SiPTH: Singing Transcription Based on Hysteresis Defined on the Pitch-Time Curve

Emilio Molina, Lorenzo J. Tardón, Ana M. Barbancho, and Isabel Barbancho, *Senior Member, IEEE*

***Abstract*—In this paper, we present a method for monophonic singing transcription based on hysteresis defined on the pitch-time curve. This method is designed to perform note segmentation even when the pitch evolution during the same note behaves unstably, as in the case of untrained singers. The selected approach estimates the regions in which the chroma is stable, these regions are classified as voiced or unvoiced according to a decision tree classifier using two descriptors based on aperiodicity and power. Then, a note segmentation stage based on pitch intervals of the sung signal is carried out. To this end, a dynamic averaging of the pitch curve is performed after the beginning of a note is detected in order to roughly estimate the pitch. Deviations of the actual pitch curve with respect to this average are measured to determine the next note change according to a hysteresis process defined on the pitch-time curve. Finally, each note is labeled using three single values: rounded pitch (to semitones), duration and volume. Also, a complete evaluation methodology that includes the definition of different relevant types of errors, measures and a method for the computation of the evaluation measures are presented. The proposed system improves significantly the performance of the baseline approach, and attains results similar to previous approaches.**

***Index Terms*—Acoustic signal processing, singing voice analysis, pitch, fundamental frequency, singing transcription.**

I. Introduction

MELODY transcription techniques are aimed to generate a symbolic output from audio input. This is an important task in the music information retrieval field since melody plays a major role in Western music [1]. Nowadays, there is lot of literature on monophonic and polyphonic melody transcription, commonly following a generic approach in order to be applied to different types of music and instruments. Melodic transcription can be performed at different levels: low-level description (energy, F0), or higher structural levels (note segmentation, ornament detection, etc.) [2]. In this paper we address the specific problem of monophonic singing transcription at note-level, which can be defined as follows: Given the acoustic waveform of a single-voice singing performance, produce a sequence of notes and rests which is melodically and rhythmically as close to the performance as possible [3]. The transcription of ornaments or timbre aspects is out of the scope of this paper.

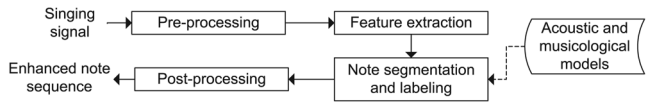

Fig. 1. Diagram of a singing transcription algorithm shown in [3].

Singing transcription is a task related to both melody transcription and speech recognition, and it is challenging even in the case of monophonic signals without accompaniment. This fact is due to the continuous character of the human voice and its acoustic and musical particularities, which are often singer-dependent [4]. Furthermore, automatic singing transcription can be applied to many different contexts. One of the renowned applications of singing transcription is query-by-humming [5], [6], but also other types of applications are related to this task, like singing tutors [7], [8], computer games [9], or the conversion of singing into notes [10] or scores [11], [12].

In the literature, singing transcription has been addressed from many different perspectives. A simple but commonly referenced approach to singing transcription was proposed by McNab [13], the approach relied on several simple pitch-based and amplitude-based segmentation methods. Other singing transcription systems also include rules to deal with intonation issues [14] or auditory models to improve the pitch tracking performance [15], [16]. In a later approach, Ryynänen proposes a probabilistic model of the note event [11], which is described together with a review on the topic in [3]. This probabilistic model has inspired more recent approaches, such as the one in [17]. Finally, Gómez and Bonada [4] address singing transcription for the specific task of a capella flamenco transcription, making use of the note segmentation algorithm defined in [18], which first transcribes the melody into short notes and then performs an iterative process to consolidate them.

Most of the approaches for singing transcription usually fit the schema shown in Fig. 1, as described in [3]. First, a *pre-processing* stage is usually applied to the signal to facilitate the feature extraction process. Some of the techniques applied at this stage are noise reduction [19] or spectral whitening to flatten strong formants in the signal spectrum [20] to facilitate the measurement of the fundamental frequency [3]. The following stage is *low-level feature extraction*. The features typically extracted are pitch, energy, and some other measures to detect unvoiced regions, such as aperiodicity [3] or zero-crossing rate [18]. Then, a *note segmentation and labeling* process produces a symbolic transcription of the input. Finally, this transcription can be analyzed in a *post-processing* block in order

Manuscript received October 01, 2013; revised February 07, 2014; accepted June 02, 2014. Date of publication June 17, 2014; date of current version January 15, 2015. This work was supported by the Ministerio de Economía y Competitividad of the Spanish Government under Project No. TIN2013-47276-C6-2-R and by the Ministerio de Educación, Cultura y Deporte through the "Programa Nacional de Movilidad de Recursos Humanos del Plan Nacional de I-D+i 2008-2011, prorrogado por Acuerdo de Consejo de Ministros de 7 de octubre de 2011." The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Emmanuel Vincent.

The authors are with the Universidad de Málaga, Andalucia Tech, ATIC Research Group, ETSI Telecomunicación, E29071 Málaga, Spain (e-mail: emm@ic.uma.es; lorenzo@ic.uma.es; abp@ic.uma.es; ibp@ic.uma.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2014.2331102

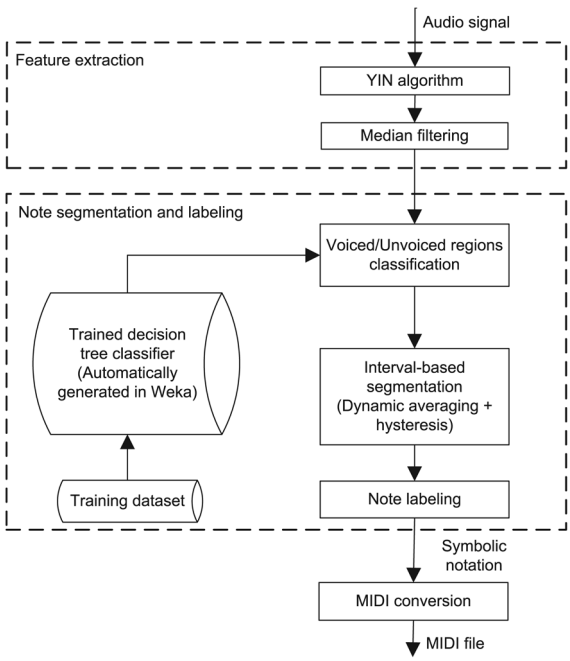

Fig. 2. Scheme of the proposed algorithm for note segmentation and labeling.

to remove spurious notes and to obtain a musically meaningful output.

In this article, we propose an improved method (SiPTH) for pitch-based note segmentation and labeling of monophonic singing audio waveforms. Note that in this paper, we always use the term *pitch* when referring to the F0 of a signal. Our approach implements an *interval-based* note segmentation. We estimate the note changes through the definition of a novel hysteresis process on the pitch-time curve, obtained using the Yin algorithm [21] with certain specific parameters, followed by a number of stages developed for this transcription task. Specifically, the pitch information extracted is used later in the *note segmentation and labeling* block, which provides a note-level representation of the input audio waveform.

Hysteresis is a strongly non-linear phenomenon which occurs in many industrial, physical and economic systems. The exact definition of hysteresis varies from area to area and from paper to paper [22], but it typically implies a non-linear dependence of a system not only on its current state, but also on its past states. In our approach, we apply this concept to the note segmentation problem so that only large and/or sustained pitch deviations produce a change of note. The name SiPTH makes reference to the *si*nging transcription task addressed and to the *p*itch-*t*ime *h*ysteresis effect considered to perform note segmentation.

