Transcription of the Singing Melody in Polyphonic Music

Matti Ryynänen and Anssi Klapuri
Institute of Signal Processing, Tampere University Of Technology
P.O.Box 553, FI-33101 Tampere, Finland
{matti.ryynanen, anssi.klapuri}@tut.fi

Abstract

This paper proposes a method for the automatic transcription of singing melodies in polyphonic music. The method is based on multiple-F0 estimation followed by acoustic and musicological modeling. The acoustic model consists of separate models for singing notes and for no-melody segments. The musicological model uses key estimation and note bigrams to determine the transition probabilities between notes. Viterbi decoding produces a sequence of notes and rests as a transcription of the singing melody. The performance of the method is evaluated using the RWC popular music database for which the recall rate was 63% and precision rate 46%. A significant improvement was achieved compared to a baseline method from MIREX05 evaluations.

Keywords: singing transcription, acoustic modeling, musicological modeling, key estimation, HMM

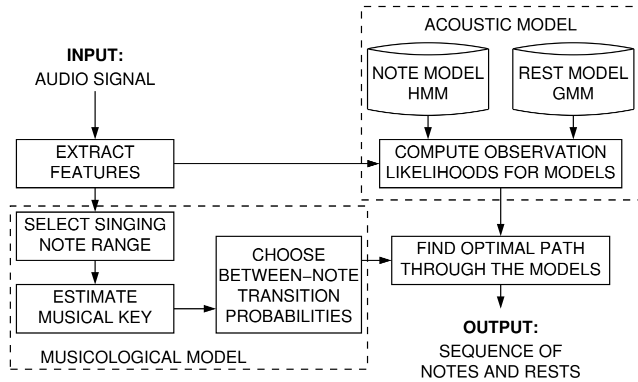

Figure 1. The block diagram of the transcription method.

1. Introduction

Singing melody transcription refers to the automatic extraction of a parametric representation (e.g., a MIDI file) of the singing performance within a polyphonic music excerpt. A melody is an organized sequence of consecutive notes and rests, where a note has a single pitch (a note name), a beginning (onset) time, and an ending (offset) time. Automatic transcription of singing melodies provides an important tool for MIR applications, since a compact MIDI file of a singing melody can be efficiently used to identify the song.

Recently, melody transcription has become an active research topic. The conventional approach is to estimate the fundamental frequency (F0) trajectory of the melody within polyphonic music, such as in [1], [2], [3], [4]. Another class of transcribers produce discrete notes as a representation of the melody [5], [6]. The proposed method belongs to the latter category.

The proposed method resembles our polyphonic music transcription method [7] but here it has been tailored for singing melody transcription and includes improvements, such as an acoustic model for rest segments in singing. As a baseline in our simulations, we use an early version of the proposed method which was evaluated second best in the Music Information Retrieval Evaluation eXchange 2005 (MIREX05) [1] audio-melody extraction contest. Ten state-of-the-art melody transcription methods were evaluated in this contest where the goal was to estimate the F0 trajectory of the melody within polyphonic music. Our related work includes monophonic singing transcription [8].

Figure 1 shows a block diagram of the proposed method. First, an audio signal is frame-wise processed with two feature extractors, including a multiple-F0 estimator and an accent estimator. The acoustic modeling uses these features to derive a hidden Markov model (HMM) for note events and a Gaussian mixture model (GMM) for singing rest segments. The musicological model uses the F0s to determine the note range of the melody, to estimate the musical key, and to choose between-note transition probabilities. A standard Viterbi decoding finds the optimal path through the models, thus producing the transcribed sequence of notes and rests. The decoding simultaneously resolves the note onsets, the note offsets, and the note pitch labels.

For training and testing our transcription system, we use the RWC (Real World Computing) Popular Music Database which consists of 100 acoustic recordings of typical pop songs [9]. For each recording, the database includes a reference MIDI file which contains a manual annotation of the singing-melody notes. The annotated melody notes are here referred to as the reference notes. Since there exist slight

[1] The evaluation results and extended abstracts are available at www.music-ir.org/evaluation/mirex-results/audio-melody

time deviations between the recordings and the reference notes, all the notes within one reference file are collectively time-scaled to synchronize them with the acoustic signal. The synchronization could be performed reliably for 96 of the songs and the first 60 seconds of each song are used. On the average, each song excerpt contains approximately 84 reference melody notes.

