An Automatic Singing Transcription System with Multilingual Singing Lyric Recognizer and Robust Melody Tracker

[1]Chong-kai Wang, [1]Ren-yuan Lyu, [2]Yuang-chin Chiang

[1]Department of Electrical Engineering, Chang Gung University, Taoyuan 333, Taiwan
[2]Institute of Statistics, National Tsing Hua University, Hsinchu 300, Taiwan
{ford,rylyu,chiang}@msp.csie.cgu.edu.tw

Abstract

A singing transcription system which transcribes human singing voice to musical notes is described in this paper. The fact that human singing rarely follows standard musical scale makes it a challenge to implement such a system. This system utilizes some new methods to deal with the issue of imprecise musical scale of input voice of a human singer, such as ***spectral standard deviation*** used for note segmentation, ***Adaptive Round Semitone*** used for melody tracking and ***Tune Map*** acting as a musical grammar constraint in melody tracking. Furthermore, a large vocabulary ***speech recognizer*** performing the lyric recognition tasks is also added, which is a new trial in a singing transcription system.

1. Introduction

Automatic musical transcription has been an interesting issue in multimedia applications. A system that is capable of transcribing orchestra input to musical notes is called a polyphonic transcription system. In contrast, a monophonic system takes as input single instrumental sound or human singing. Generally speaking, a transcription system for musical instrument performs quite well because the input instrumental sounds have contained accurate pitch information and thus are easily decoded. In the case of human singing, the task becomes more difficult due to the fact that a human singer can sing inconsistently and easily drifted away from standard musical scale. Unlike a musical professional, a common person might find it difficult to write down the melody in his mind or in his ears. He can choose to sing to a musical transcription system, but quite rarely he can sing as accurate as an instrument. This will deteriorate a transcription system which expects standard pitch scale input.

There are two basic problems in a human singing transcription system: note tracking and melody tracking. Note tracking is to segment singing waveform into voiced/unvoiced region, for voiced regions we mean a waveform segment where pitch period can be estimated. Each voiced region is often regarded as a musical note, and in music score, a syllable of a lyric word often mates one or many music notes. Melody tracking is to map singing pitches into musical notation correctly, even if a poor singer generates a bad melody. This paper reports our singing transcription system that, in addition to these two basic tasks, adds a lyric recognition function.

As depicted in <Fig.1>, the system includes four modules: Note Segmentation, Pitch Tracking, Melody Tracking and Lyric Recognition. The first three modules are cascaded to convert singing voice into musical note representation, and the Lyric Recognition module recognizes singing voice into text lyrics. A user just sings to a microphone and the transcription system converts the singing voice into a music score including notes and lyrics.

Each of the modules can be realized in a variety of ways. We briefly summarize here the methods used in the system that are different from the others. In Note Segmentation module, a new statistic called spectral standard deviation is used to segment the voice waveform into voiced and unvoiced regions. In Pitch Tracking module, we use a temporal pitch detection algorithm plus rule-based filtering to get the pitch contour. In Melody Tracking module, a statistic model called Adaptive Round Semitones (ARS) and a music grammar constraint called Tune Map is applied to improve the robustness of the system. Besides melody line recognition, a large vocabulary speech recognizer acts as the singing lyric recognition module. This, to our knowledge, is new to a singing transcription system.

This paper is organized as follows. Section 2 is a brief description for what we have used in the Note Segmentation and Pitch Tracking modules. Section 3 is for the newly proposed ARS algorithm and Tune Map model in detail. Section 4 explores the lyric recognition module that can handle multiple pronunciations in Mandarin and Taiwanese, 2 most important languages used in Taiwan. Section 5 is a conclusion.

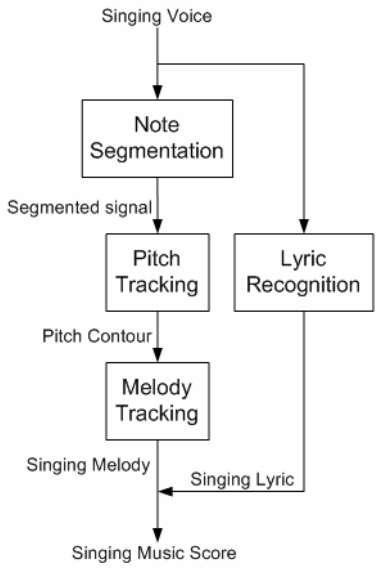

Figure 1. System overview

2. Note Segmentation and Pitch Tracking

There are many state-of-the-art techniques for the front-end voice processing. First of all, the singing voice signal should be end-point detected to determine the starting and ending

point of one utterance of singing. We use an energy-based algorithm to perform end-point detection which discriminates between silence and singing.