This paper is organized according to the diagram shown in Fig. 2. In Section II, all the details on the *low-level feature extraction* scheme are explained. This block is based on the Yin algorithm (Section II-A) and the application of a median filter to smooth the resulting curves (Section II-B). The following sections (III, IV, V) correspond to the different blocks of the *note segmentation and labeling* sub-system. In Section III, the algorithm for *voiced and unvoiced region classification* is described. The general idea is to use a previously trained decision tree generated using the Weka data-mining software [23] to identify voiced/unvoiced regions. Once the voiced regions are detected, an *interval-based segmentation* stage for legato phrases is performed (Section IV). This algorithm is a novel interval-based segmentation, which detects note changes through a hysteresis process defined on the pitch-time curve. Then, pitch, power and duration are assigned to the segmented notes to generate the symbolic output from the singing audio signal (Section V). The evaluation methodology and the dataset are described in Section VI. The results and comparisons against other methods are presented in Section VII. Finally, some conclusions are drawn in Section VIII.

II. Low-level Feature Extraction

The proposed scheme first estimates the pitch of the singing voice. The estimation of the pitch has been studied for decades [24], especially in the case of speech [25] and, nowadays, the literature reports a wide set of methods for this purpose.

In our approach, we use the well-known Yin algorithm [21] to perform low-level feature extraction.

A. The Yin Algorithm

The Yin algorithm was developed by de Cheveigné and Kawahara in 2002 [21]. It has been found to be effective in many music transcription systems [26], [11], [27]. This algorithm resembles the idea of the autocorrelation method [28] but introduces relevant improvements. Specifically, the *cumulative mean normalized difference function* $d'_t(\tau)$ peaks at the optimal local period leading to lower error rates than the traditional autocorrelation function (see [21] for details). The cumulative mean normalized difference function $d'_t(\tau)$ is based on the squared difference function $d_t(\tau)$, which is defined as follows:

$$d_t(\tau) = \sum_{j=t}^{t+W} (x_j - x_{j+\tau})^2 \tag{1}$$

where τ is an integer lag variable such that $\tau \in [0, W)$, t is the time index, W is the window size and x_τ is the amplitude of the input signal x at time τ. The difference function is then normalized by the cumulative mean of the function over shorter lag periods:

$$d'_t(\tau) = \begin{cases} 1 & \tau = 0 \\ \dfrac{d_t(\tau)}{\frac{1}{\tau}\sum_{j=1}^{\tau} d_t(j)} & \text{otherwise} \end{cases} \tag{2}$$

The Yin algorithm finds the local minimum with the smallest lag period τ' to perform a parabolic interpolation over the interval $\{\tau' - 1, \tau' + 1\}$ in order to accurately find the minimum period τ_p, which can be converted to frequency using the expression $\mathrm{F0} = f_s/\tau_p$, where f_s is the sampling rate. The aperiodicity measure ap, also called voicing parameter [17], is given by $d'_t(\tau_p)$. This parameter is a function of the strength of the correlation at τ_p, which is related to the overall degree of signal periodicity within the current frame.

The chosen implementation of the Yin algorithm was made by its original author in Matlab [29]. It computes three different curves at frame level: fundamental frequency (F0), RMS (pwr) and aperiodicity (ap). In our case, we apply the Yin algorithm with the following parameters: $\texttt{sr} = 11025$ Hz, $\texttt{minf0} = 80$ Hz, $\texttt{maxf0} = 700$ Hz, $\texttt{thresh} = 0.1$, $\texttt{relfag} = 1$, $\texttt{hop} = 32$ samples, $\texttt{wsize} = 150$ samples, $\texttt{lpf} = 2756$ Hz.

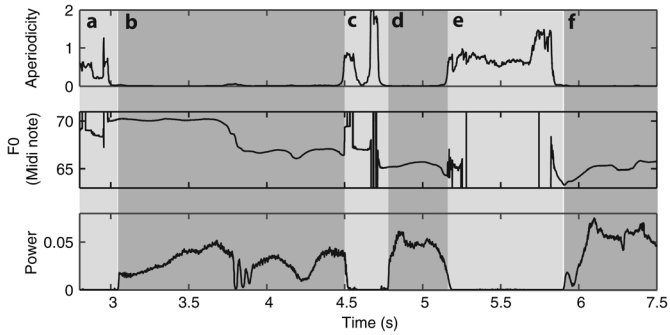

Fig. 3. Output of the YIN algorithm for a child singing performance: fundamental frequency, power, and aperiodicity over time. In this figure, actual sung notes have been marked with shadowed rectangles.

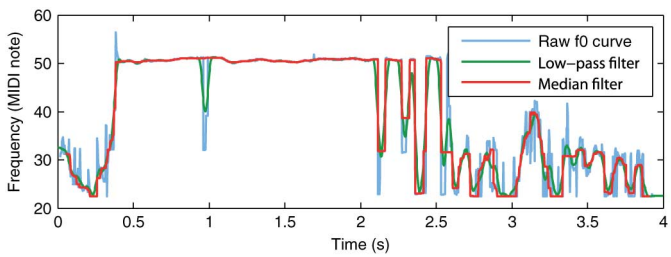

Fig. 4. Sample of application of low-pass and median filtering to a raw pitch curve. The spurious gap at second 1 is removed by the median filter whilst it remains after low-pass filtering.

Voiced frames usually present low aperiodicity, high energy and stable F0. These facts are illustrated in Fig. 3 in which voiced frames have been highlighted in dark grey (intervals *b*, *d* and *f*). The curves shown in Fig. 3 have been obtained from the waveform of a child singing a popular song.

B. Median Filtering

The estimated F0 curve is often noisy due to natural fluctuations of the sound and estimation errors. In order to avoid spurious errors, which could decrease the accuracy of later stages of the system, we apply a median filter to the F0 curve. Median filtering for speech processing was proposed by Rabiner in 1978 [28], and it has been applied to some previous systems for singing transcription [14], [17]. This type of filtering completely removes certain spurious errors, whereas low-pass filtering smooths them. As an example, in Fig. 4 these two types of filters (moving average and moving median) have been applied to a pitch curve. A spurious gap in the F0 curve has been perfectly removed by median filtering. Note the different result of low-pass filtering (moving average) the same signal.

We evaluated the performance of different window sizes (3, 5, 7 samples, as in [30]). The best results (best system performance) were found using a 3 point-median filter.

III. Voiced/Unvoiced Frame Classification

In this section, we propose a method to estimate whether a certain frame of the input signal is *voiced* or *unvoiced*. The process of estimating voiced regions (let region stand for a number of consecutive frames classified as voiced/unvoiced) in singing or speech is usually called *voicing*. In the present paper, only vowels and the consonants 'm','n','l' are considered voiced, as proposed in [14]. Previous approaches estimate voiced sounds using a wide variety of descriptors: the RMS [14], the instantaneous aperiodicity measure [3], the evidence of pitch [31], [32], or the zero crossing rate (ZCR) combined with the RMS [28], [18]. In our method, we mix some ideas from these previous approaches and include some novel improvements that will be described in this section. Specifically, our method is based on the following hypotheses:

1) The pitch slope within a voiced sound is under a certain threshold (apart from octave errors) [33].
2) The energy during a voiced sound is high. Voiced regions should correspond to stable high energy regions.
3) The aperiodicity during a voiced sound is low. It should correspond to stable low aperiodicity intervals.