2. Feature Extraction

The front-end of the method consists of two frame-wise feature extractors: a multiple-F0 estimator and an accent estimator. The input for the extractors is a monophonic audio signal. For stereophonic input audio, the two channels are summed together and divided by two, prior to the feature extraction.

2.1. Multiple-F0 Estimation

We use the multiple-F0 estimator proposed in [10] in a fashion similar to [7]. The estimator applies an auditory model where an input signal is passed through a 70-channel bandpass filterbank and the subband signals are compressed, half-wave rectified, and lowpass filtered. STFTs are computed within the bands and the magnitude spectra are summed across channels to obtain a summary spectrum for subsequent processing. Periodicity analysis is then carried out by simulating a bank of comb filters in the frequency domain. F0s are estimated one at a time, the found sounds are canceled from the mixture, and the estimation is repeated for the residual.

We use the estimator to analyze audio signal in overlapping 92.9 ms frames with 23.2 ms interval between the beginnings of successive frames. As an output, the estimator produces four feature matrices X, S, Y, and D of size $6 \times t_{\max}$ ($t_{\max}$ is the number of analysis frames):

- F0 estimates in matrix X and their salience values in matrix S. For a F0 estimate $x_{it} = [X]_{it}$, the salience value $s_{it} = [S]_{it}$ roughly expresses how prominent x_{it} is in the analysis frame t.

- Onsetting F0 estimates in matrix Y and their onset strengths in matrix D. If a sound with F0 estimate $y_{it} = [Y]_{it}$ sets on in frame t, the onset strength value $d_{it} = [D]_{it}$ is high.

The F0 values in both X and Y are expressed as unrounded MIDI note numbers by $69 + 12\log_2(\text{F0}/440)$. Logarithm is taken from the elements of S and D to compress their dynamic range, and the values in these matrices are normalized over all elements to zero mean and unit variance.

2.2. Accent Signal for Note Onsets

The accent signal a_t indicates the degree of phenomenal accent in frame t, and it is here used to indicate the potential note onsets. There was room for improvement in the note-onset transcription of [7], and the task is even more challenging for singing voice. Therefore, we add the accent signal feature which has been successfully used in singing transcription [8]. The accent estimation method proposed in [11] is used to produce accent signals at four frequency channels. The bandwise accent signals are then summed across the channels to obtain the accent signal a_t which is decimated by factor 4 to match the frame rate of the multiple-F0 estimator. Again, logarithm is applied to the accent signal and the signal is normalized to zero mean and unit variance.

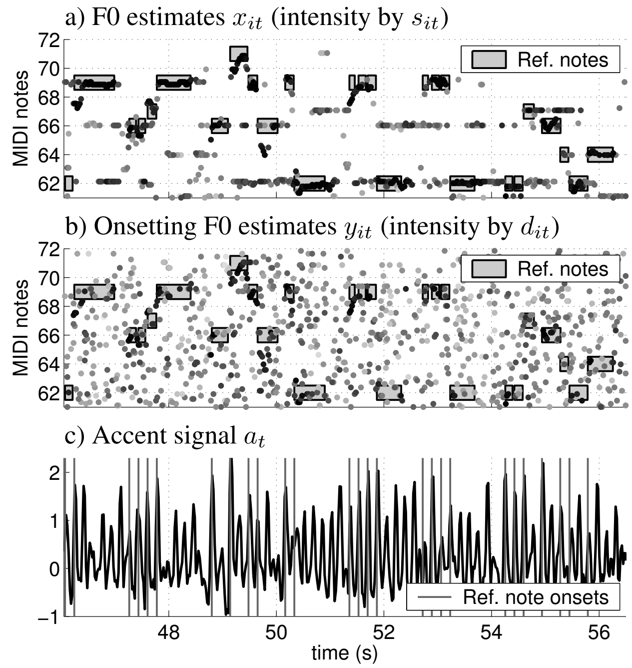

Figure 2. The features extracted from a segment of song RWC-MDB-P-2001 No. 14. See text for details.

