ICASSP 2003, Hong Kong

MULTIDIMENSIONAL HUMMING TRANSCRIPTION USING A STATISTICAL APPROACH FOR QUERY BY HUMMING SYSTEMS

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
E-mail: {hshih,shri,cckuo}@sipi.usc.edu

ABSTRACT

A new statistical pattern recognition approach applied to human humming transcription is proposed in this research. A music note has two important attributes, *i.e.* pitch and duration. The proposed algorithm generates multidimensional humming transcriptions, which contain both pitch and duration information. Query by humming provides a natural means for content-based retrieval from music databases, and this research provides 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 the pitch of the note is modeled by a pitch model using a Gaussian mixture model. Preliminary real-time recognition experiments are carried out with models trained by data obtained from eight human objects, 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 duration contours. This paper focuses on automatic humming recognition and transcription.

To enhance our previously proposed system in [2], we focus on recognizing both the melody contour and the duration contour of a piece of humming in this research. These two contours are combined to form a multidimensional humming transcription. The multidimensional transcription is assumed to be a sequence of notes in a piece of humming defined by their detected durations and pitch intervals. A note model is defined by a hidden Markov model with features modeled by Gaussian mixture models. A pitch model of a note is defined by pitch features and modeled by Gaussian mixture models. During the training phase, note and pitch models are trained with humming data obtained from real people. During recognition, the incoming piece of the humming waveform is decoded with trained note models for note segmentation, and then pitch values of segmented notes are detected with trained pitch models. The detection process is done in real-time.

The rest of the paper is organized as follows. In Sec. 3, the proposed algorithm is described in detail. Experimental results are given in Sec. 4 and conclusions and future work are presented in Sec. 5.

2. REVIEW OF PREVIOUS WORK

Most approaches to humming analysis, mainly developed for query by humming systems, use non-statistical signal processing. Ghias *et al.* [1], and Jang *et al.*[3] used autocorrelation to calculate pitch periods. McNab *et al.* [4, 5] adopted the Gold-Rabiner algorithm [6]. A major problem with the non-statistical approach is its robustness to inter-speaker variability and other signal distortions. Users, especially those with minimal or no music training, hum with varying levels of accuracy in terms of pitch and rhythm. As a result, most deterministic methods tend to use only a coarse melodic contour, *e.g.* labeled in terms of rising/stable/falling relative pitch directions [1]. This minimizes potential errors in the representation required for query and search. However, the scalability of this approach is somewhat limited. In particular, the representation is too rough to incorporate higher level music knowledge. Another problem with the non-statistical approach is the lack of real-time processing capability. Most of these methods rely on full utterance level feature measurements that demand buffering of long humming data, thereby limiting real-time processing.

To overcome the above two problems, we proposed an HMM-based humming recognition system in [2]. The HMM-based humming recognition system transcribed a humming piece into a melody contour. Some similar work was performed by Raphael [7, 8] and Durey *et al.* [9, 10]. Raphael [7, 8] attempted to solve the segmentation problem of automatic musical accompaniment, where hidden Markov models were used to segment notes and rests in a music piece while pitch information was discarded. Durey *et al.* [9, 10] used hidden Markov models to spot melody in music pieces. A small piece of query melody was passed into the melody spotting system, and then a list of music pieces containing the query melody were returned. Their approach was similar to word spotting in a speech recognition system. Both of them were based on instrumental or MIDI generated music pieces. These music pieces were accurate in tune and rhythm. Our proposed system aims at dealing with human humming which is much less accurate than instrumental music. The proposed system detects not only when a note is hummed but also which note is hummed.

3. PROPOSED ALGORITHM

Our approach to multidimensional humming transcription is summarized in Fig. 1. Similar to any data-driven pattern recognition approach, models are derived from data representing the underlying classes for recognition. Details of database preparation are given in Sec. 3.1. The proposed algorithm can be divided into two stages. At the first stage, a humming piece is first passed into the note decoder for note segmentation. At the second stage, a segmented note of the humming piece is then passed to the pitch detector for pitch tracking. The statistical models, selected features, training process and decoding process of the note segmentation stage are addressed in Sec. 3.2. Pitch feature selection, pitch analysis, and pitch model generation of the pitch tracking stage are described in Sec. 3.3. Finally, the generation of a multidimensional humming transcription is given in Sec. 3.4.

