# APPLICATIONS OF A SEMI-AUTOMATIC MELODY EXTRACTION INTERFACE FOR INDIAN MUSIC

Vishweshwara Rao, Sachin Pant, Madhumita Bhaskar and Preeti Rao
Department of Electrical Engineering, IIT Bombay
{vishu, sachinp, madhu88, prao}@ee.iitb.ac.in

**Abstract**

*Automatic extraction of the melody from polyphonic music recordings is a challenging problem for which no general solutions currently exist. We present a novel interface for semi-automatic melody extraction with the goal to provide highly accurate pitch tracks of the lead voice with minimal user intervention. Audio-visual feedback facilitates the validation of the obtained melodic contour, and user control of the analysis parameters enables direct and effective control over the voice pitch detection module by an intelligent user. This paper describes the interface and also an application of the interface for note-level representation of the vocal melody extracted from polyphonic audio in a query-by-humming system. In such a system, the continuous melodic contour needs to be further processed to obtain a suitable note sequence-like representation for more efficient search.*

## 1. Introduction

Melody extraction from polyphony finds application as a front end in several music information retrieval (MIR) based applications, such as query-by-humming, cover song identification and audio-to-score alignment, as well as musicology and pedagogy. A rough definition of the melody of a song is the monophonic pitch sequence that a listener might reproduce if asked to hum a segment of polyphonic music [1]. Polyphony indicates that more than one musical sound source may be simultaneously present. The melodic pitch sequence is usually manifested as the fundamental frequency (F0) contour of the lead musical instrument in the polyphonic mixture (here considered to be the singing voice). Although there exists a considerable body of work in pitch extraction from monophonic (single-source) audio, advances in research that enable melodic pitch extraction from polyphonic audio (a harder problem because of increased signal complexity due to polyphony) have only recently been made (in the last decade).

This paper describes a graphical user interface (GUI) for semi-automatic melody extraction from polyphonic music along with an application of such an interface in a query-by-humming system. Melody based reference templates required for the searchable database in query-by-humming systems must be extracted from polyphonic soundtracks. The final objective in the design of the user interface is to facilitate the extraction and validation of the voice pitch from polyphonic music with minimal human intervention. Since the manual marking of vocal segment (sung phrase) boundaries is much easier than automatic detection of the frame-by-frame voice pitch, the focus on the design of the back-end melody extraction program [2] has been on automatic and high accuracy vocal pitch extraction.

The next section describes the design layout and operation of our melody extraction interface. Section 3 brings out the salient features of interface with respect to facilitating melody extraction and refinement. Section 4 describes the application of the interface in a query-by-humming system specifically focusing on an algorithm for stylization i.e. obtaining a note-level representation from the continuous melodic contour.

## 2. Interface: Description and Operation

The basic features that are expected from any interface intended for audio analysis/editing comprise of waveform and spectrogram displays, selection and zooming features, and audio playback. In this section we describe the layout of the interface that, in addition to these basic features, also has features designed for the melody extraction task. The operation of the interface is also described.

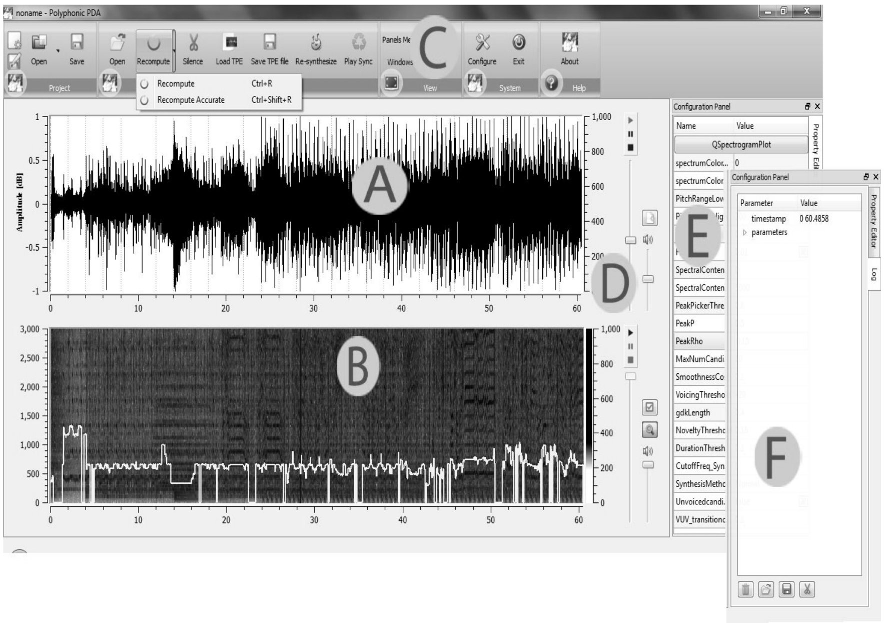

Fig. 1. Snapshot of the melody extraction interface A: Waveform viewer, B: Spectrogram and pitch contour viewer, C: Menu bar, D: Scrolling and playback control, E: Parameter window and F: Log viewer.

