Transcription of the Singing Melody in Polyphonic Music (MIREX 2006)

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

We introduce a method for the MIREX 2006 "Audio Melody Extraction" task in which the goal is to estimate fundamental frequency (F0) trajectory of the main melody within polyphonic music. The introduced method is based on multiple-F0 estimation followed by acoustic and musicological modeling. The acoustic model consists of separate models for melody 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 melody. The method details are published in ISMIR 2006 proceedings. As an extension to this method, we use a simple F0 estimate selection to produce the required F0 trajectory for the task evaluation. Although the method was developed for the automatic transcription of singing melodies in polyphonic music, it is also applicable in general melody transcription tasks.

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.

Recently, melody transcription has become an active research topic. The conventional approach is to estimate the 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 introduced method belongs to the latter category, and it is published in [7]. Here the method is, however, extended with a post-processing step of F0 selection so that the required output format is produced for the MIREX evaluation.

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

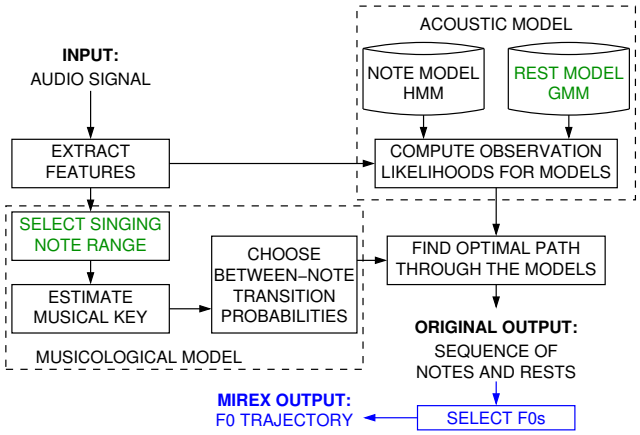

Figure 1. The block diagram of the transcription method. The blue section indicates an extension to the method for the MIREX style output. The green sections indicate the main changes to the system compared to our MIREX 2005 method.

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. The proposed method resembles our polyphonic music transcription method [8] and our MIREX 2005 method but now it has been tailored for singing melody transcription and includes improvements, such as an acoustic model for rest segments in singing and singing note range selection. These are indicated in green in Fig. 1.

As an extension to the system, we use a simple selection of F0s in the vicinity of each transcribed note to produce the required output. This is indicated with the blue section in Fig. 1.

2. Method Description

We briefly introduce the method in the following. For more details, please see [7].

2.1. Feature Extraction

The front-end of the method consists of two frame-wise feature extractors: a multiple-F0 estimator and an accent estimator. We use the multiple-F0 estimator proposed in [9] in a fashion similar to [8]. 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.

There was room for improvement in the note-onset transcription of [8], 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 [10]. We apply the accent estimation method proposed in [11].

2.2. Acoustic and Musicological Modeling

Our method uses two different abstraction levels to model 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.

2.2.1. Acoustic Models

Note events are modeled with a 3-state left-to-right HMM. The model allocates one note HMM for each MIDI note in the estimated note range (see Fig. 2). We use a GMM for modeling the time segments where no singing-melody notes are sounding, that is, rests. For training the note and rest models, we use the RWC (Real World Computing) Popular Music Database which consists of 100 acoustic recordings of typical pop songs with annotated melodies [12].

2.2.2. Musicological Modeling

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. The note range is determined from the estimated F0s.

The musicological model controls transitions between the note models and the rest model in a manner similar to that used in [8]. The musicological model first finds the most probable relative-key pair using a musical key estimation method [10]. The relative-key pair is then used to choose the note bigram probabilities estimated from a large database of monophonic melodies.

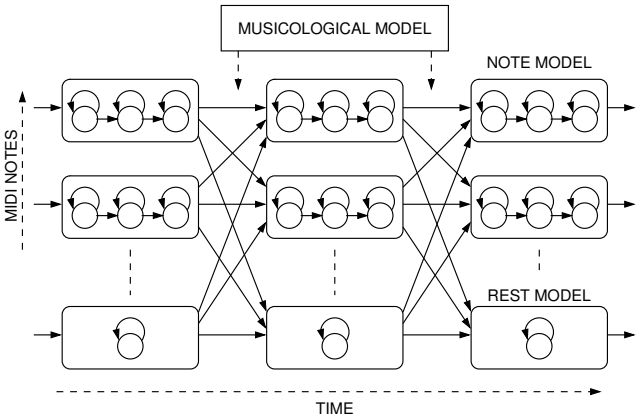

Figure 2. The network of note models and the rest model.