Figure 2 shows an example of the features compared to reference notes. Panels a) and b) show the F0 estimates x_{it} and the onsetting F0s y_{it} with the reference notes, respectively. The gray level indicates the salience values s_{it} in panel a) and the onset strengths d_{it} in panel b). Panel c) shows the accent signal a_t and the note onsets in the reference melody.

3. Acoustic and Musicological Modeling

Our method uses two different abstraction levels to model singing melodies: low-level acoustic modeling and high-level musicological modeling. The acoustic modeling aims at capturing the acoustic content of singing whereas the musicological model employs information about typical melodic intervals. This approach is analogous to speech recognition systems where the acoustic model corresponds to a word model and the musicological model to a "language model", for example.

3.1. Note Event Model

Note events are modeled with a 3-state left-to-right HMM. The note HMM state $q_i, 1 \leq i \leq 3$, represents the typical values of the features in the i:th temporal segment of note events. The model allocates one note HMM for each MIDI note in the estimated note range (explained in Section 3.3). Given the extracted features X, S, Y, D, and a_t, the observation vector $\mathbf{o}_{n,t} \in \mathbb{R}^5$ is defined for a note HMM with nominal pitch n in frame t as

$$\mathbf{o}_{n,t} = (\Delta x_{n,t}, s_{jt}, \Delta y_{n,t}, d_{kt}, a_t) \ , \tag{1}$$

where

$$\Delta x_{n,t} = x_{jt} - n \ , \tag{2}$$
$$\Delta y_{n,t} = y_{kt} - n \ . \tag{3}$$

Index j is obtained using

$$m = \arg\max_i \{s_{it}\} \ , \tag{4}$$
$$j = \begin{cases} m \ , & \text{if } |x_{mt} - n| \leq 1 \\ \arg\min_i \{|x_{it} - n|\} \ , & \text{otherwise.} \end{cases} \tag{5}$$

Index k is chosen similarly to (4)–(5) by substituting k, y_{it}, and d_{it} in place of j, x_{it}, and s_{it}, respectively.

An observation vector thus consists of five features: (i) the F0 difference $\Delta x_{n,t}$ between the measured F0 and the nominal pitch n of the modeled note and (ii) its corresponding salience value s_{jt}; (iii) the onsetting F0 difference $\Delta y_{n,t}$ and (iv) its strength d_{kt}; and (v) the accent signal value a_t. For a note model n, the maximum-salience F0 estimate and its salience value are associated with the note if the absolute F0 difference is less or equal to one semitone (see (4)–(5)), otherwise the nearest F0 estimate is used. A similar selection is performed to choose index k for the onsetting F0s.

We use the F0 difference as a feature instead of the absolute F0 value so that only one set of note-HMM parameters needs to be trained. In other words, we have a distinct note HMM for each nominal pitch n but they all share the same trained parameters. This can be done since the observation vector (1) is tailored to be different for each note model n. Since the F0 difference varies a lot for singing voice, we use the maximum-salience F0 in contrast to the nearest F0 used in [7]. For the same reason, the onset strength values are slightly increased during singing notes, and therefore, we decided to use the onsetting F0s and their strengths similarly to normal F0 measurements.

The note model is trained as follows. For the time region of a reference note with nominal pitch n, the observation vectors (1) constitute a training sequence. Since for some reference notes there are no reliable F0 measurements available, the observation sequence is accepted for the training only if the median of the absolute F0 differences $|\Delta x_{n,t}|$ during the note is less than one semitone. The note HMM parameters are then obtained using the Baum-Welch algorithm. The observation likelihood distributions are modeled with a four-component GMM.