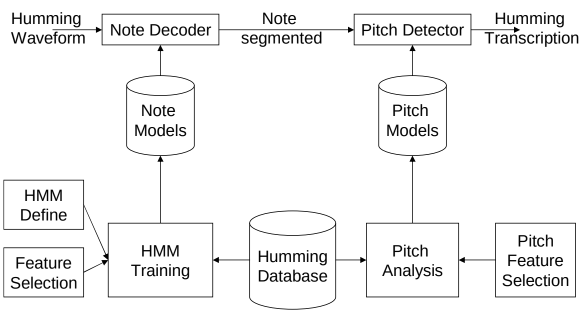

Fig. 1. The functional block diagram of the humming recognition system.

3.1. Database Collection and Preparation

There is no publicly available database for human humming. A small humming database was created in our previous work [2] for system design and preliminary experiments. The same database is used here and briefly described below.

A. Humming Recording

The recordings were done using a high-quality close talking Shure microphone (model: SM12A-CN) at 44.1kHz and a high-quality recorder in a quiet office environment. The recorder recorded humming pieces in uncompressed wave files. The wave files were then low-pass filtered at 8kHz and down sample to 16kHz to reduce noise and other high frequency components that are outside the normal human voice range.

B. Data Transcription

A humming piece contains a sequence of notes, which were labeled by human listeners. Manual segmentation of notes was included, since it provided the boundary information for pitch model training. 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 to map the humming signal to notes, a more robust and general method that focuses on relative changes in pitch values of a melody contour is adopted. 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.

The labeling convention is based on the rationale that a human is sensitive to the pitch value of adjacent notes rather than 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"* where *"R"* is the reference note or no change in the pitch value, *"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, depending on the humming data.

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

3.2. Note Segmentation

The first stage of the proposed algorithm is note segmentation, where the process of segmenting notes of a humming piece is conducted. First, a feature set which can characterize a note is chosen. Then, the HMM definition is chosen before training. During the training phase, notes' HMM models are trained using the selected feature set. Finally, the trained note model is used by the note decoder for note segmentation.

A. Feature Selection

The choice of good features is the key to 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 our base feature set include mel-frequency cepstral coefficients (MFCC), energy measures and their first- and second-derivatives.

Mel-Frequency Cepstral Coefficients (MFCCs) are obtained through nonlinear filterbank analysis motivated by human hearing mechanisms. They are popular in ASR. MFCC is used to characterize the acoustical shape of humming notes. For our analysis, 26 filterbank channels are chosen, and the first 12 MFCCs are selected as features.

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 are asked to hum using basic sounds that are a combination of a stop consonant and a vowel (e.g., "da", "la").

The 39-element feature vector contains 12 MFCCs, 1 energy measure, and their first and second derivatives. Experimental results will be shown in Sec. 4.

B. Hidden Markov Model

Hidden Markov models (HMM) with Gaussian mixture models (GMM) for observations corresponding to each state of an HMM are used to characterize a note model. Each note is 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 (see Sec. 3.2) 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 associated complexity may not justify the slight performance gain. Two distinctive note models are defined for "regular notes" and "rest".

C. Training Process

The parameters of HMMs are estimated during a supervised training process using a maximum likelihood approach with

Baum-Welch re-estimation. The first step in determining the parameters of an HMM is to make a rough guess about their values. Then, the Baum-Welch algorithm is applied to these initial values to improve their accuracy in the maximum likelihood sense.

D. Note Decoder

The Viterbi decoding algorithm is used in the decoding process. The recognition problem is to find a state sequence of a model which is most likely to have been generated by the data. The Viterbi decoding algorithm assumes that the maximum likelihood state sequence travels through the optimal path along each state.

3.3. Pitch tracking

After a note is segmented from a humming piece, it is passed to the second stage to decide its pitch. The pitch detector decides the pitch of a segmented note based on the statistical information of pitch models. The statistical information of pitch models is obtained from the humming database off-line. The detailed implementation of each component of the pitch detector is given below.