2.1. Note segmentation using spectral standard deviation

A human user will sing either with monosyllable sounds, like "du", "la", or with song lyrics as they wish. However, according to the authors' knowledge, present singing transcription systems put constraints on users to hum a melody by stop consonant as the initial part of each note such as "ta", "da", or constrain them to sing in a compulsory tempo. We can consider that all songs are designed in 2 types, one is "one note to one syllable" and the other is "many notes to one syllable". Take <Fig.2> as an example, the first three measures belong to the first type, and the last measure belongs to the second type. Also notice in <Fig.2> that the two notes with lyric "in - a" in the 2nd measure have the same pitch and are very short such that their pitch contours merge together and are difficult to differentiate. Therefore, the correctness of singing transcription depends on tracking all lyric syllables and picking the broken region in the pitch contour.

Tracking the lyric syllable is basically a voiced detection problem. There are many voiced/unvoiced discriminating algorithms in literatures, some of which use complicated features. Here, we propose a simple method for voiced/unvoiced classification using spectral standard deviation $V(n)$ for the n-th frame of a singing voice, which is defined as follows:

$$V(n) = \sqrt{\frac{\sum_{k=1}^{N/4} \left(S(k) - \bar{S}\right)^2}{N-1}}, \qquad (1)$$

where N is the FFT size, $S(k)$ is the k-th component of the spectrum., $\bar{S}$ is the mean of $S(k)$, and n is the frame index of segment of windowed singing voice. Notice that only a quarter of the spectral components are used. A threshold was then determined by experiments, and all local maximum and local minimum were marked. The intersection of $V(n)$ and threshold of $V(n)$ line implies the voiced/unvoiced boundary. The voiced/voiced boundary will be determined when neighboring local maximum and local minimum of $V(n)$ differ with each other too much. An example of Note Tracking of the song Jingle Bells was shown in <Fig.3>.

2.2. Pitch Tracking with post processing

After segmenting the voiced regions of waveform, it follows to extract the pitch or fundamental frequency (F_0) of human singing signal. Each segmented section will correspond to a musical note. There are many algorithms to extract pitch from audio signal, primarily developed for speech processing related researches. Here, we adopt autocorrelation function method. [1]

The extracted pitch of a singing signal is usually neither continuous nor smooth enough. Thus some heuristic smoothing techniques should be applied. Here, we use a 5-point median filter. The pitch value in a vowel region of a syllable is usually well defined and approximately a constant. The sum average of the pitch value within each smoothed voiced region was then calculated and waited to be processed in the Melody Tracking module.

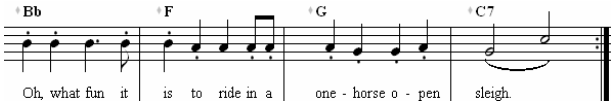

Figure 2. Score of Jingle Bells – a note vs. lyric mapping example.