3.2. Rest Model

We use a GMM for modeling the time segments where no singing-melody notes are sounding, that is, rests. Rests are clearly defined for monophonic singing melodies, and therefore, we can now train an acoustic rest model instead of using artificial rest-state probabilities derived from note-model probabilities as in [7]. The observation vector $\mathbf{o}_{r,t}$ for rest consists of the maximum salience and onset strength in each frame t, i.e.,

$$\mathbf{o}_{r,t} = (\max_i\{s_{it}\}, \max_j\{d_{jt}\}) \ . \tag{6}$$

The model itself is a four-component GMM (analogous to a 1-state HMM) trained on the frames of the no-melody segments. The logarithmic observation likelihoods of the rest model are scaled to the same dynamic range with those of the note model by multiplying with an experimentally-found constant.

3.3. Note Range Estimation

The note range estimation aims at constraining the possible pitch range of the transcribed notes. Since singing melodies usually lie within narrow note ranges, the selection makes the system more robust against spurious too-high notes and the interference of prominent bass line notes. This also reduces the computational load due to the smaller amount of note models that need to be evaluated. If the note range estimation is disabled, we use a note range from MIDI note 44 to 84.

The proposed procedure takes the maximum-salience F0 estimate in each frame. If an estimate is on MIDI note range 50–74 and its salience value is above a threshold 1.0, the estimate is considered as valid. Then we calculate the salience-weighted mean of the valid F0s to obtain the note-range mean, i.e., $n_{\text{mean}} = \langle(\sum_i x_i s_i)/(\sum_i s_i)\rangle$, where operator $\langle\cdot\rangle$ is the nearest integer function, x_i is a valid F0 estimate, and s_i its salience. The note range is then set to be $n_{\text{mean}} \pm 12$, i.e., a two octave range centered around the mean.

In 95% of the songs, all reference notes are covered by the estimated ranges, and even in the worst case over 80% of notes are covered. Averaged over all songs, the estimated note ranges cover over 99.5% of the reference notes.

3.4. Key Estimation and Note Bigrams

The musicological model controls transitions between the note models and the rest model in a manner similar to that used in [7]. The musicological model is based on the fact that some note sequences are more common than others in a certain musical key. A musical key is roughly defined by the basic note scale used in a song. A major key and a minor key

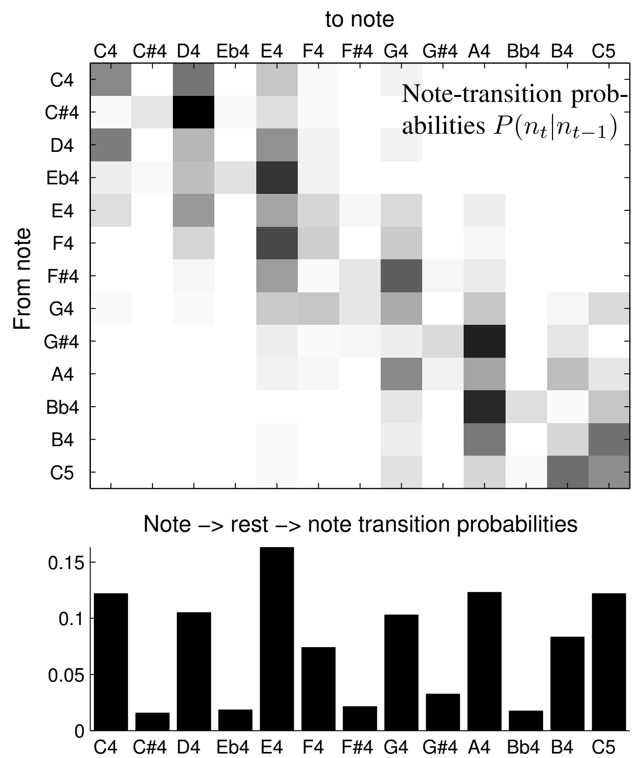

Figure 3. Musicological transition probabilities over one octave for the relative-key pair C major / A minor.

are called a relative-key pair if they consist of scales with the same notes (e.g., the C major and the A minor).

