VOICE-MELODY TRANSCRIPTION UNDER A SPEECH RECOGNITION FRAMEWORK

Dan-ning Jiang[*], *Michael Picheny*[**], *Yong Qin*[*]

[*]IBM China Research Lab
[**]IBM Watson Research Center
{jiangdn, qinyong}@cn.ibm.com, picheny@us.ibm.com

ABSTRACT

This paper presents a robust voice-melody transcription system using a speech recognition framework. While many previous voice-melody transcription systems have utilized non-statistical approaches, statistical recognition technology can potentially achieve more robust results. A cepstrum-based acoustic model is employed to avoid the hard-decisions that have to be made when using explicit voiced-unvoiced segmentation and pitch extraction, and a key-independent 4-gram language model is employed to capture prior probabilities of different melodic sequences. Evaluations are done from the perspective of both note recognition error rate and Query-by-Humming end-to-end performance. The results are compared with three other voice-melody transcription systems. Experiments have shown that our system is state-of-the-art: it is much more robust than other systems on data containing noise, and close to the best of all the systems on the clean data set.

Index Terms **—** voice-melody transcription, Query-by-Humming

1. INTRODUCTION

Voice-melody transcription refers to the extraction of the symbolic representation (the musical note sequence) from the acoustic signal corresponding to human singing. One of the most important applications of voice-melody transcription is as the front-end of a Query-by-Humming (QBH) system, which retrieves songs according to a query extracted from actual human singing. Many QBH systems first do voice-melody transcription and then use the symbolic representation to match melodies in the database. Thus, good transcription accuracy is crucial for the song retrieval performance. Another application is as an auxiliary means to create a MIDI sequence for electronic music when no MIDI input instrument is available.

Most prior work has focused on non-statistical approaches [1][2][3], utilizing variations on different pitch extraction algorithms and note segmentation schemes. The major problem associated with these approaches is the lack of robustness when inter-speaker and environmental signal distortions are encountered. Some attempts have also been made in utilizing statistical approaches. Shih et al. [4] proposed a two-stage statistical approach. Notes in the singing signal are first segmented by two HMM models trained with MFCC features, which represent "regular note" and "rest" (silence) respectively. The signal is assumed to only contain sounds which are a combination of a stop consonant and a vowel, like "da" or "ta". Then, the tone of each note is recognized with a GMM pitch model. Viitaniemi et al. [5] incorporated a HMM-based pitch trajectory model, a bi-gram musicological model, and a duration model in a simple probabilistic framework. The tempo of the singing signal is assumed to be known.

With the recent success in continuous speech recognition, it is natural to leverage this advanced technology in voice-melody transcription. This paper presents a robust voice-melody transcription system using a state-of-the-art speech recognition framework. In this system, the acoustic model of each semi-tone is trained with higher-order cepstral features instead of pitch, thus avoiding the hard-decisions that have to be made in explicit voiced-unvoiced segmentation and pitch extraction. A key-independent 4-gram language model is employed to capture prior probabilities of different melodic sequences. The transcription results are obtained via a single-pass global search process [6].

The paper is organized as follows. Section 2 describes our voice-melody transcription system and the training of acoustic model and language model. Section 3 presents a baseline Query-by-Humming back-end system, which will be used in the QBH end-to-end evaluation. Experimental results are given in Section 4.

2. THE MELODY TRANSCRIPTION SYSTEM

Given the acoustic observation sequence $X = X_1 X_2 \cdots X_n$, the goal of speech recognition is to find the word sequence $\hat{W} = w_1 w_2 \cdots w_m$ with the maximum posterior probability $P(W \mid X)$, as shown in the following formula:

$$\hat{W} = \arg\max_{w} P(W \mid X) = \arg\max_{w} \frac{P(W)P(X \mid W)}{P(X)} \qquad (1)$$

To use the formula in melody transcription, we first need to interpret W as the note sequence. Then, we train an acoustic model to allow us to compute $P(X \mid W)$ and a language model to compute $P(W)$ respectively. Finally, the transcription results are obtained through one global search process [6].