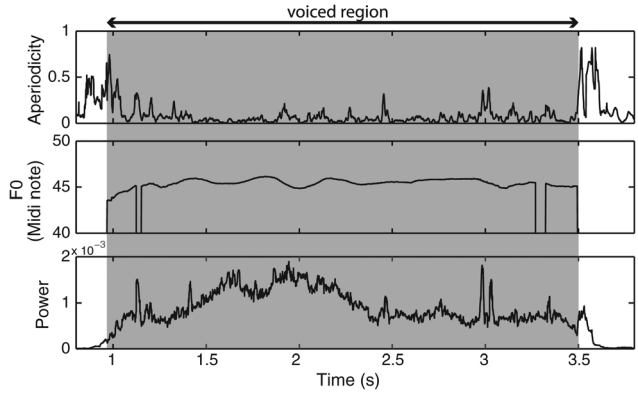

Fig. 5. Extracted features for the case of a rough timbre voice (old male voice). Pitch stability is a better criterion to identify the voiced segment than aperiodicity.

In the case of noisy recordings, unstable loudness and/or rough timbre voices, we have observed that aperiodicity and energy measures present an unstable behavior with many spurious values (see Fig. 5). In contrast, in these cases the pitch curve is usually stable for most of the voiced sounds (apart from octave errors). Therefore, our method is related to the analysis of *pitch contours*. A pitch contour is a temporal sequence of F0 values grouped using heuristics based on auditory streaming cues [32]. In this paper, we introduce the novel concept of *chroma contour*, which is an octave-independent version of the pitch contour (more details are provided in Section III-A). In our approach, only chroma contours are candidates to be voiced regions of the input signal. Thus, our voicing method performs three steps: (1) Estimation of chroma contours, (2) Characterization of chroma contours and (3) Voiced/unvoiced classification of frames.

A. Estimation of Chroma Contours

We propose a method to track stable *chroma* values instead of stable pitch values in order to reduce the effect of octave errors during pitch estimation. For this goal, we have defined two versions of the chroma: basic chroma $C(l) = \{\mathrm{F0}(l) \mod 12\}$ and shifted chroma $C'(l) = \{(\mathrm{F0}(l) + 6) \bmod 12\}$, where $\mathrm{F0}(l)$ is the fundamental frequency in semitones at frame l, and mod is the modulo operation. These expressions have been used to define the chroma gap:

$$\varrho(l) = \min(|C(l) - C(l-1)|, |C'(l) - C'(l-1)|) \qquad (3)$$

where l is the current frame index, and the $\min$ operator selects the minimum of the two values. Note that this expression avoids meaningless outcomes derived from the usage of the modulo 12 operation when pitch values are around a multiple of 12. This

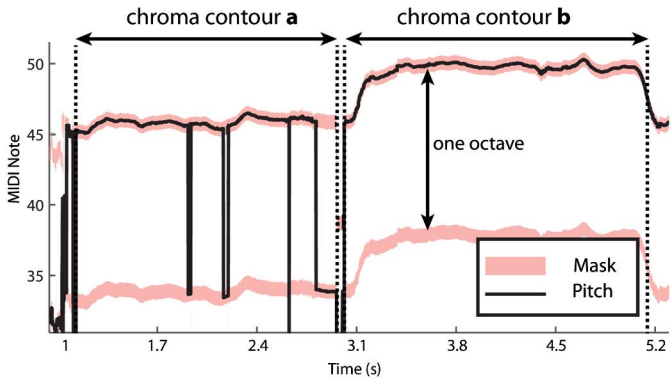

Fig. 6. Stable pitch detection. The black curve represents the estimated pitch value; red regions represent the mask where pitch values can vary between consecutive frames. The use of a mask that allows fast octave jumps avoids fake note changes if octave errors happen.

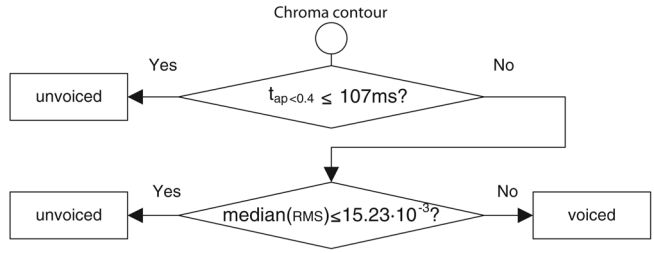

Fig. 7. Decision tree generated by using the C4.5 algorithm implemented in the Weka data-mining software with a very low confidence factor (strong pruning) for the classification of voice/unvoiced frames.

fact is now illustrated: let $\mathrm{F0}(l-1) = 59.9$ and $\mathrm{F0}(l) = 60$, then $C(l) - C(l-1) = 11.9$ and $C'(l) - C'(l-1) = 0.1$, leading to $\varrho(l) = 0.1$, as desired. We define a *chroma contour*, $\mathbf{c}$, as a vector that contains all the chroma values of a set of consecutive frames such that the chroma gap remains under a certain threshold, ϱ_{th}: $\mathbf{c} = [C(l_c), C(l_c+1,)\ldots, C(l_n)]$, with $\varrho(l_i) < \varrho_{th}, \forall l_i = \{l_c, l_c+1, \ldots, l_n\}$ (see Fig. 6). Note that we omit the chroma contour index for simplicity.

The maximum chroma gap, ϱ_{th}, must be set. According to [33], the maximum pitch slope found in a large set of speakers is 216 semitones per second (st/s). With a hop size hop samples/frame and a sampling rate sr samples/s (see Section II-A), the time hop we are using in our analysis is $h_s = \frac{hop}{sr} = 2.9$ ms, then the maximum pitch gap between consecutive frames according to this work is 216 st/s $\cdot\, h_s = 0.63$ st. In our case, the maximum chroma gap between consecutive frames has been set to $\varrho_{th} = 1$ semitone. Observe that the algorithm described to estimate the chroma contours can be seen as an octave-independent pitch tracking process (Fig. 6).

B. Characterization of Chroma Contours

We have observed that chroma contours can correspond to unvoiced sounds under certain circumstances, e.g. some sibilant sounds or periodic background noises. So, an additional process is needed to refine the voiced/unvoiced classification of chroma contours. To this end, we analyzed the music collection described in Section VI, which contains 1154 seconds of singing audio. We computed a set of 20 descriptors for each voiced/unvoiced region (specifically 4243 regions, being 2149 voiced and 2094 unvoiced): mean and median of the RMS, mean and median of the aperiodicity, zero crossing rate (ZCR), length in milliseconds of the longest segment with aperiodicity under a set of thresholds $\{0.1, 0.2, \ldots 0.5\}$ and length in milliseconds of the longest segment with RMS over a set of thresholds $\{0.01, 0.02, \ldots 0.1\}$. This set of descriptors has been used to train a J48 decision tree [34] in the Weka data-mining software using a 66% of the dataset for training (2829 instances chosen in random order), and the remainder 34% for testing (1414 instances). J48 is a open source Java implementation of the algorithm C4.5 [35] for the generation of decision trees. We have used the default set of parameters for the `weka.classifiers.trees.J48` classifier, except for the coincidence factor C. The default value for C is 0.25, but we have set $C = 3 \cdot 10^{-8}$ in order to perform strong pruning to reduce the over-fitting. In decision trees, the over-fitting phenomena can occur when the size of the tree is too large compared to the number of training examples [36]. In our case, the generated decision tree only uses two descriptors: the length in milliseconds of the longest segment with aperiodicity under 0.4 and the median of the RMS, achieving an F-measure of 0.988. At the sight of the results obtained, we conclude that voiced chroma contours can be accurately identified with a simple decision tree, which is described in Section III-C, that only uses the two descriptors selected.