The musicological model first finds the most probable relative-key pair using a musical key estimation method [8]. The method produces likelihoods for different major and minor keys from those F0 estimates x_{it} (rounded to the nearest MIDI note numbers) for which salience value is larger than a fixed threshold (here we use zero). The most probable relative-key pair is estimated for the whole recording and this key pair is then used to choose transition probabilities between note models and the rest model. The current method assumes that the key is not changed during the music excerpt. In general, this is an unrealistic assumption, however, acceptable for short excerpts of popular music. Time-adaptive key estimation is left for future work.

The transition probabilities between note HMMs are defined by note bigrams which were estimated from a large database of monophonic melodies, as reported in [12]. As a result, given the previous note and the most probable relative-key pair r, the note bigram probability $P(n_t = j|n_{t-1} = i, r)$ gives a transition probability to move from note i to note j.

The musicological model assumes that it is more probable both to start and to end a note sequence with a note which is frequently occurring in the musical key. A rest-to-note transition corresponds to starting a note sequence and a note-to-rest transition corresponds to ending a note sequence. Krumhansl reported the occurrence probabilities of different notes with respect to the musical key, estimated from a large amount of classical music [13, p. 67]. The musicological model applies these distributions as probabilities for the note-to-rest and the rest-to-note transitions so that the most probable relative-key pair is taken into account. Figure 3 shows the musicological transition probabilities for between-note, note-to-rest, and rest-to-note transitions in the relative-key pair C major / A minor. If the musicological model is disabled, uniform distributions over all transitions are used.

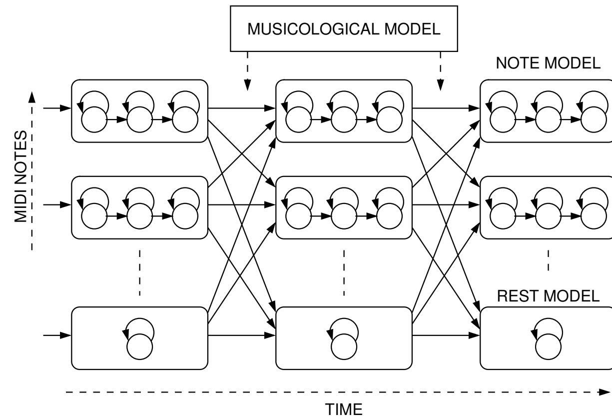

Figure 4. The network of note models and the rest model.

3.5. Finding the Optimal Path and Post-processing

The note event models and the rest model form a network of models where the note and rest transitions are controlled by the musicological model. This is illustrated in Figure 4. We use the Viterbi algorithm to find the optimal path through the network to produce a sequence of notes and rests, i.e., the transcribed melody. Notice that this simultaneously produces the note pitch labels, the note onsets, and the note offsets. A note sets on when the optimal path enters the first state of a model and sets off when the path exits the last state. The note pitch label is determined by the MIDI note number of the note model. Figure 5 shows an example transcription after finding the path.

As an optional post-processing step, we may use a simple rule-based glissando correction. The term glissando refers to a fundamental-frequency slide to the nominal note pitch. Glissando is usually employed at the beginning of long notes which often begin flat (too low) and the fundamental frequency is matched to the note pitch during the first 200 ms of a note [14, p. 203].

If a transcribed note shorter than 180 ms is immediately followed by a longer note with $+1$ or $+2$ interval between the notes, these two notes are merged as one which starts at

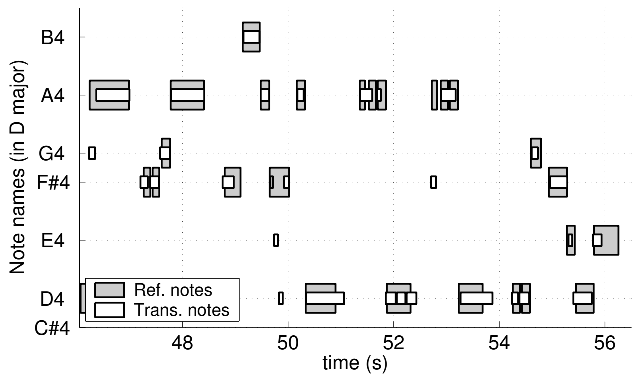

Figure 5. The transcription of the melody from song RWC-MDB-P-2001 No. 14. Figure 2 shows the features for this time segment.