2.2.3. Finding the Optimal Path

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 2. 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.

2.3. Determining the F0 Trajectory for MIREX 2006

The remaining task is to determine F0s in every frame of a 10 ms grid. This is done based on the note-level transcription. Since we have the transcribed MIDI notes, we simply select the frame-wise F0s which were associated to the note during the transcription. If the absolute difference between the note model and an associated F0 is more than two semitones, we use the center frequency of the MIDI note instead. Then we use linear interpolation to output the F0s at every 10 ms. For the rest segments, we output the most prominent F0 estimates which lie on the estimated note range as the *unvoiced* F0 estimates (i.e., negative F0 values in the output), or zero if no such value is found.

Transcribed notes may end slightly too early due to salience decrease typical during note endings. If some unvoiced F0s immediately continue the F0 trajectory of the transcribed note, those unvoiced F0s are converted into voiced estimates. By starting from the end of a note, unvoiced F0 estimates are frame-by-frame converted to voiced if the absolute difference between consecutive estimates is less than 0.5

Figure 3 shows the method output for an excerpt in MIREX 2004 dataset. The green circles indicate the annotated F0s (in MIDI note numbers). The grey boxes shows the transcribed sequence of notes and rests. The actual output of the method is the voiced F0 estimates (the blue dots). In addition, the unvoiced F0 estimates (negative F0 values in

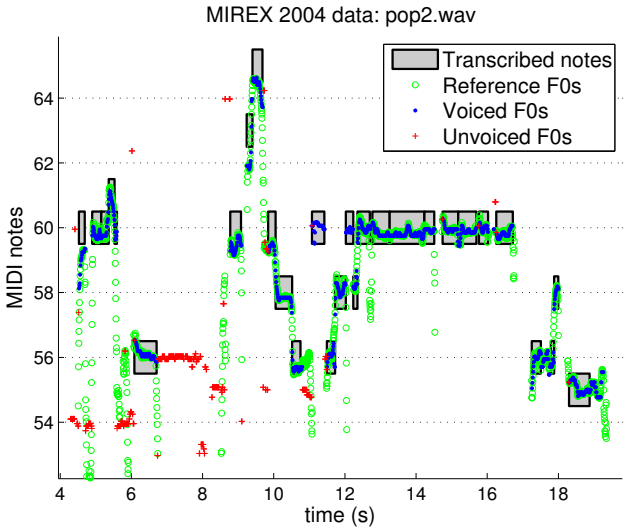

Figure 3. The method output for `pop2.wav` in MIREX 2004 dataset. See text for details.

the output file) are shown in red. The last note shows an example of note extension where some unvoiced F0s have been changed to voiced since they naturally continue the F0 trajectory of the note.

3. About the Implementation

The method has been implemented as Matlab M-files and MEX-files, and it should run in Linux Matlab versions 6.5 and 7.2. The execution time on a 1.7 GHz Linux PC is about twice the real-time without any particular optimizations.

4. Evaluation Results

The method performed second best in the evaluations. Since the method was developed for singing transcription, it performed better for vocal melodies than non-vocal melodies. Table 1 compares the performance of our methods in 2005 and 2006 for the "MIREX 2005 Dataset - All". The new method works better than the old one with these criteria. In particular, the rest modeling improves the voicing detection, clearly indicated by "Voicing d-prime" and "Vx False Alarm" rates. In addition, the 2006 method is considerably faster due to a faster multiple-F0 estimation method.

The method was developed for singing note transcription (not for singing F0-estimation), and the difference is more explicit with the criteria for discrete note events used in [7]. See Table 1 in [7] for comparison.

Table 1. Comparison of our methods from 2005 and 2006 for "MIREX 2005 Dataset - All".

Criterion	2005	2006
Overall Accuracy	64.3	67.9
Raw Pitch Accuracy	68.6	71.5
Raw Chroma Accuracy	74.1	75.0
Vx Recall	90.3	78.2
Vx False Alarm	39.5	16.5
Voicing d-prime	1.56	1.75
Runtime (s) (suggestive)	10970	773

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. Ryynänen and A. Klapuri, "Transcription of the singing melody in polyphonic music," in *Proc. 7th International Conference on Music Information Retrieval*, (Victoria, Canada), Oct. 2006.

[8] 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.

[9] 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.

[10] 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.

[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. 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.