the B.S. and M.S. degrees from the University of Southern California (USC), Los Angeles, in 2002 and 2004, respectively. He is currently pursuing the Ph.D. degree in electrical engineering at USC.

Since 2002, he has been a Research Assistant in the Speech Analysis and Interpretation Laboratory (SAIL), USC. His research interests are in audio signal processing and music information retrieval. His current research includes, query by humming, query by example, music fingerprinting, retrieval with expressive performances, tonality modeling, and uncertainty quantification.

Elaine Chew (M'05) received the B.A.S. degree (with honors) in mathematical and computational sciences with honors and in music with distinction, from Stanford University, Stanford, CA, in 1992, and the S.M. and Ph.D. degrees in operations research from the Massachusetts Institute of Technology, Cambridge, MA, in 1998 and 2000, respectively.

She joined the University of Southern California (USC) Viterbi School of Engineering, Los Angeles, in the fall of 2001, where she is Associate Professor of Industrial and Systems Engineering and of Electrical Engineering. She is the first honoree of the Viterbi Early Career Chair and serves as a Research Area Director of the Integrated Media Systems Center, a National Science Foundation (NSF) Engineering Research Center. At USC, she founded and heads the Music Computation and Cognition Laboratory, where she conducts and directs research at the intersection of music and engineering. Awards and support for her pioneering work include the Presidential Early Career Award in Science and Engineering and the NSF Early Career and Information Technology Research grants. She is currently a 2007–2008 Fellow at the Radcliffe Institute for Advanced Study at Harvard University. Her research spans the areas of automated music analysis and visualization, expressive performance analysis and synthesis, and music information retrieval. She has authored/co-authored over 50 refereed journal and conference papers on mathematical and computational modeling of music. She serves on the founding editorial boards of the *Journal of Mathematics and Music* (2006-), *ACM Computers in Entertainment* (2003-), and *Journal of Music and Meaning* (2004-), and on the editors' panel of *Computing in Musicology* (2005-). She has served as guest editor and organizer of special clusters on computation in Music for the *INFORMS Journal on Computing* (Summer 2006), and on music visualization for *ACM Computing in Entertainment* (October 2005). She is guest editing a special issue on computation for the *Journal of Mathematics and Music*.

Prof. Chew has served on the program and scientific committees of the IEEE International Workshop on Multimedia Information Processing and Retrieval (2006, 2007), International Conferences on Sound and Music Computing (2007), New Interfaces for Musical Expression (2006, 2007), and Music and Artificial Intelligence (2002), and the IJCAI International Workshop on Artificial Intelligence and Music. She is on the organizing committee for the 2008 National Academy of Sciences Kavli German–American Frontiers of Science meeting, and serves on the steering committee of, and is the 2008 program cochair for, the International Conference on Music Information Retrieval.

Panayiotis G. Georgiou (M'02) received the B.A. and M.Eng. degrees (with honors) from Cambridge University (Pembroke College), Cambridge, U.K., in 1996 and the M.Sc. and Ph.D. degrees from the University of Southern California, Los Angeles, in 1998 and 2002, respectively.

Since 2003, he has been a member of the Speech Analysis and Interpretation Laboratory, Department of Electrical Engineering, University of Southern California, Los Angeles, first as a Research Associate and currently as a Research Assistant Professor. His interests span the fields of human social and cognitive signal processing. He has worked on and published over 30 papers in the fields of statistical signal processing, alpha stable distributions, speech and multimodal signal processing and interfaces, speech translation, language modeling, immersive sound processing, sound source localization, and speaker identification. His current focus is on multimodal cognitive environments and speech-to-speech translation.

Dr. Georgiou was awarded a Commonwealth scholarship from Cambridge-Commonwealth Trust from 1992–1996.

Shrikanth S. Narayanan (S'88–M'95–SM'02) received the Ph.D. degree from the University of California, Los Angeles, in 1995.

He is Andrew J. Viterbi Professor of Engineering at the University of Southern California (USC), Los Angeles, where he holds appointments as Professor in electrical engineering and jointly in computer science, linguistics, and psychology. Prior to joining USC, he was with AT&T Bell Labs and AT&T Research, first as a Senior Member, and later as a Principal Member of its Technical Staff from 1995–2000. At USC, he is a member of the Signal and Image Processing Institute and a Research Area Director of the Integrated Media Systems Center, an NSF Engineering Research Center. He has published over 235 papers and has 14 granted/pending U.S. patents.

Dr. Narayanan is a recipient of an NSF CAREER Award, USC Engineering Junior Research Award, USC Electrical Engineering Northrop Grumman Research Award, a Provost Fellowship from the USC Center for Interdisciplinary Research, a Mellon Award for Excellence in Mentoring, and a recipient of a 2005 Best Paper Award from the IEEE Signal Processing Society. Papers by his students have won best student paper awards at ICSLP'02, ICASSP'05, and MMSP'06. He is an Editor for the *Computer Speech and Language Journal* (2007-present) and an Associate Editor for the IEEE *Signal Processing Magazine*. He was also an Associate Editor of the IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING (2000–2004). He serves on the Speech Processing and Multimedia Signal Processing technical committees of the IEEE Signal Processing Society and the Speech Communication Committee of the Acoustical Society of America. He is a Fellow of the Acoustical Society of America and a member of Tau Beta Pi, Phi Kappa Phi, and Eta Kappa Nu.