# A STATISTICAL APPROACH TO HUMMING RECOGNITION

*Hsuan-Huei Shih, Shrikanth S. Narayanan and C.-C. Jay Kuo*

Integrated Media Systems Center and Department of Electrical Engineering
University of Southern California, Los Angeles, CA 90089-2564
Tel: (213) 740-8386, Fax: (213) 740-4651, E-mail: {hshih,shri,cckuo}@sipi.usc.edu

## ABSTRACT

A statistical pattern recognition approach applied to human humming data is examined in this research. Query by humming provides a natural means for content-based retrieval from music databases. The proposed system aims at providing a robust front-end for such an application. The segment of a note in the humming waveform is modeled by a hidden Markov model (HMM) while data features such as pitch measures are modeled by a Gaussian mixture model (GMM). Preliminary real-time recognition experiments are carried out based on humming data obtained from eight users and an overall correct recognition rate of around 80% is demonstrated.

## 1. INTRODUCTION

Content-based multimedia data retrieval is an emerging research area. Enabling natural interaction with multimedia databases is a critical component of such efforts. Music databases form a significant portion of media applications, and there is a great need in developing methods for indexing and interacting with them. Querying music databases using human humming as the query key has recently gained attention as a viable option [1]. This requires signal processing for automatically mapping human humming waveforms to symbol strings representing the underlying melody and tempo contours. This paper focuses on automatic humming recognition.

Most approaches to humming recognition that have been proposed for Query By Humming systems are based on non-statistical signal processing. They include methods based on autocorrelation [1, 2, 3] and filter banks [4]. A major problem with non-statistical approaches is robustness to inter-speaker variability and other signal distortions. Users hum with varying levels of accuracy. Most users, especially those with minimal or no musical training, do not hum very accurately. Hence, methods attempting to estimate the exact melody and tempo of humming tend to be prone to errors. In turn, these errors render the pattern searching process during database retrieval complicated. Most deterministic methods hence provide only a course melody contour, e.g. labeled in terms of rising/stable/falling pitch.

In this work, we target at a finer description of melody contour, i.e., at the note level. The proposed statistical approach provides note level decoding. Since it is data-driven, it is more amenable to robust processing in terms of handling variability. Conceptually, the approach tries to mimic a human's perceptual processing of humming as against attempting to model the production of humming. Such statistical approaches have been a great success in automatic speech recognition and can be adopted and extended to recognizing human humming and singing. In particular, we focus on recognizing the melody contour of a piece of humming. A melody contour is assumed to be a sequence of notes in a piece of humming defined by their detected pitch values. A note model is defined by a hidden Markov model with features modeled by Gaussian mixture models. During the training phase, note models are trained with humming data obtained from real users. During recognition, the incoming piece of the humming waveform is decoded by using the trained note models.

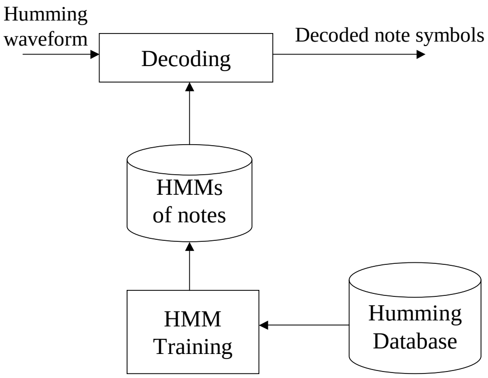

**Fig. 1**. The functional block diagram of the humming recognition system.

## 2. METHODS

Our approach to humming recognition is summarized in Fig. 1. Like any data-driven pattern recognition approach, models are derived from data representing the underlying classes for recognition. Details of the database preparation are given in Sec. 2.1. Statitistical modeling assumes the availability of signal-derived features that provide discriminability for recognition. Selection of features for characterizing a humming note are addressed in Sec. 2.2.

### 2.1. Database Collection and Preparation

There is no publicly available database of human humming as of to date. Hence, the initial focus of our work was devoted to creating a humming database to enable both training and testing.