C. Voiced/Unvoiced Classification of Frames

All the frames of the input signal that do not belong to a chroma contour are directly classified as *unvoiced*. The frames belonging to a chroma contour can be voiced or unvoiced depending on the results of the decision tree for such chroma contour. As explained in Section III-B, two descriptors are computed for each chroma contour. Then, all the frames belonging to the same chroma contour are classified together with the decision tree shown in Fig. 7.

IV. Interval-based Note Segmentation

The estimation of voiced chroma contours results in a rough note segmentation. Silences and some consonants are detected as unvoiced regions between notes, producing a good segmentation of non-legato phrases. However, pitch variations within legato fragments are not segmented yet, since they all belong to the same chroma contour. Probably, the simplest possible segmentation could be done by simply rounding a rough pitch estimate to the closest MIDI note n_i, assuming $A4 - 440$ Hz standard tuning and taking all pitch changes as note boundaries [26]. However, the singing voice has not a constant tuning reference, especially in the case of untrained singers, and this simple quantization produces many fake note changes. Therefore, we propose a novel interval-based segmentation algorithm that detects a note change only if large and/or sustained pitch deviations are found. This approach is appropriate to deal with vibrato, or with untrained singers whose pitch curve can rapidly oscillate during each note.

In the following subsections, we describe the details of our algorithm. First, in Section IV-A we introduce the concept of dynamic averaging to obtain a curve that roughly estimates the pitch of the notes even when their exact boundaries are unknown. Then, in Section IV-B, we explain how a hysteresis relationship between the instantaneous F0 and the dynamic average is defined in order to detect meaningful pitch deviations. Finally,

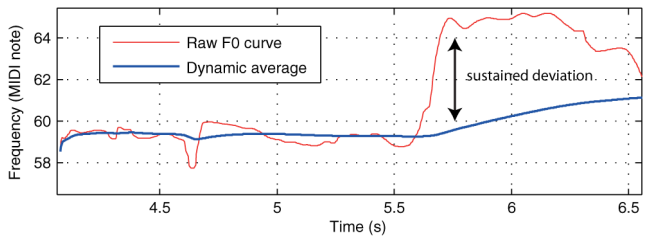

Fig. 8. Dynamic averaging of the pitch curve. Fast variations of pitch at the beginning of the note are tracked, whereas later strong changes can be easily detected.

in Section IV-C, we discuss about the exact time instant where note changes should be placed to define note-labeled audio segments.

A. Dynamic Averaging

In order to obtain a more stable version of the pitch curve, we compute the *dynamic average* of F0 in the voiced frames as follows:

$$\mathrm{F0}_A(l) = \frac{\sum_{k=l_0}^{l} \mathrm{F0}(k)}{l - l_0 + 1} \tag{4}$$

where l_0, with $l_0 \leq l$, is the closest index of the first frame of a voiced region or the first frame of a new note detected according to the description in Section IV-C. $\mathrm{F0}_A(l)$ stands for the dynamic average at frame l, with $\mathrm{F0}(k)$ the pitch detected at frame k. When l is close to l_0, with $l \geq l_0$, $\mathrm{F0}_A$ is similar to the F0 curve detected (Fig. 8). However, as the duration of the detected note grows, $\mathrm{F0}_A(l)$ turns into a more stable, representative pitch value of the note. It is important to observe that this dynamic average does not represent the final transcribed pitch value of the notes. Instead, the transcribed pitch of each note is accurately computed at a later stage using a weighted alpha-trimmed mean filter (Section V-A).

A slight variation of the dynamic average concept has been previously used by McNab *et al.* in [13]. In their work, regions with slow F0 variation are grouped and dynamically averaged in order to estimate the successive note changes. Our approach uses a different criterion to estimate note changes (see Section IV-B). While McNab *et al.* consider a note change as soon as the instantaneous F0 deviates from the dynamic average, we consider a note change only if a large and/or sustained deviation of the F0 with respect to the dynamic average is found. We detect large and/or sustained pitch deviations by means of the definition of a novel hysteresis effect of the pitch-time curve.

B. Hysteresis

In order to find note changes, we compute the cumulative pitch deviation (or deviation area) $\Gamma(l)$ between the instantaneous pitch curve F0 and the dynamic average $\mathrm{F0}_A$. Let l_0 stand for the first frame index of a note (the note index has been dropped for simplicity). Note that the first frame of each note either coincides with the first frame of a voiced chroma contour or it is found according to the criterion described in Section IV-C. The cumulative pitch deviation at frame l_0 is $\Gamma(l_0) = 0$, according to Section IV-A, and it is calculated using the following recursive equation, for $l > l_0$:

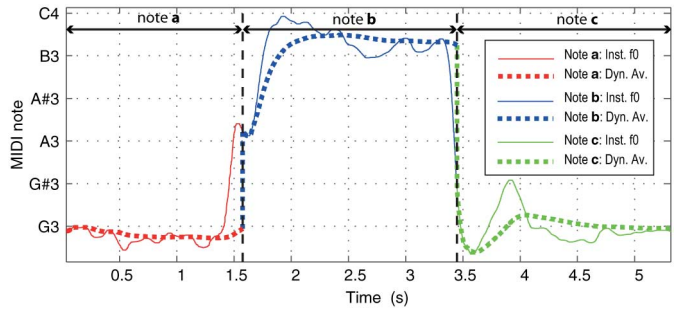

Fig. 9. Representation of the hysteresis process for the detection of note changes. Samples are taken from real data: from $\approx G3$ to $\approx B3$ to $\approx G3$. The instantaneous F0 and the dynamic average $\mathrm{F0}_A$ for each note are shown. Strong and/or sustained deviations of the instantaneous F0 with respect to the dynamic average trigger the detection of note changes. Observe that although the instantaneous F0 estimated for the final note deviates more than a semitone, the system does not detect a spurious note change.

$$\Gamma(l) = \begin{cases} \Gamma(l-1) & , \text{ if } |\delta_{\mathrm{F0}}(l)| < \delta_{th} \\ \Gamma(l-1) + \delta_{\mathrm{F0}}(l) \cdot h_s & , \text{ otherwise} \end{cases} \tag{5}$$

with $\delta_{\mathrm{F0}}(l) = \mathrm{F0}(l) - \mathrm{F0}_A(l)$ the instantaneous pitch deviation, in semitones, between the instantaneous pitch detected and the dynamic average pitch curve $\mathrm{F0}_A$. h_s is the hop size in seconds (defined in Section III-A). δ_{th} is named interval threshold (in semitones). Note that instantaneous pitch deviations of magnitude under δ_{th} are not considered significant. The recursion in eq. (5) ends when the current chroma contour ends or a new note is detected.