A. Pitch Analysis

Short-time autocorrelation is chosen for pitch analysis. The main advantage of using short-time autocorrelation is its relative low computational cost in comparison with other existing pitch detection algorithms. A frame size of 20 msec with 10 msec frame overlap was adopted throughout our experiments. The frame-based analysis is performed on a note segment, which usually has several frames. Multiple frames of a segmented note are used for pitch model analysis. After applying autocorrelation to those frames, pitch features are extracted. The selected pitch features include: the first, second, and third harmonics, the pitch median, and the pitch log standard deviation.

The first harmonic is also known as the fundamental frequency or the pitch. The first harmonic is the most important pitch information. However, speaker variability and noise may cause errors in the pitch tracking process. The second and third harmonics also play important roles in music. Therefore, the second and third harmonics are included to improve the robustness of pitch analysis. In fact, the second and third harmonics are shifted to the position of the first harmonic. In other words, the frequency values of H_2-H_1 and $H_3 - H_2$ are calculated, where H_n is the frequency value of the n^{th} harmonic.

Because of noise, some frames' pitch values are very different when compared with other frames' pitch values in the same note segment. Taking the average of them is not a good choice, since distant pitch values move the mean to the location where is away from the target value. The median pitch value of a note segment proves to be a better choice in experiments.

The outlying pitch values have also an impact on the standard deviation of a note segment. To overcome this problem, these pitch values should be moved back to the range where most pitch values belong. Since the smallest interval of two different notes is a semitone, we claim that pitch values away from the median value more than one semitone have a significant drift. The pitch values drifted more than a semitone are moved back to the median. Then, the standard deviation is calculated. Pitch values of notes are not linear in the frequency domain. In fact, they are linearly distributed in the log frequency domain. Therefore, calculating the standard deviation in the log scale makes more sense. The log pitch mean and the log standard deviation of a note segment are calculated.

B. Pitch Model

Pitch models are used for different pitch intervals, which are defined to be the difference in semitones of two adjacent notes:

$$\text{pitch interval} = \frac{\log(\text{current pitch}) - \log(\text{previous pitch})}{log \sqrt[12]{2}} \quad (1)$$

A pitch model has two attributes: the length of the interval (in terms of the number of semitones) and the pitch log standard deviation in the interval. The two attributes are modeled by the Gaussian mixture model. The boundary information and the ground truth of a pitch interval were obtained from manual transcription. The calculated pitch intervals and log standard deviations, which correspond to the ground truth pitch interval, are collected. A 2-D Gaussian mixture model is generated based on the collected information.

C. Pitch Detector

The pitch detector decides the pitch change of a segmented note with respect to its previous note. The first note of a humming piece is always marked as the reference note, and its detecting is in principle not needed. However, the first note's pitch is still calculated as reference in our experiments. The later notes of the humming piece are detected by the pitch detector. The pitch intervals and the pitch log standard deviations are calculated. They are used to select the best model that gives the maximum likelihood value as the detected result.

3.4. Transcription Generation

After the note segmentation stage and the pitch detection stage, a humming piece has all the information required for transcription. The transcription of the humming piece results in an sequence of length N with two attributes per symbol, where N is the number of notes. The two attributes are the duration and the pitch change (or pitch interval) of a note. The "Rest" note is labeled as "Rest" in the pitch interval attribute, since they do not have a pitch value.

4. EXPERIMENTAL RESULTS

The proposed algorithm consists of two stages: note segmentation and pitch detection. For the note segmentation stage, the 39-element feature vector consists of 12 MFCCs, 1 energy measure and their first and second derivatives. The frame size was chosen to be 20 msec, and the frame skip 3 msec (which means two consecutive frames have an overlap of 17 msec.). The detail of choosing the frame size and the frame rate was given in [2]. A 3-state left-to-right HMM with 2 GMMs was used to model notes, and each model was trained 10 times. For the pitch detection stage, the frame size was chosen to be 20 msec and the frame skip 10 msec. The number of Gaussian mixtures was set to one since the data set was quite limited.

A Graphic User Interface (GUI) program, called HTKEdit, was written based on the Hidden Markov Model Toolkit (HTK) [11]. The program can be used to train note models and pitch models. It has a pitch analysis tool for the pitch model study. It can also take the trained note model and the trained pitch model to perform both off-line and real-time transcription. The detail functionalities of HTKEdit and demo can be found at [12].