2.1. Acoustic model

2.1.1. Higher-order cepstral features

Nearly all other voice-melody transcription systems use pitch to represent the tonal information of notes. Although it is quite straight-forward, a hard-decision has to be made in explicit voiced-unvoiced segmentation and pitch extraction. Even by using the most advanced pitch extraction algorithms, voiced-unvoiced misclassification and octave errors always exist, and they have a very negative influence on the transcription results. The situation is

especially serious if the singing signal is mixed with noise. To improve the robustness of the acoustic model, we use higher-order cepstral features as a kind of mid-level representation of tonal information.

Figure 1 shows the cepstrum of a C4 (central C, MIDI note number is 60) frame (top) and that of an unvoiced frame (bottom). The low-frequency component in the cepstral domain is windowed out because it does not contain significant tonal information.

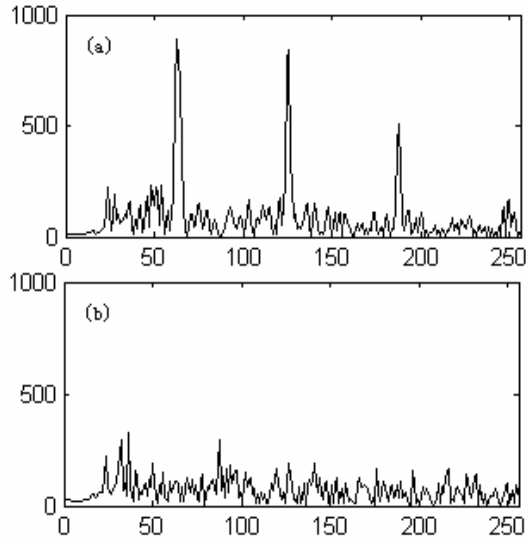

Figure 1. The raw cepstrum of (a) a C4 frame, (b) an unvoiced frame. The low-frequency component in the cepstral domain is windowed out. The sampling rate is 16 KHz.

For a voiced frame, the cepstrum has several dominant peaks while the cepstrum of an unvoiced frame is flat. The lag number in the cepstral domain has a simple relation with the frequency it represents:

$$lagN * f(lagN) = sampleRate \quad (2)$$

The raw cepstra are further processed prior to training the acoustic model. First, the segment relating to the possible pitch region is cut from the overall cepstrum. In our experiments, for male singers, the segment starts from $lagN = 48$ and ends at $lagN = 240$ (the sampling rate is 16 KHz), which corresponds to the range from C2 to E4 in the piano keyboard or MIDI note numbers 36-64; for female singers, the segment starts from $lagN = 24$ and ends at $lagN = 120$, which corresponds to the range from C3 to E5 in the piano keyboard or MIDI note numbers 48-76. Then, the cepstral segment is compressed to a fixed length (24 in our experiments) by averaging the cepstral feature values in each adjacent cepstral region determined by the compression ratio. Note that, male singers' cepstra are compressed twice as much as the female singers' cepstra, which means the tones contained in the male singers' signal are relatively increased by one octave.

2.1.2. Database preparation and training process

The database for acoustic model training contains three singers' data (one female and two males). Each singer was asked to sing with two styles: style 1 is sung with the syllable "da", and style 2 is sung with actual lyrics. The total amount of the style-1 data is about 45 minutes, and the amount of the style-2 data is about 1 hour. All the data were recorded in office environments at a 16 KHz sampling rate. An amateur pianist was asked to do the transcriptions for the data. Because of the cepstral feature processing described in Section 2.1.1, all notes in the male singers' transcriptions were increased by one octave.

Each semi-tone from E3 to D5, as well as silence, is modeled with a HMM. We jointly trained all the HMM models using the forward-backward algorithm. Each HMM model has 3 states and the probability density function of each state is modeled with a 4-component GMM model. The acoustic model is first trained with the 45 minutes' style-1 data, and then trained several iterations more with the 1 hour's style-2 data. However, we found that training the acoustic model with the style-2 data did not help the transcription results. Thus, in this paper, we only present results using the acoustic model trained with the style-1 data. As examples, figure 2 shows the mean vectors of all the three states of C4's HMM model (top) and C5's HMM model (bottom).