## 2.1. Description

A snapshot of the interface is provided in Fig. 1. It consists of a waveform viewer (A), a spectrogram viewer (B), a menu bar (C), controls for audio viewing, scrolling and playback (D), a parameter window (E), a log viewer (F). The waveform and spectrogram of the audio are displayed in A & B respectively. The horizontal axis corresponds to the timeline, with data moving from left to right. The right vertical axis in B corresponds to F0 frequency. The vertical bar at the center is the "present time". Controls (D) are provided for playing back the audio, controlling the volume, and the progress of the audio. Moving markers provide timing information. The menu bar (C) and the parameter window (E) control the use of the melody extractor. A log viewer (F) is provided to save and load the analysis parameters for the different segments.

## 2.2 Audio Analysis

The audio example to be analyzed, in .wav format, is loaded into the interface. The waveform and spectrogram of the music are automatically displayed. The melody extractor is invoked by pressing the ↻ button in the menu. By default, the melody extractor is called in single-F0 tracking mode, which is found to perform quite accurately on most audio examples. Alternatively the user may also select to use a more accurate, but slower, melody extraction algorithm (See Section 3.5.) by checking the dual-F0 option in the drop-down menu under the ↻ button. This function is especially useful when an accompanying pitched instrument is of comparable, or greater, loudness than the voice. The resulting pitch contour is displayed as a bright curve in B. The estimated F0 contour is plotted on top of the spectrogram, which helps visually validate the estimated melody by observing the shape and/or the extent of overlap between the pitch contour and any of the voice harmonics. Voice harmonics are typically characterized by their jittery/unsteady nature. Audio feedback is also provided by pressing the ► button on the right of the spectrogram. This plays back a vowel re-synthesis of the estimated F0 contour of the selected audio segment. The extracted pitch contour of the entire audio clip can be synthesized using the ⚙ button from the menu (C).

### 2.3. Saving and Loading Sessions

The interface provides an option for saving the final melody and the parameters used for different selected regions by using the 💾 button. A user can save the pitch extracted in a specific file format (TPE), which has three columns containing the Time stamp (in sec), the Pitch (in Hz), and the frame-level signal Energy. This amounts to saving a session. This TPE file can be reloaded later. Also, the parameters of the melody extractor used during the analysis can be saved in an XML file.

## 3. Interface: Salient Features

In the design of the interface we have attempted to incorporate several features that increase its functionality. The salient features of our melody extraction interface are described below.

### 3.1. Novel Melody Extractor

The melody extraction back-end system used by our interface has been extensively evaluated on polyphonic vocal music and has demonstrated very accurate voice pitch extraction performance [2]. We found that the design considerations we made also resulted in performance on par with state-of-the-art systems when evaluated at the Audio Melody Extraction Task at MIREX 2008 & 2009. The system utilizes a spectral harmonic-matching pitch detection algorithm (PDA) followed by a computationally-efficient, optimal-path finding technique that tracks the melody within musically-related melodic smoothness constraints. An independent vocal segment detection system then identifies audio segments in which the melodic line is active/silent by the use of a melodic pitch-based energy feature.

Further our melody extraction system uses non-training-based algorithmic modules i.e. is completely parametric. The performance of systems, which incorporate pattern classification or machine learning techniques [3], [4], is highly dependent on the diversity and characteristics of the training data available. In polyphonic music the range of accompanying instruments and playing (particularly singing) styles across genres are far too varied for such techniques to be generally applicable. When using our interface, users with a little experience and training can easily develop an intuitive feel for parameter selections that result in accurate voice-pitch contours.

### 3.2. Validation

The user can validate the extracted melodic contour by a combination of audio (vowel re-synthesis of extracted pitch) and visual (spectrogram) feedback. We have found that by-and-large the audio feedback is sufficient for melody validation except in the case of rapid pitch modulations, where matching the extracted pitch trajectory with that of a clearly visible harmonic in the spectrogram serves as a more reliable validation mechanism.

Currently there are two options for the vowel-energy used in synthesis. The frame-level signal energy may be used but we have found that this leads to audible bursts especially if the audio has a lot of percussion. Alternatively we have also provided a constant-energy synthesis option which allows the user to focus on purely the pitch content of the synthesis without distractions from sudden changes in energy. This option can be selected from the parameter list (E).