In order to find note changes, let l^* denote the first frame index in the current note such that $|\Gamma(l^*)| \geq \Gamma_{th}$, with Γ_{th} a certain deviation area threshold. This event indicates that a new note has been detected. The initial frame of the new note will be precisely defined according to the criterion in Section IV-C. Then, l_0 will be replaced by the first frame index of the new note and the dynamic average and the cumulative pitch deviation values will be reset to restart the detection process (Fig. 9).

The influence of δ_{th} and Γ_{th} on the performance of the scheme is evaluated in Section VII, leading to the selection of the following values: $\delta_{th} = 0.5$ semitones and $\Gamma_{th} = 0.1$ semitones $\times$ seconds.

C. Exact Position of the Onset

Defining the exact position of the note change in singing voice is not an easy task. When singing legato notes, the transition between notes is naturally smoothed and it becomes an interval, not an instant.

In this paper, a note segmentation method that makes use of two specific events related to the cumulative pitch deviation, is proposed. These events are (see Fig. 10):

- The frame l^* when the cumulative pitch deviation (deviation area) exceeds the threshold Γ_{th}.
- The first frame, l', of the last one of the significant pitch deviation areas (where $|\delta_{\mathrm{F0}}(l)| \geq \delta_{th}$) in the current note.

A note change is considered to happen in the middle point between these two time instants: the first frame of a new note l_0 is found by rounding to the nearest integer the mean of the two

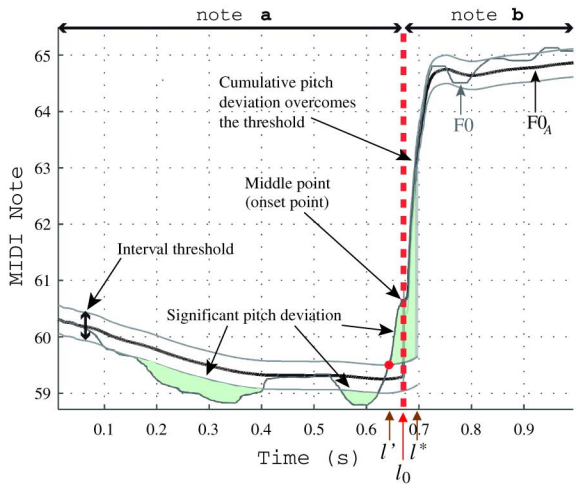

Fig. 10. Segmentation process. The dynamic averaging and the hysteresis effect are used to detect note changes. In this example, the interval threshold δ_{th} has been set to 0.25 semitones.

frame indexes considered. This choice has empirically proved to be a good compromise in most cases to define note segments.

V. Note Labeling and Conversion to MIDI

The different note segments detected, must be labeled to generate a symbolic notation. In the proposed labeling procedure, each note is assigned the following values: *pitch*, *onset/offset time positions* and *volume*.

A. Assigned Pitch

A constant pitch value must be assigned to each note in order to perform symbolic transcription. This value is computed in two steps: first, the precise pitch value of the note is estimated, then it is rounded to the closest semitone.

In order to estimate the precise pitch value of each note, we assume that pitch transients and unstable oscillations are not representative of the perceived pitch, and therefore they must not be considered (a similar idea has been applied in previous approaches [13], [37], [15]). To this end, we propose the use of the *energy-weighted α-trimmed mean* filter [38] over the pitch curve for each note segment. In this filter, the extreme (high and low) values of F0 (typically outliers) are excluded and only the remainder values are considered in the weighted average. Let $\mathrm{F0}(a_i)$ denote the pitch values of the frames of a note arranged in ascending order of magnitude: $\mathrm{F0}(a_1) \leq \mathrm{F0}(a_2) \leq \ldots \leq \mathrm{F0}(a_L)$, with L the number of frames of a certain note. Then, the pitch value $\mathrm{F0}_\alpha$ assigned to each note is computed as follows:

$$\mathrm{F0}_\alpha = \begin{cases} \dfrac{\sum_{i=[\alpha L]+1}^{L-[\alpha L]} \mathrm{F0}(a_i)\cdot \mathrm{E}(a_i)}{\sum_{i=[\alpha L]+1}^{L-[\alpha L]} \mathrm{E}(a_i)}, & L \text{ odd} \\ \dfrac{\sum_{i=[\alpha(L-1)]+1}^{L-[\alpha(L-1)]} \mathrm{F0}(a_i)\cdot \mathrm{E}(a_i)}{\sum_{i=[\alpha(L-1)]+1}^{L-[\alpha(L-1)]} \mathrm{E}(a_i)}, & L \text{ even} \end{cases} \quad (6)$$

where $E(a_i)$ is the energy (sum of square values of the signal) in frame a_i. The parameter α indicates the amount of values to be removed ($\alpha \in [0, 0.5]$): with $\alpha = 0$ the conventional mean is obtained whereas with $\alpha = 0.5$ all the values except central one are removed (leading to the median filter). In our case we have used $\alpha = 0.3$ (more details about tuning α are provided in Section VII). The operator $[\cdot]$ is the greatest integer function.

Once the precise pitch value $\mathrm{F0}_\alpha$ of each note is known, in order to obtain a symbolic transcription of the result, a reference tuning must be considered. We have assumed the standard tuning reference: $A4 = 440$ Hz. Any other tuning could be used, in fact, some previous approaches consider the possibility of a different tuning reference that can be constant [14] or smoothly time variant [13]. However the selection of the standard tuning allows as to use the MIDI scale to perform the transcription. So, the $\mathrm{F0}_\alpha$ of each note is rounded to the nearest semitone of the MIDI scale in order to compute the assigned pitch value $\mathrm{F0}'$.

B. Onset and Offset Time

The estimated onset and offset[1] times are key aspects for a proper rhythmic transcription of the singing melody. In our approach, the onsets of the transcribed notes are placed according to the procedure described in Section IV-C. Similarly, the offset time is found when either an unvoiced region or a note change is found.

C. Velocity

According to the MIDI specification [40], the velocity of a note represents its loudness. We estimate the loudness of each note by averaging its power evolution. In the proposed approach we assume that the gain of the input signal has been adjusted to cover the whole dynamic range. We have not evaluated this aspect of the transcription. However, a qualitative analysis of the results showed that the volume of the transcribed notes was perceptually similar to the original audio.

D. MIDI Conversion

The final MIDI transcription was performed with the MIDI tool kit for Matlab developed by Ken Schutte [41]. This tool kit allows to read and write MIDI files by using Matlab matrices easily. In the proposed scheme, each note corresponds to a MIDI *note message* including information about onset and offset instants, MIDI note number (rounded pitch), and velocity.

VI. Evaluation Methodology

The standard approach for the evaluation of melody transcription systems is to compare the automatic transcriptions with human annotations. In this section, we describe the music collection gathered for evaluation (Section VI-A), the chosen criteria to build the ground truth, (Section VI-B) and a novel set of evaluation measures (the definition of the measures can be found in Section VI-C and details on the computation can be found in Section VI-D).

[1]We define the *offset* time as the time frame when an active note changes to an inactive state [39].