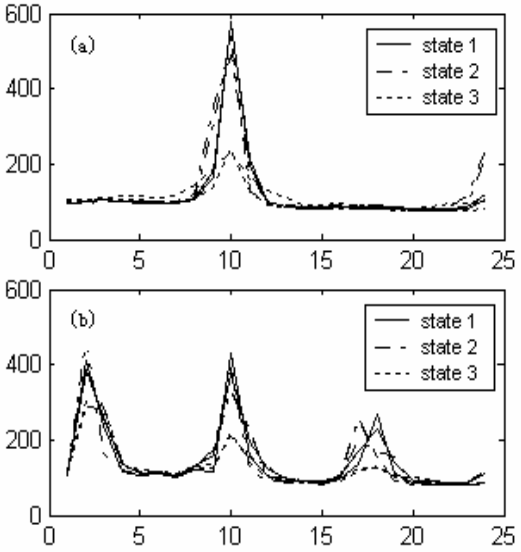

Figure 2. The mean vectors of all the three states of (a) C4's HMM, (b) C5's HMM.

2.2. Language model

In the Western musical key system, there are seven tones in each key scale, while one octave contains twelve semi-tones. It means that some semi-tones rarely appear in a song. This is why incorporation of a language model in the system can improve the transcription results.

We selected 522 melodies from the EsAC-database, a public available European folksong melody database[1], as the training data of the language model. The melodies are first normalized to the same key, C major. Then, there are two possible ways to build the language model. One is key-dependent and the other is key-independent. The former means building one language model for each major key respectively, and only the model which best matches the singing signal is used. It has the risk that when a wrong model is used, the transcription results can be significantly degraded. We also built the language model in a key-independent manner. The normalized melodies are shifted to the other 11 keys, and the original and all the shifted melodies form one big database

[1] The EsAC-database is available from http://www.cs.uu.nl/events/dech1999/dahlig/.

to train the n-gram model. In this case, the number of n should be relatively large to offer meaningful constrains. We set n as 4 in the system after some experimentation.

3. THE BASELINE QBH BACK-END

To evaluate the proposed voice-melody transcription system in an end-to-end fashion, a baseline Query-by-Humming back-end system was developed. It first represents the query and melodies in the database with the same symbolic representation, and then matches the query with each melody. Finally, a rank list is given as the query results.

In the baseline back-end, the melody feature is represented by a sequence of note intervals (i.e., the relative changes in tone between adjacent notes). Currently no rhythm features are used. The match procedure is divided into three stages: (1) a fast match, to locate the possible match points by querying n-gram indexes; (2) a rough match, to reduce the match candidates with a simplified string match algorithm; (3) a fine match, to give the final match score using a simple Dynamic Time Warping algorithm.

3.1. Fast match

In the fast match, all the melodies in the database are first broken into overlapping n-grams and a reverse index mapping the n-gram to a set of melodies is built. Then, a fuzzy approach is proposed because there are errors in the transcription of the query signal. The entire query transcription is expanded as a graph, as shown in figure 3. Nodes in the graph represent notes in the transcription, and arcs represent the note intervals. Besides the arcs corresponding to the exact note intervals between every two adjacent nodes, arcs which represent intervals one semi-tone larger or smaller are also added in the graph. All n-gram patterns in the graph are queried against the pre-built n-gram indexes. The fast match output is a set of vectors (P,k,Q), where P is the start point of the n-gram in the query graph, k and Q are the matched melody item number and start point of the n-gram in the melody item respectively.

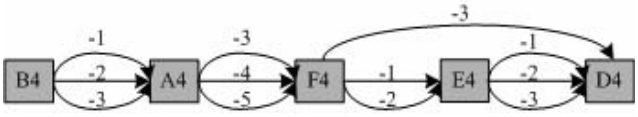

Figure 3. The query graph used in fast match.

Another problem is how to set the value of n. If n is larger, then the n-gram pattern is more discriminative, but at the same time the recall rate may decrease because of possible transcription errors. We used 4-grams (3 intervals computed from 4 notes) in the experiments as a good compromise using mutual information between n-gram patterns and melody identity on a database of 1325 melodies.