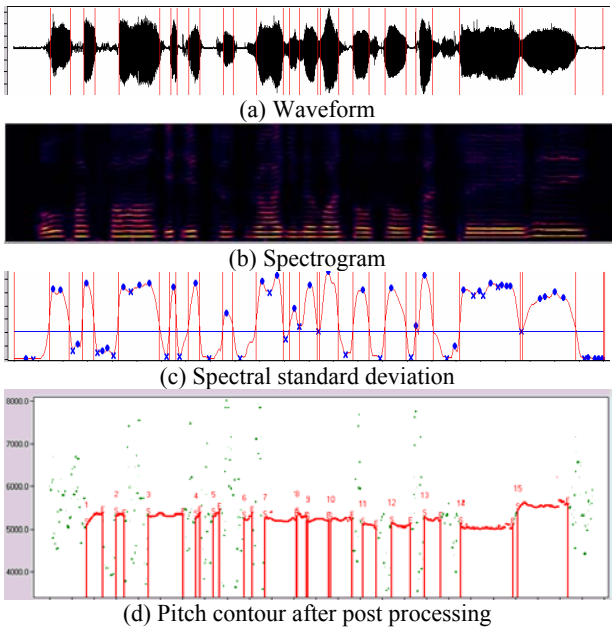

Figure 3. Note Tracking

3. Melody Tracking

The Melody Tracking module of the system deals with the average pitch value f (frequency in Hertz) for each note and transcribes them into music note representation, which have been defined as MIDI numbers for all possible semitones in the computer music processing domain. The pitch value f (in Hz) and the corresponding MIDI number N can be related as the following formula:

$$N = 69 + 12 \times \log_2 \frac{f}{440} \qquad (2)$$

For example, the note middle A with pitch being 440 *Hz* will be assigned a MIDI number 69, and the note A#, which is one semitone higher than middle A, is then assigned 70 as its MIDI number. On the equal tempered scale, the pitch interval between two consecutive MIDI numbers could be divided into 100 *cents*, e.g., the note middle A is 6900 cents. Thus, 100N is called pitch-in-cents.

3.1. Converting pitch to MIDI number

3.1.1. Round MIDI and modification

A straightforward approach is to convert pitch-in-cents to MIDI number by rounding the real value of pitch-in-cents/100 to the nearest integer. This is termed as *Round MIDI*.

However, human's pitch generation process is much more unstable than that of musical instruments. An untrained singer rarely generates voice with correct pitch value. It was said that only about 1 in 10000 people claim to have tone-absolute pitch perception/generation [2]. Furthermore, the pitch level

can drift upward or downward due to the singer's mood, and thus the pitch interval may not keep constant during singing. The experiments done in the other literatures show that the straightforward Round MIDI algorithm did not work well for melody tracking, thus several improvements have been proposed as follows.

- *McNab's Moving Tuning* [3] and *"Round Intervals"* rounds the difference of pitch in cents between adjacent voiced regions and treats them as the difference of MIDI number between adjacent notes.
- *Haus' Modified Round MIDI* [4] estimates an offset from all notes, adjusts all notes with this offset, and then does the same rounding as Round MIDI.

Here, we design a new method called Adaptive Round Semitones (ARS) which was shown to be the most effective among those in the literatures. [5][9]

3.1.2. Adaptive Round Semitones (ARS)

The Adaptive Round Semitones (ARS) was designed on the following three assumptions:

1. Most singers can generate or percept only tone-relative pitch.
2. The tune scale of human singing voice is not necessarily 100 cents per semitone.
3. The tune scale will probably change with time while singing and be dependent on previous notes.

For example, while a singer singing a rising melody sequence, he will inadvertently sing higher and higher. For a poor singer, the melody sung outward does not always match what he wishes to sing, although the listeners usually can understand what he sang. This is because he has a changeable and unusual tune scale. One important motive of designing ARS is to deal with these issues. This results in a robust melody tracker. The details of ARS were described in another paper of the authors. [5]

3.2. Musical grammar constraint

According to well tempered tuning system, an octave is equally divided into twelve notes; a scale is a series of notes selected among these twelve notes. Each of these notes is called a degree, and each degree has its own name designated by a Roman numeral. Most songs are composed conforming to an underline music grammar. Tone distribution of a song melody is not fully randomized. A well-composed song could be clustered in music theory by its tunes, such as C-Major, E-minor…etc. A Major scale is a structure of tones, namely "Do", "Re", "Mi", "Fa", "So", "La", "Si", "Do" (Degree name: Ⅰ, Ⅱ, Ⅲ, Ⅳ, Ⅴ, Ⅵ, Ⅶ, Ⅰ), where the differences in semitones of each continuing tones are 2,2,1,2,2,2,1. Such a tone structure could be looked upon as a grammar to put a constraint on the possibility of notes, which can be followed by another note.

For example, if the pitch differences in semitones of three notes N_1, N_2, N_3, were identified as 1 and 2, then these three notes will be one of the sequences "Mi-Fa-So" and "Si-Do-Re". If there comes a confusing note N_4 which is 1 or 2 semitones higher than N_3. We can easily determine that N_4 is 2 semitones higher according to the Major scale structure distribution.