A. Music Collection

Our dataset consists of 38 melodies sung by adult and child untrained singers, recorded with a sample rate of 44100 Hz and a resolution of 16 bits. Generally, the recordings are not clean and some background noise is present. The duration of the excerpts ranges from 15 to 86 seconds and the total duration of the whole dataset is 1154 seconds. This music collection can be broken down into three categories, according to the type of singer:

- Children (our own recordings): 14 melodies of traditional children songs (557 seconds) sung by 8 different children (5-11 years old).
- Adult male: 13 pop melodies (315 seconds) sung by 8 different adult male untrained singers. These recordings were randomly chosen from the public MTG-QBH dataset [42].
- Adult female: 11 pop melodies (281 seconds) sung by 5 different adult female untrained singers. These recordings were also randomly chosen from the public MTG-QBH dataset.

Note that in this collection the pitch and the loudness can be unstable and vibratos are not frequent.

B. Ground Truth

The described music collection has been manually annotated to build the ground truth. Since there is no standard criteria to manually annotate musical content [2], we have defined our own methodology according to the specific context and goals of our system. First, we have transcribed the audio recordings with a baseline algorithm (see Section VII-A), and then all the transcription errors have been corrected by an expert musician with more than 10 years of academic training in music. The transcription errors were corrected by listening, at the same time, to the synthesized transcription and the original audio. The musician was given a set of instructions about the specific criteria to annotate the singing melody:

- The onsets are placed at the beginning of voiced segments and in each clear change of pitch or phoneme. In the case of 'l', 'm', 'n' voiced consonants + vowel (e.g. 'la'), the onset is not placed at the beginning of the consonant but at the beginning of the vowel.
- The annotated pitch of each note is the closest semitone to the pitch of the sung note, as perceived by the expert.
- Ornaments such as pitch bending at the beginning of the notes or vibratos are not annotated. Some considerations about this type of ornaments can be found in [37].
- Portamento between notes is ignored.

C. Evaluation Measures

In the literature, we can find many different approaches to compare automatic transcriptions against the ground truth. In [11], two different evaluation measures for singing transcription are proposed: *frame-based error* and *note-based error*. The frame-based error considers the ratio of correctly transcribed frames, and the note-based error considers the ratio of correctly transcribed notes (their duration is ignored). According to [11], a frame or note is correctly transcribed when the rounded pitch (to semitones) of the frame or note equals the ground truth, and the onset of the transcribed note is within a tolerance window of ± 50 ms. In [4], a similar measure has been used together with three more measures typically applied to melody extraction [32]: *voicing recall*, *voicing false alarm* and *raw chroma accuracy*. Other approaches try to break down the type of transcription errors, e.g. insertions, deletions, etc. [15], [43], but the duration of the errors is not considered.

In this paper, we propose a novel set of evaluation measures that reports details about the specific type of transcription mistakes make and their duration:

1) Voicing: We consider two measures as stipulated by MIREX for audio melody extraction, which are also used in [4]: *voicing recall*, i.e. percentage of voiced frames in the reference that are classified as voiced by the algorithm, *voicing false alarm*, i.e. percentage of unvoiced frames in the reference that are classified as voiced by the algorithm.

2) Pitch Accuracy: We measure the *raw pitch accuracy*, i.e. the percentage of voiced frames where the pitch estimation is correct. In our case, we consider that the pitch is correct if the rounded pitch (to semitones) is the same.

3) Note-based and Frame-based Error Rates by Categories: We classify each note from both the transcription and the ground truth into one of the following six categories:

1) Non-detected note (ND): A note n_i in the reference melody that does not overlap any note n_j at the transcribed melody, neither in time nor in pitch.
2) Spurious note (PU): A note n_j in the transcribed melody that does not overlap any note n_i in the reference melody, neither in time nor in pitch.
3) Split note (S): A single note n_i from the reference melody that has been incorrectly segmented into different consecutive notes $n_{j_1}, n_{j_2} \dots n_{j_n}$ in the transcribed melody. The onset difference between n_i and n_{j_1} must be within ± 50 ms, the whole group $n_{j_1}, n_{j_2} \dots n_{j_n}$ must overlap more than 50% of n_i, and the rounded pitch (to semitones) of n_i must be the same as $n_{j_1}, n_{j_2} \dots n_{j_n}$.
4) Merged note (M): A single note n_j at the transcribed melody that results from several merged notes $n_{i_1}, n_{i_2} \dots n_{i_n}$ in the reference melody. The onset difference between n_j and n_{i_1} must be within ± 50 ms, the whole group $n_{i_1}, n_{i_2} \dots n_{i_n}$ must overlap more than 50% of n_j and the rounded pitch (to semitones) of n_j must be the same as $n_{i_1}, n_{i_2} \dots n_{i_n}$. If a note is classified as Split and Merged, then it will be considered neither Split nor Merged since this fact means that there are two pairs of overlapped notes and they will be classified into one of the following categories: CD or BD (to be defined).
5) Correctly detected note (CD): A note n_j from the transcribed melody that *hits* a note n_i from the reference melody in time and pitch. We define a hit in a similar way to [11]: the rounded pitch (to semitones) must be the same, the onset difference between n_j and n_{i_1} must be within ± 50 ms, n_i must overlap n_j more than the 50% of both n_i and n_j, as described in Section VI-D. If a note has been already classified as Split or Merged, it is not classified as Correctly Detected.
6) Badly detected note (BD): A note n_j from the transcribed melody that overlaps a note n_i from the reference, but it has not been classified into any of the previous categories. This case corresponds to transcribed notes that have the same pitch as the reference, but the onset difference is larger than ± 50 ms or their duration is very different.

Additionally, we compute the number of frames belonging to the notes in each of the six categories. Note that these categories are computed in order, from ND to BD. The proposed algorithm to identify them is described in Section VI-D. Therefore, we have considered the note-rate ($\mathrm{NR}_{\mathbf{X}}$) and frame-rate ($\mathrm{FrR}_{\mathbf{X}}$) for each category $\mathbf{X} \in \mathrm{S}, \mathrm{M}, \mathrm{CD}, \mathrm{BD}$, defined as follows:

$$\mathrm{NR}_{\mathbf{X}} = \frac{1}{2}\left(\frac{N_{\gamma\mathbf{X}}}{N_\gamma} + \frac{N_{\phi\mathbf{X}}}{N_\phi}\right) \quad \mathrm{FrR}_{\mathbf{X}} = \frac{1}{2}\left(\frac{Fr_{\gamma\mathbf{X}}}{Fr_\gamma} + \frac{Fr_{\phi\mathbf{X}}}{Fr_\phi}\right) \tag{7}$$

where N_γ is the total number of notes in the ground truth, N_ϕ is the total number of notes in the transcription, $N_{\gamma\mathbf{X}}$ is the number of notes in the ground truth belonging to category $\mathbf{X}$ (i.e. S, M, ...), $N_{\phi\mathbf{X}}$ is the number of notes in transcription belonging to category, $\mathbf{X}Fr_\gamma$ is the number of frames of all the notes in the ground truth, Fr_ϕ is the number of frames of all the notes in the transcription, $Fr_{\gamma\mathbf{X}}$ is the number of frames of the notes in the ground truth belonging to category $\mathbf{X}$, and $Fr_{\phi\mathbf{X}}$ is the number of frames of the notes in the transcription belonging to category $\mathbf{X}$. Note that the importance of frame-based measures relies on the fact that these measures account for the performance evaluation taking into account the actual duration of the notes belonging to a certain category $\mathbf{X}$ with respect to the duration of all the notes. Conversely, note-based measures do not consider the actual duration of the notes.