3.2. Rough match and fine match

String match algorithms are applied in the rough match and the fine match. In the rough match, for each match point (P,k,Q) returned by the fast match, a simplified string match algorithm is performed forwards and backwards starting from the original match point. These two scores are summed up as the rough match score. Here we describe the forward rough match process as an example. Suppose the query note interval sequence $a_1 a_2 \cdots a_m$ is matched with the melody note interval sequence $b_1 b_2 \cdots b_n$, and the current match point is $p=(i,j)$, where $1 \le i < m$ and $1 \le j \le n$. Then, the next match point in the search path is:

$$Next(p) = \arg\min \begin{cases} d(a_{i+1}, b_{j+1}) \\ d(a_i, b_{j+1}) + d(\phi, b_{j+1}) \\ d(a_{i+1}, b_j) + d(a_{i+1}, \phi) \end{cases} \quad (3)$$

In the formula, $d(\phi, b_{j+1})$ and $d(a_{i+1}, \phi)$ are the deletion cost and insertion cost respectively, which are set as constant; $d(a_{i+1}, b_{j+1})$ is the distance between the two note intervals. The match process starts from $p=(1,1)$, and terminates when $i=m$. It only calculates the match cost along one search path, which is more efficient than the DTW algorithms. The rough match scores of all the match points returned by the fast match are ranked and the top N matches are selected as the candidates for the fine match.

A DTW algorithm is applied in the fine match. It fills a cost matrix $D_{0\ldots m,0\ldots n}$. The initialization conditions are: $D_{0,0}=0$, $D_{0,j}=INF$ $(1 \le j \le n)$, and $D_{i,0}=INF$ $(1 \le i \le m)$, where INF is a very large number. For each $1 \le i \le m$ and $1 \le j \le n$, $D_{i,j}$ is computed as:

$$D_{i,j} = \min \begin{cases} D_{i-1,j-1} + d(a_i, b_j) \\ D_{i,j-1} + d(a_i, b_j) + d(\phi, b_j) \\ D_{i-1,j} + d(a_i, b_j) + d(a_i, \phi) \end{cases} \quad (4)$$

The fine match score is given as $\min_{1 \le j \le n} D_{m,j}$.

4. EXPERIMENTS

4.1. Experimental conditions

In our experiments, the evaluation data sets are part of the QBSH (Query by Singing/Humming) corpus [2] by the Multimedia Information Retrieval Lab at CS Department of National Tsinghua University, Taiwan. The whole corpus contains 2012 singing/humming clips from 85 common singers. The corpus was recorded in office environments, and each clip is 8s's long.

We selected two data sets from the corpus. One is clean, containing 140 clips from 8 singers (4 males and 4 females); the other is relatively noisy, containing 78 clips from 4 singers (3 males and 1 female). Figure 4 shows the spectrogram of a segment of noisy data. All the evaluation data were sung with lyrics. They were manually labeled according to what the singers actually sang, which sometimes is not exactly the same as the score of the target song. There is no overlap between singers in the evaluation data set and those in the training data set.

Besides our voice-melody transcription system (named the "M-Decoder"), three other systems are also evaluated and compared, which are represented as System-1, System-2, and System-3 respectively. System-2 and System-3 are successful commercial software for voice-melody transcription, while System-1 is also state-of-the-art and has been integrated into a QBH system.

[2] The corpus is available from http://neural.cs.nthu.edu.tw/jang2/dataSet/childSong4public/QBSH-corpus/.

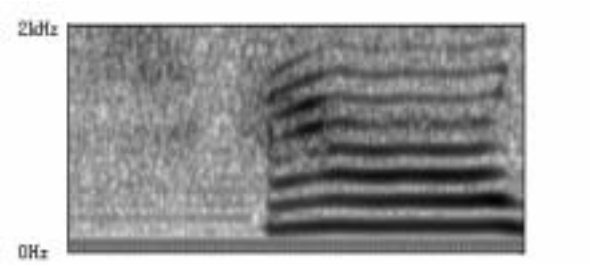

Figure 4. The spectrogram of a segment of noisy data.

4.2. Evaluation results

The melody transcription results are evaluated both in note recognition error rate and QBH end-to-end performance. To calculate the note recognition error rate, the automatic generated melody transcriptions are aligned with the hand-labeled transcriptions by an Edit Distance algorithm. The transcriptions are processed in two steps before the alignment: (1) Silence in the transcription is deleted; (2) Adjacent notes with the same tone are merged as one note.