Table 1. Simulation results summary (%).

Method	R	P	F	M
MIREX05 method (baseline)	56	28	37	51
Acoustic models (notes, rest)	60	**42**	48	54
+ Note-range estimation	61	43	49	54
+ Key estimation and note bigrams	**63**	45	51	53
+ Glissando correction	63	**48**	53	54

the first note onset, and has the MIDI note number and the offset of the latter note.

4. Simulation Results

The melody transcription method was evaluated using three-fold cross-validation on the 96 songs in RWC popular music database. We used the performance criteria proposed in [7], including the recall rate (R), the precision rate (P), and mean overlap ratio (M). The recall rate denotes how many of the reference notes were correctly transcribed and the precision rate how many of the transcribed notes are correct. A reference note is correctly transcribed by a note in the transcription if their MIDI note numbers are equal, the absolute difference between their onset times is less than 150 ms, and the transcribed note is not already associated with another reference note. The mean overlap ratio measures the average temporal overlap between transcribed and reference notes. In addition, we report the f-measure $F = 2RP/(R + P)$ to give an overall measure of performance.

The recall rate, the precision rate, the f-measure, and the mean overlap ratio are calculated separately for the transcriptions of each recording, and the average over all the transcriptions for each criterion are reported.

4.1. Transcription Results

Table 1 summarizes the melody transcription results for different simulation setups. As a baseline method, we use our melody-transcription method in the MIREX05 evaluations. The baseline method is a slight modification of the polyphonic music transcription method proposed in [7], and it uses multiple-F0 estimation (two F0s per frame), note event modeling, and note bigrams with key estimation.

Table 2. Results with perfect note range, perfect key, and worst case key (%).

Method	R	P	F	M
Perfect note range estimation	64	47	53	53
Perfect key estimation	63	45	51	53
Worst-case key estimation	37	29	32	57

The proposed transcription method reached recall rate 63%, precision rate 48%, f-measure 53%, and mean overlap ratio 54% when for the baseline method the corresponding results were 56%, 28%, 37%, and 51%. The rest model significantly improves the precision compared to the baseline method. By adding note-range estimation, the recall and precision rates are slightly increased. Using key estimation with note bigrams further improves both recall and precision rates. Finally, using simple post-processing to correct glissandi, precision rate is increased, since it reduces the number of incorrectly transcribed notes. The balance of recall and precision rates can be adjusted with the weighting of the rest model.

We studied the influence of imperfections in the note range estimation and in the key estimation to the overall performance of the method. The results are summarized in Table 2. We used the method with all the other components but the post processing (the results on the second last line in Table 1). By using this method but setting the note range limits according to the minimum and maximum of the reference notes, the recall and precision rates increase by one and two percentage units, respectively. However, no improvement is obtained from using manually annotated key signatures instead of the estimated keys (see key estimation results in Sec. 4.2). This suggests that small errors in key-estimation are not crucial to the overall performance. We also simulated the worst-case scenario of key estimation by converting every reference key into a worst-case key by shifting its tonic by a tritone (e.g., C major key is shifted to F♯ major). This dropped the recall and precision rates to 37% and 29%, respectively, thus indicating that the key estimation plays a major role in the method.

The perceived quality of the transcriptions is rather good. Due to the expressive nature of singing, the transcriptions include additional notes resulting from glissandi and vibrato. The additional notes sound rather natural although they are erroneous according to the evaluation criteria. Demonstrations of the singing melody transcriptions done with the proposed method are available at `http://www.cs.tut.fi/sgn/arg/matti/demos/melofrompoly`.

Table 3. Key estimation results.