Since Non-detected notes (ND) are only present in the ground truth, and Spurious notes (PU) are only present in the transcription, we define the note-rate and the frame-rate measures for these categories as follows:

$$\mathrm{NR}_{\mathbf{ND}} = \frac{N_{\gamma\mathbf{ND}}}{N_\gamma} \quad \mathrm{FrR}_{\mathbf{ND}} = \frac{Fr_{\gamma\mathbf{ND}}}{Fr_\gamma} \tag{8}$$

$$\mathrm{NR}_{\mathbf{PU}} = \frac{N_{\phi\mathbf{PU}}}{N_\phi} \quad \mathrm{FrR}_{\mathbf{PU}} = \frac{Fr_{\phi\mathbf{PU}}}{Fr_\phi} \tag{9}$$

The proposed evaluation measures are computed for each melody separately and then all the error rates are averaged to report the final results.

D. Algorithm to Identify the Category of Transcription Errors

Let $\vec{n}v_i^\gamma$ denote a vector of length N_{frames} containing the rounded pitch value (at frame level) of the note i of the reference melody. Note that the pitch value is rounded to exact semitones. If the note i is played at frame $l \in [1, N_{\mathrm{frames}}]$, then $nv_i{}^\gamma(l)$ equals the MIDI number of the note, otherwise $nv_i{}^\gamma(l) = 0$. The same procedure is applied to the notes in the transcribed melody in order to define a vector $\vec{n}v_j^\phi$ containing the pitch value of the note j in the transcribed melody. As an example, suppose that a ground truth melody consists of three consecutive notes: G4 + 20 cents (MIDI number 67.2), A4 - 10 cents (MIDI number 68.9) and B4 + 30 cents (MIDI number 71.3). Then, the vector $\vec{n}v_i^\gamma$ for each note will be:

$$\begin{aligned}
\vec{nv}_1^\gamma &= [\,67 \quad 67 \quad 67 \quad \cdots \quad 0 \quad 0 \quad 0 \quad \cdots \quad 0 \quad 0 \quad 0\,] \\
\vec{nv}_2^\gamma &= [\,0 \quad 0 \quad 0 \quad \cdots \quad 69 \quad 69 \quad 69 \quad \cdots \quad 0 \quad 0 \quad 0\,] \\
\vec{nv}_3^\gamma &= [\,0 \quad 0 \quad 0 \quad \cdots \quad 0 \quad 0 \quad 0 \quad \cdots \quad 71 \quad 71 \quad 71\,]
\end{aligned} \tag{10}$$

The vectors are combined to define the matrices M_γ (size $N_\gamma \times N_{\mathrm{frames}}$) and M_ϕ (size $N_\phi \times N_{\mathrm{frames}}$):

$$M_\gamma = \begin{pmatrix} \vec{nv}_1^\gamma \\ \vec{nv}_2^\gamma \\ \vdots \\ \vec{nv}_{N_\gamma}^\gamma \end{pmatrix} \qquad M_\phi = \begin{pmatrix} \vec{nv}_1^\phi \\ \vec{nv}_2^\phi \\ \vdots \\ \vec{nv}_{N_\phi}^\phi \end{pmatrix} \tag{11}$$

These two matrices are used to build a new matrix M with size $N_\gamma \times N_\phi$. Each element in M, m_{ij}, represents the number of overlapped frames between the note i in the ground truth and the note j in the transcribed melody. This matrix is computed as follows:

$$M = \mathcal{F}(M_\gamma, M_\phi) \tag{12}$$

where the function $\mathcal{F}(\cdot)$ counts the number of overlapped frames in time and pitch between the ground truth and the transcription. This function defines every element in M as:

$$m_{ij} = \sum_{k=1}^{N_{\mathrm{frames}}} f(m_{\gamma_i}(k), m_{\phi_j}(k)) \tag{13}$$

where the function $f(\cdot)$ returns 1 if a coincidence in pitch and time between the ground truth and the transcription is found, otherwise it returns 0:

$$f(m_{\gamma_i}(k), m_{\phi_j}(k)) = \begin{cases} 1, & \text{if} \quad m_{\gamma_i}(k) = m_{\phi_j}(k) \\ 0, & \text{if} \quad m_{\gamma_i}(k) \neq m_{\phi_j}(k) \end{cases} \tag{14}$$

With all this, the matrix M provides information about the reciprocal overlap between the ground truth and the transcription. Two different normalization factors should be applied to this matrix in order to obtain $M_{\gamma\to\phi}$ and $M_{\phi\to\gamma}$. In the case of $M_{\gamma\to\phi}$, each row i should be divided by the length of the note i in the ground truth (in frames). On the other hand, for $M_{\phi\to\gamma}$, each row should be divided by the length of the note j in the transcribed melody. The length of each note n_i in frames is defined as $\lambda(n_i) = l_{\mathrm{offset}}(n_i) - l_{\mathrm{onset}}(n_i)$. Let $\vec{l}_\gamma$ and $\vec{l}_\phi$ denote two vectors containing the required normalization factors:

$$\vec{l}_\gamma = \left[\,\lambda^{-1}(n_1^\gamma) \quad \lambda^{-1}(n_2^\gamma) \quad \cdots \quad \lambda^{-1}(n_{N_\gamma}^\gamma)\,\right] \tag{15}$$

$$\vec{l}_\phi = \left[\,\lambda^{-1}(n_1^\phi) \quad \lambda^{-1}(n_2^\phi) \quad \cdots \quad \lambda^{-1}(n_{N_\phi}^\phi)\,\right] \tag{16}$$

and let $\mathrm{diag}(\vec{x})$ denote the operation that produces a diagonal matrix whose non-null elements are given by $\vec{x}$. Using this operator, two normalization matrices are defined:

$$L_\gamma = \mathrm{diag}(\vec{l}_\gamma) \quad L_T = \mathrm{diag}(\vec{l}_\phi) \tag{17}$$

Then, the matrices that provide information about the ratio of overlap between the ground truth and the transcription and vice versa, $M_{\gamma\to\phi}$ and $M_{\phi\to\gamma}$ respectively, can be computed according to:

$$M_{\gamma\to\phi} = L_\gamma \cdot M \tag{18}$$

$$M_{\phi\to\gamma} = M \cdot L_\phi \tag{19}$$

Additionally, we define the *onset function* $o(\vec{n}v)$ as the index of the first non-zero value of the vector $\vec{n}v$. For instance, for a given note $\vec{n}v_1^{\phi} = [0 \quad 0 \quad 0 \quad 60 \quad 60 \quad \cdots]$, $o(\vec{n}v_1^{\phi}) = 4$. We then define the following two vectors:

$$\overrightarrow{O_\gamma} = \begin{pmatrix} o(\vec{n}v_1^{\gamma}) \\ o(\vec{n}v_2^{\gamma}) \\ \vdots \\ o(\vec{n}v_{N_\gamma}^{\gamma}) \end{pmatrix} \qquad \overrightarrow{O_\phi} = \begin{pmatrix} o(\vec{n}v_1^{\phi}) \\ o(\vec{n}v_2^{\phi}) \\ \vdots \\ o(\vec{n}v_{N_\phi}^{\phi}) \end{pmatrix} \tag{20}$$

These vectors are used to define the *onset difference matrix* OD, which contains the absolute onset difference in milliseconds between all the notes of the transcribed melody and all the notes of the reference. The elements of OD, od_{ij}, are defined as follows:

$$od_{ij} = |O_{\gamma_i} - O_{\phi_j}| \cdot h_s \tag{21}$$

with h_s the hop size in seconds.