#### 2.1.1. *Humming Recording*

The preliminary database used in this work was collected from nine subjects, four females and five males. Users were asked to hum specific melodies using a stop consonant-vowel syllable (such as "da" or "la") as the basic sound unit. Each person was asked to hum three different melodies that included the forward C major scale, the backward C major scale, and a short song "2 little tigers". One of the male's humming was deemed highly inaccurate by informal listening and hence was excluded from our experiments. The recordings were done by using a high-quality close talking Shure microphone (model: SM12A-CN) at 44.1kHz and mini disk recorders in a quiet office environment. Recorded signals were transferred to a computer and low-pass filtered at 8kHz to reduce noise and other high frequency components that are outside the normal human humming range.

#### 2.1.2. *Data Transcription*

A humming piece contains a sequence of notes, and these notes were segmented and labeled by human listeners. Since this was an early investigation, manual segmentation of notes was included to provide comparisons against automatic methods. In practice, very few people can hum a note, for example, a *center-do (C5)* in the Western classical scale, at the same pitch every time without the aid of any reference pitch. Therefore, the use of absolute pitch values to label a note was not deemed to be a viable option. Since the goal of this work is in mapping the humming signal to notes, a more robust and general method is to focus on relative changes in pitch values of a melody contour. A note has two main attributes, namely, the pitch (measured by the fundamental frequency of voicing) and the duration. Hence, relative pitch values were used to label a humming piece instead of absolute pitch values.

Two different labeling conventions were considered. The first one uses the first note's pitch as the reference pitch for labeling the subsequent notes in the rest of the signal. Let *"R"* denote the reference note; let *"Dn"* and *"Un"* represent notes that are lower or higher in pitch with respect to the reference by $n$ half-steps. For example, a humming piece corresponding to *do-re-mi-fa* will be labeled as *"R-U2-U4-U5"* while the humming corresponding to *do-ti-la-sol* will be labeled as *"R-D1-D3-D5"* where *"R"* is the reference note, *"U2"* denotes a pitch value higher than the reference by two half-steps and *"D1"* denotes a pitch value lower than the reference by one half-step. The numbers following *"D"* or *"U"* are variable, and depend on the humming data. The second labeling convention is based on the rationale that a human is sensitive to the pitch value of adjacent notes rather the first note. According to this convention, the humming piece for *do-re-mi-fa* will labeled as *"R-U2-U2-U1"*, and a humming piece corresponding to *do-ti-la-sol* is labeled as *"R-D1-D2-D2"*.

Transcriptions were saved in separate files and were used during supervised training of note models and to provide the reference transcriptions to evaluate recognition results.

### 2.2. Features for Humming

The choice of good features is key to a good humming recognition performance. Since human humming production is similar to speech, features used to characterize a phoneme in automatic speech recognition (ASR) are considered for modeling notes in humming recognition. Features used in ASR include linear prediction coefficients or filterbank coefficients, energy measures and their derivatives. While pitch is typically ignored in most ASR systems, it is essential to the current work. Our base feature set included mel-frequency cepstral coefficients (MFCC), the energy measure and pitch to characterize humming notes. The proposed 14-element feature vector contained 12 MFCCs, 1 energy measure, and 1 pitch ratio component. Pitch values were not used directly since different users (and their hummings) may have different keys. The difference between current pitch and reference pitch was calculate in the log scale as

$$\log(current\ pitch) - \log(reference\ pitch) \qquad (1)$$

As discussed in Sec. 2.1.2, the reference pitch can be the pitch of the first note, or the pitch of the immediate previous note.