Form music theory of Bach's well tempered scale, a semitone-level shifting of a sequence melody will not change its inner relation. The singer's singing tune is not a unique result from his/her singing key. Equal temperament made possible modern keyboard music with full modulation between all keys. The music grammar constraint is a method that adjusts a singer's tune and a song's tune in the semitone-level.

The idea that adds constraints of music grammar to help track the melody comes from the language modeling technique widely used in speech recognition research, where it improves the overall recognition performance significantly.

3.3. Tune Map

Music grammar constraints described above can be collected to a constraint table (Table.1). It shows the rules of music grammar according to music theory. Because that Major and Minor scale are cyclic shift in a similar manner, we can handle both Major and Minor scale songs using the same constraint table. In tone-relative pitch notion, we are interested in pitch difference of two neighboring notes. We construct a binary tree-like path map called Tune Map as shown in <Fig.4> which shows the moving path of notes. There are two kinds of nodes in Tune Map, one is "one-branch-node", and the other is "two-branch-node". Each node refers to a tone; the value on each note refers to the semitone offset between this tone and the first tone of the whole utterance, and the set besides the note refers to the probably degree name of this note.

We make a one-branch-node if the pitch interval rounding error of two neighboring notes does not exceeds ±20%. In the other case, we make it as a two-branch-node, and two of the branches will be connected to the node having tone upper and lower boundary of the rounding scale. The root node refers to the first note of the utterance, and then we add new branches and nodes for next nodes. While adding a new node, we should look up the Constraint Table to certificate the node. Certification process does logic AND for current probably degree name and the corresponding probably degree name of current tone offset. In the last stage, we can connect a tone path form the survival node to the root node. Finally, the melody of the singing utterance can be tracked.

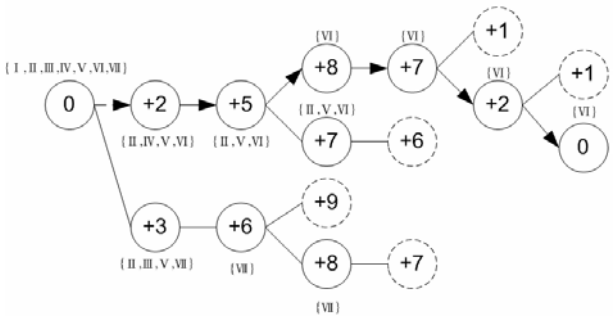

Figure 4. Tune Map

4. Lyric Recognition

The automatic singing transcription also includes a lyric recognition module, using speech recognition techniques.

Chou et. al. [6] reports that singing and speech signal can not easily be discriminated if MFCCs were used as feature. Thus, it is interesting to know how a speech recognizer performs if a singing signal is fed into the system. On the

other hand, it is also known [7] that if "connected-word" utterances are used as input into a continuous speech recognizer, the performance of the recognizer is degraded. This is due to the fact that the connected-word utterance does not match the training data of a continuous speech recognizer. How the discrepancy between the singing testing data and speech training data will affect a recognizer is interesting to know.

A second motivation of including a lyric transcription module concerns the task of musical retrieval. An approach of musical retrieval system is by humming, which is known as query-by-humming. A user hums a tune for a certain length of time, and the system extracts features from the tune, usually pitch related, and then applying pattern recognition techniques to search musical database for a match. [8] Since our system allows users to sing rather than hum, the input to the system contains extra lyric information. If we augmented a query system with results from both musical transcription and lyric recognition, one can expect better retrieval performance.

Our lyric recognition module is actually a multilingual large vocabulary speech recognizer, handling singing in both Taiwanese and Mandarin, the two most important languages used here in Taiwan. The lyrics of a song database in Taiwanese and in Mandarin are phonetically transcribed with their respective languages, with multiple pronunciations to handle the variation across dialects. Each lyric sentence corresponds to one or more phonetic sequences. Then a simple linear net is built from those phonetic sequences as the recognition network. Using the song database under this study we have 3207 lyric sentence and 19595 phonetic sequences. The latter is automatically generated with the help of our multilingual dictionary.