Table 1 and table 2 list the note recognition error rates of all the four systems on the clean and noisy evaluation data set respectively. The measured errors are the sum of insertion and deletion errors, and frequency errors (the recognized note deviates from the hand-labeled note more than one semi-tone). The results show that on the clean data set, System-2 [7] gets the best results, while our system (M-Decoder) performs similarly. The other two systems are obviously worse. On the noisy data set, our system is much better than other three systems.

Table 1. The note recognition error rates on the clean data set.

Systems	del. + ins. err.	frequency err.	total err.
System-1	16.7%	4.5%	21.2%
System-2	8.0%	2.4%	10.5%
System-3	18.1%	5.2%	23.4%
M-Decoder	9.4%	2.4%	11.8%

Table 2. The note recognition error rates on the noisy data set.

Systems	del. + ins. err.	frequency err.	total err.
System-1	23.4%	7.0%	30.6%
System-2	20.7%	26.2%	47.0%
System-3	22.5%	8.6%	31.2%
M-Decoder	15.0%	4.2%	19.3%

The baseline QBH back-end system described in Section 3 is used in the end-to-end evaluation. The melody database contains 1325 melodies. Table 3 and table 4 list the retrieval results on the clean data set and noisy data set respectively. The results when hand-labeled transcriptions are used as queries are also given. Again, our system performs close to the best of all the systems (System 2) on the clean data set, while much better than all other systems on the noisy data set.

5. CONCLUSIONS

This paper proposed a robust voice-melody transcription system under a speech recognition framework. We evaluated the system both in note recognition error rate and QBH end-to-end performance. The results were compared with other three voice-melody transcription systems. Experiments showed that the proposed system is state-of-the art. It is much better than all other systems on the noisy data set, while still close to the best of all the systems on the clean data set. The melody transcription results can be further improved by collecting more training data and leverage more advanced acoustic model training methods, such as context-dependent models, LDA and MLLT training. Better melody match methods are also required to improve the song retrieval performance.

Table 3. The QBH end-to-end performance on the clean data set.

Systems	rank = 1	rank <= 5	rank <= 10
System-1	32.2%	54.0%	61.1%
System-2	54.6%	76.8%	82.3%
System-3	30.4%	46.8%	56.9%
M-Decoder	53.1%	75.5%	81.4%
Hand labels	75.8%	91.8%	93.9%

Table 4. The QBH end-to-end performance on the noisy data set.

Systems	rank = 1	rank <= 5	rank <= 10
System-1	24.1%	40.1%	43.0%
System-2	31.7%	38.0%	41.0%
System-3	28.6%	46.4%	51.4%
M-Decoder	38.7%	63.7%	71.2%
Hand labels	78.8%	94.7%	97.0%

6. ACKNOWLEDGEMENTS

We thank Dr. Hagen Soltau and Dr. Stanley Chen for their kind help on the IBM speech recognition tools; Dr. Jason Pelecanos and Dr. Zhong Su for many beneficial discussions. We also thank Dr. Ewa Dahlig and Prof. Roger Jang for the shared database.

7. REFERENCES

[1] G.Haus and E.Pollastri. "An Audio Front End for Query-by-Humming Systems," *Proc. of International Symposium of Music Information Retrieval,* 2001.

[2] P.M. Brossier, J.P. Bello, and M.D. Plumbley. "Fast Labeling of Notes in Music Signals," *Proc. of International Symposium of Music Information Retrieval,* 2004.

[3] T.De Mulder, J.P. Martens, M.Lesaffre, et al. "Recent Improvements of an Auditory Model based Front-end for the Transcription of Vocal Queries," *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing,* 2004.

[4] H Shih, S S. Narayanan, and C.J. Kuo. "Multidimensional Humming Transcription Using a Statistical Approach for Query by Humming Systems," *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing,* 2003.

[5] T. Viitaniemi, A. Klapuri, A. Eronen. "A Probabilistic Model for the Transcription of Single-Voice Melodies," *Proc. of the Finnish Signal Processing Symposium*, 2003.

[6] S. Chen, B. Kingsbury, L. Mangu, et al. "Advances in Speech Transcription at IBM under the DARPA EARS Program," to be published in *IEEE Trans. on Audio, Speech, and Language Processing*.

[7] P.Y. Rolland, G. Raskinis, and J.Ganascia. "Musical Content-based Retrieval: an Overview of the Melodiscov Approach and System," *Proc. of ACM Multimedia*, 1999.