#### 2.2.1. *Mel-Frequency Cepstral Coefficients*

Mel-Frequency Cepstral Coefficients (MFCCs) are obtained through a nonlinear filterbank analysis motivated by human hearing mechanisms. They are popular in ASR. Filterbank analysis is first done to yield log energy values of *N* Mel-Frequency filterbank channels. These cepstral coefficients are then calculated from filterbank energies by applying the Discrete Cosine Transform (DCT) according to

$$mfcc_i = \sqrt{\frac{2}{N}} \sum_{j=1}^{N} \log(x_j) \cos \quad \frac{i\pi}{N}(j-0.5) \quad , \qquad (2)$$

where $mfcc_i$ is the *i-th* Mel-Frequency Cepstral Coefficient, and $x_j$ is the *j-th* Mel-Frequency filterbank's energy. For our analysis, 26 filterbank channels were chosen and the first 12 MFCCs were selected as features.

#### 2.2.2. *Energy Measure*

Energy is an important feature in humming recognition especially to provide temporal segmentation of notes. Typically a distinct variation in energy will occur during the transition from one note to another. This effect is especially enhanced since users were asked to hum using basic sounds that were a combination of a stop consonant and a vowel (e.g., "da", "la").

#### 2.2.3. *Pitch Tracking*

Unlike speech recognition, accurate estimation of the pitch value is integral to detecting notes in humming. As mentioned previously in Sec. 2.1.2, very few people can hum very accurately, and errors in pitch estimation should be minimized to reduce additional mismatches in pattern recognition. At the same time, we desire a computationally efficient scheme for real-time humming recognition. Several pitch tracking schemes have been proposed in the past, and two of these were considered: the simple short time autocorrelation method [5] and the more complex normalized cross-correlation method [6]. While autocorrelation method has a smaller computation cost, normalized cross-correlation scheme yields more accurate pitch values.

### 2.3. HMM Definitions

Hidden Markov models (HMM) with Gaussian mixture models (GMM) for observations corresponding to each state of the HMM are used to define a note model. Each note was modeled by a 3-state left-to-right HMM. The use of HMM provides the ability to model temporal aspects of a note especially in dealing with time elasticity. The features (Sec. 2.2) corresponding to each state occupation in an HMM are modeled by a mixture of 2 Gaussians. Although the use of 4 mixtures gave slightly better recognition results, the complexity of the training and the decoding processes were much higher than the advantage provided by a slight performance gain. Depending on the labeling convention, two possible sets of HMMs can be derived. The first method uses the first note of the sequence as its reference while the second method uses adjacent notes as the reference. In general, for a coverage over 2 octaves, we require 49 distinct note models representing ascending and descending notes each separated by a half-step on a chromatic scale. For the data available in our experiment, only 15 of these models were covered and hence trained for the first method (R, U2, U4, U5, U7, U9, U11, U12, D1, D3, D5, D7, D8, D10, D12). Only 5 models were represented by using the second method (R, U1, U2, D1, D2). In general, the number of models required depends on the specific music data; for example, if the melodies involved exhibit smooth pitch changes from note to note, the second method offers an advantage in terms of fewer number of models required. The decoding process during recognition is slightly modified. The first note is detected but the actual pitch value is discarded, since the first note is always labeled as the reference note R (for both methods).

### 2.4. Training Process

The parameters of HMMs are estimated during a supervised training process by using a maximum likelihood approach with Baum-Welch re-estimation. The Hidden Markov Model Toolkit (HTK) was used in both training and decoding processes [7]. As a part of the training process, we empirically investigated: 1) the choice of the frame size and the frame rate, and 2) the need for manual segmentation. For simplicity sake, these first set of experiments were based on just two models, i.e. one for the generic notes and the other for 'silence'. Hence, they did not involve resolving individual notes based on pitch.

#### *2.4.1. Choosing Frame Size and Rate*

An important aspect of the front-end signal processing is the selection of the appropriate frame size and the frame rate for maximal recognition accuracy. The frame size relates to the frequency resolution of the spectral analysis. A large window gives good frequency resolution while compromises the short-time stationarity assumption regarding the signal. The frame rate, which provides the number of samples a frame is advanced, dictates the smoothness of the resulting short-time analysis. The choice of the frame size and rate were made based on experimental evaluations. The recognition performance on the training set was used as the criterion in deciding the optimal parameters under a variety of frame size/rate values: 15/3, 15/2, 20/3, and 30/10. Across speakers, a frame rate of 3 msec and a frame size of 20 msec gave the best results (see Table 1) and were adopted for the rest of our experiments.