Distance on the circle of fifths	0	1	2	3	≥ 3
% of songs	**76.6**	12.8	4.26	4.26	2.13

4.2. Key Estimation Results

We also evaluated the performance of the key estimation method. We manually annotated key signatures for 94 songs of the dataset (for two songs, the key was considered too ambiguous). As an evaluation criterion, we use the key signature distance on the circle of fifths between the reference key and the estimated relative-key pair. This distance is equal to the absolute difference in the number of accidentals (sharps ♯ and flats ♭). For example, if the reference key is A major and the key estimator correctly produces a relative-key pair A major / F♯ minor, the distance is zero (three sharps for both keys). If the reference key is E minor (one sharp) and the estimated relative-key pair is F major / D minor (one flat), the distance is two.

Table 3 shows the evaluation results for the key estimation method by using the introduced distance. The method correctly estimates the relative-key pair (distance zero) for over 76% of the songs. For approximately 90% of the songs, the key estimation method produces correct or a perfect fifth key (i.e., distance one).

5. Conclusions and Future Work

This paper described a method for the automatic transcription of singing melodies in polyphonic music. The method was evaluated with realistic popular music and showed a significant improvement in transcription accuracy compared to our previous method. This was mainly due to the acoustic modeling of no-melody (i.e., rest) segments.

There is still room for improvement. One possible approach to enhance the transcription accuracy would be to elaborate timbre information to discriminate singing notes from notes played with other instruments. We did some preliminary tests to include sound source separation in our transcription system. Briefly, we first generated a large set of note candidates by iteratively decoding several possible note paths. The note candidates covered approximately 80% of the reference notes. We then run a sound-source separation algorithm on these notes, calculate MFCCs on the separated notes, model the MFCCs of the correctly transcribed candidates with a GMM to derive a timbre model, and then run the Viterbi decoding again with the timbre model. Yet this approach did not perform any better than the proposed system in the preliminary simulations. However, we believe that using timbre in singing melody transcription from polyphonic music is worth further study and has the potential of improving the results in instrument specific transcription tasks.

References

[1] J. Eggink and G. J. Brown, "Extracting melody lines from complex audio," in *Proc. 5th International Conference on Music Information Retrieval*, Oct. 2004.

[2] M. Goto, "A real-time music-scene-description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication*, vol. 43, no. 4, pp. 311–329, 2004.

[3] M. Marolt, "Audio melody extraction based on timbral similarity of melodic fragments," in *Proc. EUROCON 2005*, Nov. 2005.

[4] K. Dressler, "Extraction of the melody pitch contour from polyphonic audio," in *Proc. 6th International Conference on Music Information Retrieval*, Sept. 2005. MIREX05 extended abstract, available online http://www.music-ir.org/evaluation/mirex-results/articles/melody/dressler.pdf.

[5] G. E. Poliner and D. P. W. Ellis, "A classification approach to melody transcription," in *Proc. 6th International Conference on Music Information Retrieval*, pp. 161–166, Sept. 2005.

[6] R. P. Paiva, T. Mendes, and A. Cardoso, "On the detection of melody notes in polyphonic audio," in *Proc. 6th International Conference on Music Information Retrieval*, pp. 175–182, Sept. 2005.

[7] M. P. Ryynänen and A. Klapuri, "Polyphonic music transcription using note event modeling," in *Proc. 2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 319–322, Oct. 2005.

[8] M. Ryynänen, "Singing transcription," in *Signal Processing Methods for Music Transcription* (A. Klapuri and M. Davy, eds.), pp. 361–390, Springer Science + Business Media LLC, 2006.

[9] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical, and jazz music databases," in *Proc. 3rd International Conference on Music Information Retrieval*, Oct. 2002.

[10] A. Klapuri, "A perceptually motivated multiple-F0 estimation method," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 291–294, Oct. 2005.

[11] A. P. Klapuri, A. J. Eronen, and J. T. Astola, "Analysis of the meter of acoustic musical signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, pp. 342–355, Jan. 2006.

[12] M. P. Ryynänen and A. Klapuri, "Modelling of note events for singing transcription," in *Proc. ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio*, Oct. 2004.

[13] C. Krumhansl, *Cognitive Foundations of Musical Pitch*. Oxford University Press, 1990.

[14] J. Sundberg, "The perception of singing," in *The Psychology of Music* (D. Deutsch, ed.), ch. 6, pp. 171–214, Academic Press, second ed., 1999.