An additional feature that comes in handy during melodic contour validation is the simultaneous, time-synchronized playback of the original recording and the synthesized output. This can be initiated by clicking the ♻ button on the menu (C). A separate volume control is provided for the original audio and synthesized playback. By controlling these volumes separately, we found that users were able to make better judgments on the accuracy of the extracted voice-pitch.

### 3.3. Segment-level Parameter Selection

The analysis parameters that influence the performance of our melody extraction system are the F0 search range, frame-length, lower-octave bias and melodic smoothness tolerance. An intelligent user will be able to select or tune these parameter settings, based on observed signal characteristics, to obtain a correct output melody. For example, in the case of male singers, who usually have lower pitch than females, lowering the F0 search range and increasing the frame-length and lower-octave bias results in an accurate output. In the case of large and rapid pitch modulations, increasing the melodic smoothness tolerance is advisable.

It may sometimes be possible to get accurate voice-pitch contours by using a fixed-set of analysis parameters for the whole audio file. But many cases were observed, especially of male-female duet songs and excerpts containing variations in rates of pitch modulation, where the same parameter settings did not result in an accurate pitch contour for the whole file. In order to alleviate such a problem the interface allows different parameters to be used for different segments of audio. This allows for easy manipulation of parameters to obtain a more accurate F0 contour. The parameter window (E) provides a facility to vary the parameters used during analysis.

### 3.4. Non-Vocal Labeling

Even after processing, there may be regions in the audio which do not contain any vocal segments but for which melody has been computed. This occurs when an accompanying, pitched instrument has comparable strength as the voice because the vocal segment detection algorithm is not very robust to such accompaniment. In order to correct such errors we have provided a user-friendly method to zero-out the pitch contour in a non-vocal segment by using the ✂ tool from the menu (C).

### 3.5. Error Correction by Selective Use of Dual-F0 Back-end

State-of-the-art melody extraction algorithms have been known to incorrectly detect the pitches of loud, pitched accompanying instruments as the final melody, in spite of the voice being simultaneously present. Recently, however, we have shown that attempting to track two, instead of a single, pitch contour can result in a significant improvement in system performance [5]. However the use of this type of approach results in a considerable increase in computation time and may not be practically viable for long audio segments. However, we have provided the option for the user to selectively apply such an analysis approach i.e. track 2 F0s. On selecting this option (by selecting the dual-F0 option in the drop-down menu under the ↻ button) the system will output 2 possible, melodic contours. This is much cleaner than presenting the user with multiple locally-salient F0 candidates, as this will clutter up the visual display. The user can listen to the re-synthesis of each of these contours and select any one of them as the final melody. Typically we expect users to use this option on sung segments for which the single-F0 melody extractor always outputs some instrument pitch contour despite trying various parameter settings.

## 4. Application in a Query-by-Humming System

Melody based retrieval by “query-by-humming” (QBH) is a prominent example of MIR where a sung query (representing the tune of the desired song) is matched with melody lines extracted from song soundtracks that make up the searchable database [6]. Note-level representations (i.e. sequence of note pitch values and durations) of the database melodies provide for efficient searching when the user query is also represented similarly. A note sequence is obtained by the temporal segmentation of the melodic pitch contour into note regions and the subsequent labeling of each region with a single pitch value. Deriving a note sequence from a user query is relatively simple due to the monophonic syllabic singing constraint which makes feasible the automatic detection of note boundaries and pitch tracks [6]. The database note representations however must usually be extracted from polyphonic audio soundtracks. The semi-automatic melody extraction interface facilitates the accurate detection of the melodic pitch contour throughout the vocal segments of the audio. The syllable boundaries are indicative of note boundaries, and can be marked manually using the spectrogram and audio playback features of the interface. The subsequent step of labeling the notes with a single pitch value each, given the note boundaries and the continuously varying pitch within the note region, is a non-trivial problem for both the database and query songs. In this section, we propose some methods to address this pitch stylization problem with an evaluation of performance in the context of the QBH application.

### 4.1 Note segment labeling

Our goal is to label each note segment with a single pitch value that best represents the continuous pitch contour within the detected note boundaries. The “best” representation could be defined as the one resulting in a stylized pitch contour that is perceptually most similar to the actual pitch contour. In the present application however it may be more appropriate to consider the similarity in terms of the melodic distance metric used in QBH. We therefore measure the performance of the pitch stylization method in terms of the distance between the pitch labels of corresponding notes of reference and user query note sequences for the same melodic phrase.