#### *2.4.2. Effect of Manual Segmentation in Training*

The usefulness of bootstrapping the model training with hand segmented data was investigated by comparing it with a flat start approach. In manually segmented data, human listeners had marked each note's starting and ending points in a humming piece. In the flat start, the first training iteration used uniformly segmented data. Training and testing were done with the leave-one-out strategy that used data from all but one speaker for testing and the remaining speaker for training. A 20 msec frame size and 3 msec frame rate were used in the experiment. Several iterations of training were run in both approaches. Results showed similar convergence rates, i.e. both methods converged around 20 iterations. The resulting recognition rates were also similar. In fact, models trained with flat start provided slightly better recognition rates. These experiments show that handcrafted segmentation, a labor intensive process, is not necessary. seven subjects were used to train and the remaining subject's humming data were used in testing with the leave-one-out stragy. The Viterbi algorithm [8] was used for decoding. Two measures of performance are used: the correct rate, which accounts for note deletion and substitution errors, and the accuracy rate, which includes insertions as well.

**Table 1**. Closed-set recognition rates for various Frame size/ Frame rate values.

| Frame size/ Frame rate | 15/3 | 15/2 | 20/3 | 30/10 |
|---|---|---|---|---|
| Female 1 | 98.75 | 98.75 | 99.38 | 94.69 |
| Female 2 | 100 | 100 | 100 | 100 |
| Female 3 | 100 | 100 | 99.69 | 98.75 |
| Female 4 | 94.69 | 94.69 | 94.69 | 11.25 |
| Male 1 | 63.75 | 99.69 | 99.38 | 99.69 |
| Male 2 | 100 | 100 | 100 | 96.56 |
| Male 3 | 99.69 | 100 | 100 | 99.38 |
| Male 4 | 100 | 100 | 100 | 99.69 |
| Male 5 | 85.62 | 99.06 | 98.75 | 95.31 |
| Overall | 98.75 | 98.75 | 99.38 | 94.69 |

#### *2.4.3. Weighting Selection of Models*

Preliminary experiments showed a large number of insertions in the decoded results and hence a poor accuracy rate. One possible way to deal with insertions is to penalize unnecessary transitions from exiting one note to entering the next note. Table 2 gives the results on different note insertion penalties. It is seen that manipulating the note insertion penalty factor helps improve insertion rates while maintaining high correct recognition rates (a penalty factor of -30 gives the best results in this experiment). Another possible way to improve the performance is by adding a background noise model. One reason for numerous insertion errors is due to background noise which causes spurious segments. The noise model and note models were used to do the segment decoding. As shown in Table 3, the use of the noise model improves note recognition even without additional penalty.

#### *2.4.4. Decoding*

The decoding process during recognition is slightly modified compared to the training phase. The first note in every utterance is detected and is labeled as a reference note. After the first note is detected, the mean pitch value of the note will be calculated. Then, depending on the labeling conventions adopted, this reference pitch value will be used to calculate the log differential pitch

**Table 2**. The results of different note insertion log probability

| Penalty | correct rate % | accuracy rate % |
|---:|:---:|:---:|
| 0 | 100 | 85.63 |
| -10 | 100 | 89.38 |
| -20 | 100 | 94.06 |
| -30 | 100 | 96.88 |
| -40 | 99.38 | 97.19 |
| -50 | 98.13 | 97.19 |
| -60 | 95.63 | 95.63 |
| -70 | 94.06 | 94.06 |
| -90 | 89.69 | 89.69 |
| -100 | 87.81 | 87.81 |

**Table 3**. Effect of penalty in conjunction with a noise model

| Penalty | correct rate % | accuracy rate % |
|---:|:---:|:---:|
| -30 | 95.00 | 95.00 |
| -10 | 98.13 | 96.88 |
| -1 | 93.13 | 95.63 |
| 0 | 98.13 | 95.31 |
| 4 | 99.69 | 96.56 |
| 5 | 100 | 95.94 |
| 8 | 100 | 94.38 |

(i.e. the pitch ratio), defined in equation (1), to be included in the feature vector. Note that, for the second method, the reference pitch value during feature calculation gets updated every frame. These two different setups, of course, require different sets of models to be trained. Experimental results from these two methods will be discussed in Section 3.