The recognizer uses right-content-dependent (RCD) phonemes as acoustic units with a total number 287. For each acoustic unit, we use a CHMM with 3 states, and variable mixtures per states. Mel-cepstrum coefficients and their derivatives are used as features. To train the acoustic models of the recognizer, we use a speech database from 619 people with 32.98 hours of speech totally, including Taiwanese and Mandarin.

With 925 testing singing sentences, the word recognition rate is 93.08%, syllable recognition rate is 95.60%. This is slightly better than that of our Taiwanese large vocabulary speech recognition system. Note that the recognition performance does not seem to be affected by the discrepancy between speech training data and singing testing data. The reason must be the imposition of the linear recognition network that plays as a strong language constraint.

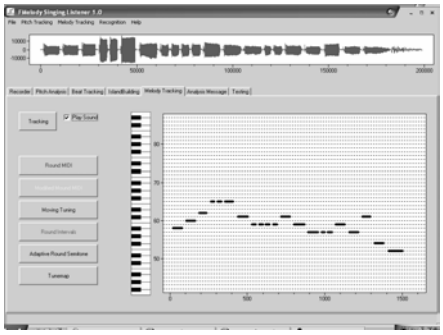

Figure 5. A snapshot of the transcription system

5. Conclusion

In our early reports [5][9], we have achieved the high performance in Melody Tracking module, which is more robust than those reported in the other literatures in melody tracking topic. We have built a demo system including our proposed full architecture in PC/Windows XP®. A snapshot is in <Fig.5>. The intergraded singing transcription system not only transcribes singing melody but also recognized singing lyric. Some stat-of-the-art techniques in speech analysis and signal processing have been applied into the system.

References

[1] M.M. Sondhi., "New method of pitch extraction," *IEEE Trans. Audio Electroacoust*, AU-16:262-266, 1968.

[2] Profita, J. and Bidder, T.G.,"Perfect Pitch," *American Journal of Medical Genetics*, 29, 763-771, 1988.

[3] R.J. McNab, L.A. Smith, and I. Witten, "Signal Processing for Melody Transcription," *Working Paper 95/22, Dept. of Computer Science*, University of Waikato, 1995.

[4] G. Haus, and E. Pollastri, "An audio front end for query-by-humming systems,"*International Symposium on Music Information Retrieval (ISMIR)*, 2001.

[5] Chong-kai Wang, Ren-yuan Lyu, and Yuang-chin Chiang, "A Robust Singing Tracker Using Adaptive Round Semitones (ARS)" (submitted)

[6] Wu Chou and Liang Gu, "Robust Singing Detection in Speech/Music Discriminator Design", *ICASSP* 2001

[7] X. Huang, M. Hwang, and L. Jiang. "Can continuous speech recognizers handle isolated speech?" *In Proceedings Eurospeech*, 1997.

[8] J.-S. Roger Jang, Jiang-Chun Chen, Ming-Yang Kao, "MIRACLE: A Music Information Retrieval System with Clustered Computing Engines", *2nd Annual International Symposium on Music Information Retrieval* 2001,

[9] Chong-kai Wang, Ren-yuan Lyu, and Yuang-chin Chiang," A singing transcription system using melody tracking algorithm based on Adaptive Round Semitones (ARS) plus music grammar constraints" *Stockholm Music Acoustic Conference (SMAC)*, 2003.

Table 1. Constraint Table

Tone offset	Probably Degree Name						
0	Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ
+1			Ⅲ				Ⅶ
+2	Ⅰ	Ⅱ		Ⅳ	Ⅴ	Ⅵ	
+3		Ⅱ	Ⅲ			Ⅵ	Ⅶ
+4	Ⅰ			Ⅳ	Ⅴ		
+5	Ⅰ	Ⅱ	Ⅲ		Ⅴ	Ⅵ	Ⅶ
+6				Ⅳ			Ⅶ
+7	Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	
+8			Ⅲ			Ⅵ	Ⅶ
+9	Ⅰ	Ⅱ		Ⅳ	Ⅴ		
+10		Ⅱ	Ⅲ		Ⅴ	Ⅵ	Ⅶ
+11	Ⅰ			Ⅳ			
+12	Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