Let we define the following parameters:

- N: the no. of correct notes
- D: the no. of deletion errors
- S: the no. of substitution errors

- I: the no. of insertion errors

Two performance measures, *i.e.* the correct recognition rate (CRR) and the accuracy rate (AR), are adopted for comparison. They are defined as

$$\text{CRR} = (N - D - S)/N, \quad \text{AR} = (N - D - S - I)/N.$$

Both off-line and real-time recognition experiments were performed.

The off-line recognition was conducted with the leave-one-out method (trained on 7 speakers and tested on 1). Among seven speakers, the average AR was 75.63% and the average CRR was 88.13%. For real-time recognition, 8 speakers's humming data was used in training and humming pieces of two participants, who were not in the humming database, were tested in real time. Among two participants, the average CRR was 80.21% and the average AR was 72.32%.

The following observation is worth mentioning. At the note segmentation stage, the CRR value could be as high as 95%. However, the CRR and AR values were calculated based on the number of notes a test humming piece had (rather than the actual segmented units of notes). Consider a three-note humming piece that is decoded into three segmented units, where the first two comes from the first note and the last one for the second and third notes. For this case, the AR and CRR values after the note segmentation result is 100%. However, when the pitch detection process was applied to the wrongly segmented units, it is difficult to correct the error made in the earlier stage. Using the previous example, the results of the first two pitches will be from the first note, and the pitch of the last note will be obtained from the second and third notes. This is the major error encounted in our experiment, which lowers the AR and CRR performance significantly after the pitch detection stage.

The main reason of separating the duration and the pitch of a note into two stage recognition is that the duration variability of notes in music is greater than the phoneme duration variability in natural speech. On one hand, the separation of the two makes our recognition task relatively easier. On the other hand, it may lower the accuracy of the final results. A joint note segmentation and pitch detection process based on HMM is still under development to achieve a better performance.

5. CONCLUSION AND FUTURE WORK

A new statistical approach to speaker-independent humming recognition was proposed in this work. Features used in note modeling are extracted directly from the data. However, pitch values of hummed data are usually based on the previous note as a reference. Preliminary experimental results showed that our approach is a promising one for further refinement.

There are a couple of issues to be investigated as future extension. First, the role of inter-note context should be investigated through context-dependent modeling of melody and tempo contours. Error made by the note decoder can be corrected by a tempo's context model before passing a note to the pitch detector. A pitch's context model can further improve detected results at the pitch level. Second, a more comprehensive database of human humming from a larger set of human subjects should be gathered to enable detailed modeling and evaluation of the recognition performance. This is being performed right now. Third, it is worthwhile to consider a joint note segmentation and pitch detection algorithm. Finally, the music retrieval performance 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.

6. ACKNOWLEDGEMENTS

The work is supported in part by National Science Foundation (NSF ITR 53-4533-2720, EEC-9529152) and in part by ALi Microelectronics Corp.

7. 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] H.-H. Shih, S. S. Narayanan, and C.-C. J. Kuo, "An hmm-based approach to humming transcription," in *2002 IEEE International Conference on Multimedia and Expo (ICME2002)*, August 2002.

[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] R. J. McNab, L. A. Smith, and Jan H. Witten, "Signal processing for melody transcription," in *the 19th Australasian Computer Science Conference*, 1996.

[5] 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 *Digital Libraries Conference*, 1996.

[6] B. Gold and L. Rabiner, "Parallel processing techniques for estimating pitch periods of speech in the time domain," in *Journal of the Acoustical Society of America*, 1969, pp. 46:442 – 448.

[7] Christopher Raphael, "Automatic segmentation of acoustic musical signals using hidden markov models," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999, pp. 360–370.

[8] Christopher Raphael, "Automated rhythm tanscription," in *International Symposium on Music Information Retrieval (ISMIR 2001)*, 2001.

[9] Adriane Swalm Durey and Mark A. Clements, "Melody spotting using hidden markov models," in *International Symposium on Music Information Retrieval (ISMIR 2001)*, 2001, pp. 109–117.

[10] Adriane Swalm Durey and Mark A. Clements, "Features for spotting using hidden markov models," in *ICASSP 2002*, 2002.

[11] "Hidden Markov Model Toolkit," URL: http://htk.eng.cam.ac.uk/.

[12] "Hidden Markov Model Toolkit Editor: HTKEdit," URL: http://sail.usc.edu/music/.