The pitch variation between note boundaries can arise from the approach and exit regions around the steady state of the note or from intended ornamentation during the intonation of the note. Note labeling involves the computation of a single representative pitch value from the pitch contour. Some simple and intuitively appealing ways of doing this are picking the (i) maximum value of pitch attained within the note segment, and (ii) average of the pitch values in the middle one-third segment of the note. In each case, the stylized pitch contour is converted to the cents scale based on normalization by the pitch label of the first note of the sequence. Next the value of each note was rounded to the nearest 100 cents to approximate the equal temperament scale. The two approaches to labeling are compared experimentally as described next.

### 4.2 Evaluation of performance

The semi-automatic melody extraction interface was used to obtain the segmented pitch contour of 4 distinct phrases from 4 popular Indian movie songs providing a total of 70 reference notes. Three musically trained singers hummed the melody of each of the song phrases giving 210 user query notes for the computation of pitch error statistics of user note with respect to the corresponding reference note. Fig 2 shows the pitch error histograms (i.e. reference note label minus user note label) for each of the labeling methods. We see that both methods yield errors within 50 cents and few very large errors. The first method (i.e. maximum pitch value) has a more concentrated distribution compared with that from the average pitch method. This seems to indicate that the maximum attained pitch is more invariant across different singers rendering the same note rather than the average pitch. However this needs further investigation with a larger dataset, and also the separate consideration of steady and gliding or other ornamented notes.

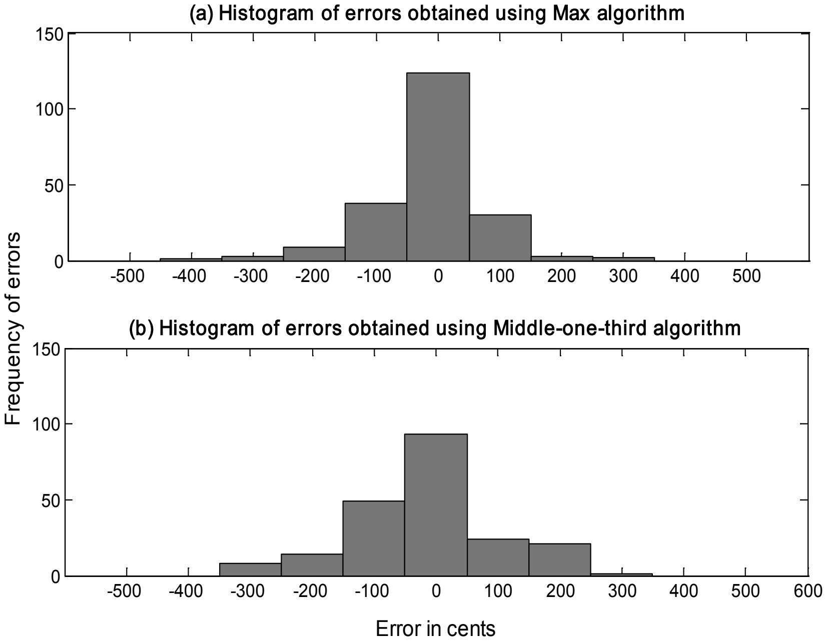

Fig.2. Histograms showing the errors for all 210 notes spread between -600 and +600 cents, after applying (a) Max algorithm (b) Middle-one-third algorithm

## References

[1] G. Poliner, et. al., "Melody transcription from music audio: Approaches and evaluation," *IEEE Trans. Audio, Speech, Lang., Process.*, vol. 15, no. 4, pp. 1247–1256, May 2007.

[2] V. Rao and P. Rao, "Melody extraction using harmonic matching," in *MIREX Audio Melody Extraction Contest Abstracts*, Philadelphia, 2008.

[3] G. Poliner and D. Ellis, "A classification approach to melody transcription," in *Proc. Intl. Conf. Music Information Retrieval,* London, 2005.

[4] H. Fujihara et. al. "F0 estimation method for singing voice in polyphonic audio signal based on statistical vocal model and Viterbi search," in *Proc. IEEE Intl. Conf. Audio Speech and Sig. Process.*, Toulouse, France, 2006.

[5] V. Rao, and P. Rao, "Improving polyphonic melody extraction by dynamic programming based multiple F0 tracking," in *Proc. 12th Intl. Conf. Digital Audio Effects (DAFx-09),* Como, Italy, Sept. 2009.

[6] M. Raju, B. Sundaram and P. Rao, "TANSEN: A query-by-humming based music retrieval system," in *Proc. National Conf. on Communications*, Chennai, 2003.