## 3. EXPERIMENTAL RESULTS

As mentioned previously, the frame size was 20 msec and the frame update was 3 msec. The 14 element feature vector comprised 12 MFCCs, 1 energy measure and 1 pitch information element. A 3-state left-to-right HMM with 2 GMMs was used to model notes, and each model was trained for 20 times. The number of note models and the decoding process for the two labeling conventions were selected according to the details given in Section 2.3.

According to our definitions, the correct recognition rate is $(N - D - S)/N$ and the accuracy rate is $(N - D - S - I)/N$, where $N$ = number of correct notes, $D$ = no. of deletion errors, $S$ = no. of substitution errors, and $I$ = no. of insertion errors. When using the first frame's pitch as the reference pitch (method no. 1), the average accuracy rate across seven speakers was 51.18% and the correct rate was 70.59% evaluated by the leave-one-out method (trained on 7 speakers and tested on 1). Notice that the accuracy rate dropped to 40%, and the correct rate dropped to 66.76% when data from the "outlier" speaker (who had poor humming) were included in the test set. These results indicate that, while variability across speakers can be tolerated, inaccurate humming can drastically alter recognition results. Results by using the second method in which the previous note's pitch was used as the reference gave 81.47% correct rate, and 78.53% accuracy rate. These results support the assumption that recognition relies more on local phenomena. Further experiments on context-dependent modeling are needed to verify this claim.

## 4. CONCLUSION AND FUTURE WORK

A statistical approach to speaker-independent humming recognition was studied in this paper. Experimental results showed that our approach provides promising results over other non-statistical methods. The research results, however, are preliminary. First, a more comprehensive database of human humming from a larger set of human subjects needs to be gathered to enable detailed modeling and evaluation of the recognition performance. Such efforts are currently underway. Second, the role of inter-note context should be investigated through context-dependent modeling of notes. Third, the role of exploiting the underlying structure in music (such as repeated patterns) or a priori content knowledge (genre type) through statistical modeling of note sequences should be investigated. Finally, the performance of music retrieval based on the output of the humming recognizer should be investigated, especially from the viewpoint of recognition errors. A long term goal is to optimize the performance of the humming recognizer to maximize the overall retrieval accuracy instead of just minimizing recognition error rates.

## 5. REFERENCES

[1] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith, "Query by humming: musical information retrieval in an audio database," in *Proceedings of ACM Multimedia Conference'95*, San Francisco, California, November 1995.

[2] 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 *In Digital Libraries Conference*, 1996.

[3] Jyh-Shing Roger Jang, Hong-Ru Lee, and Ming-Yang Kao, "Content-based music retrieval using linear scaling and branch-and-bound tree search," in *2001 IEEE International Conference on Multimedia and Expo*, August 2001, pp. 405–408.

[4] H.-H. Shih, T. Zhang, and C.-C. Kuo, "Real-time retrieval of song from music database with query-by-humming," in *ISMIP 99*, 1999, pp. 251–57.

[5] Rabiner and Schafer, *"Digital Processing of Speech Signals"*, chapter 4, pp. 141–149, Prentice Hall, 1978.

[6] D. Talkin, *"Speech coding and synthesis"*, chapter 14 "A robust algorithm for pitch tracking (RAPT)", pp. 495–518, Elsevier, 1995.

[7] "Hidden markov model toolkit," URL: http://htk.eng.cam.ac.uk/.

[8] A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," in *IEEE Transaction on Information Theory*, 1967, vol. IT-13, pp. 260–267.