Now, a set of rules are applied in order to determine the category of each note n_i from the reference and each note n_j from the transcription. These categories and rules are:

- **Not detected note (ND) (i_0)**: We consider that a note i_0 in the ground truth is not-detected if $M_{\gamma \to \phi_{i_0 j}} = 0 \; \forall j \in \{1 \ldots N_\phi\}$. That means that there is no overlap between the note i_0 in the ground truth and the whole transcription.
- **Spurious note (PU) (j_0)**: We consider that a note j_0 in the transcription is a spurious note if $M_{\phi \to \gamma_{j_0 i}} = 0 \; \forall i \in \{1 \ldots N_\gamma\}$. That means that there is no overlap between the note j_0 from the transcription and the whole ground truth.
- **Split note (S) ($i_0 \to j_0 \ldots j_n$)**: We consider that a note i_0 in the ground truth is split into a set of notes $j_0 \ldots j_n$ in the transcription if $M_{\phi \to \gamma_{j_{0\ldots n} i_0}} > 0$ with $n > 1$, $od_{i_0 j_0} < 50$ ms and $\sum_0^n M_{\phi \to \gamma_{j_{0\ldots n} i_0}} > t$.
- **Merged note (M) ($i_0 \ldots i_n \to j_0$)**: We consider that several notes $i_0 \ldots i_n$ in the ground truth are merged into a single note j_0 in the transcription if $M_{\gamma \to \phi_{i_{0\ldots n} j_0}} > 0$ with $n > 1$, $od_{i_0 j_0} < 50$ ms and $\sum_0^n M_{\gamma \to \phi_{i_{0\ldots n} j_0}} > t$. That means that there are several notes in the ground truth that overlap a single long note in the transcription.
- **Correctly detected (CD) ($i_0 \to j_0$)**: A note i_0 in the ground truth, has been correctly transcribed a note j_0 in the transcription, if $M_{\gamma \to \phi_{i_0 j_0}} > t$, $M_{\phi \to \gamma_{j_0 i_0}} > t$ and $od_{i_0 j_0} < 50$ ms. Notes that have been previously classified as *Split* or *Merged* are not considered as coincident notes. Note that this implies a bidirectional coincidence in both time and pitch between the ground truth and the transcription.
- **Badly detected (BD) ($i_0 \to j_0$)**: A note i_0, in the ground truth, has been badly transcribed as note j_0 in the transcription, if $M_{\gamma \to \phi_{i_0 j_0}} > 0$ and it has not been classified into any of the previous categories.

Note that these categories are computed in the order described. In Fig. 11 we show a comprehensive example to understand each type of error.

VII. Results & Discussion

In this section, the performance of the proposed scheme is evaluated according to the described evaluation methodology. The results are compared against a simple baseline approach based on the Yin algorithm, a HMM based approach based on Ryynänen work [11][3] and the transcription scheme develop by Gomez and Bonada in [4].

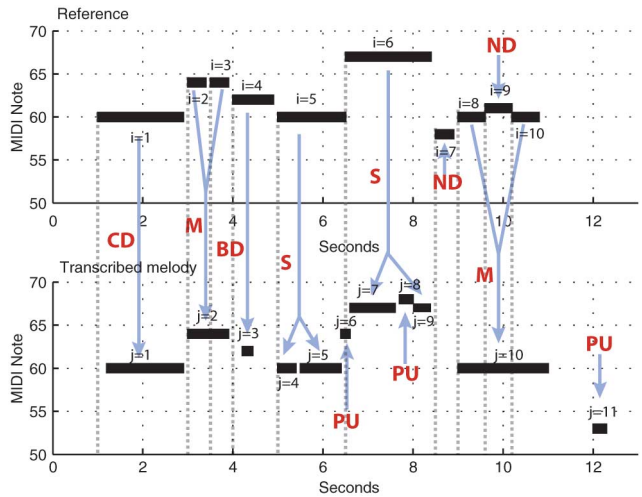

Fig. 11. Example of comparison between the ground truth and the transcribed melody. All the error types defined are illustrated using a sample outcome of the proposed evaluation algorithm.

A. Baseline Approach

We have compared our algorithm with a baseline approach. According to [26], the simplest possible segmentation consists of simply rounding a rough pitch estimate to the closest MIDI note n_i and taking all pitch changes as note boundaries. Therefore, we have implemented a baseline approach to estimate the pitch using the Yin algorithm and the parameters described in Section II-A so that it can be easily implemented by other researchers for comparison purposes. Additionally, we consider a frame as unvoiced if its aperiodicity is under < 0.4, and we discard the notes shorter than 100 ms.

B. HMM-based Approach

We have also implemented a simplified version of Ryynänen's approach [11][3], in which note events and silences have been modelled with a left-to-right four-state Hidden Markov Model (HMM). The first three states have been associated to the attack-sustain-release events and the fourth state to noise/silence. For each frame, three descriptors have been obtained as described in [3]: fundamental frequency, aperiodicity, and *accent* (see [44] for details about this feature). The emission probabilities have been modelled using Gaussian mixtures models (GMM) with 3 Gaussian distributions per state. The whole model has been trained using the music collection described in Section VI-A, and each state has been manually associated with different segments of the recording as follows: state (1): first frame of each note (i.e. the onset), state (2): sustain of each note (between the onset and the offset), state (3): last frame of each note (i.e. the offset) and state (4): unvoiced regions. In our implementation, we have not included the musicological model described in [11], since we consider that the singer does not necessarily follow any musicological constrain related to note sequences.

C. Evaluation and Discussion

In Fig. 12, we show the results obtained for each evaluation measure computed for our system, the baseline approach de-

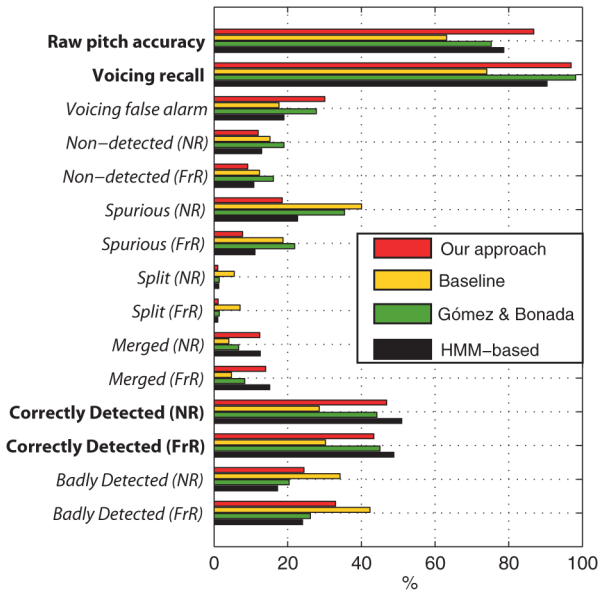

Fig. 12. Detailed performance evaluation of the monophonic singing transcription system proposed, the baseline approach, the HMM approach and the Gómez and Bonada transcription scheme [4] (confidence interval set to 0.5 semitones). The measures that should be maximized are in bold and the measures that should be minimized are in cursive. All measures have been expressed in percentage.

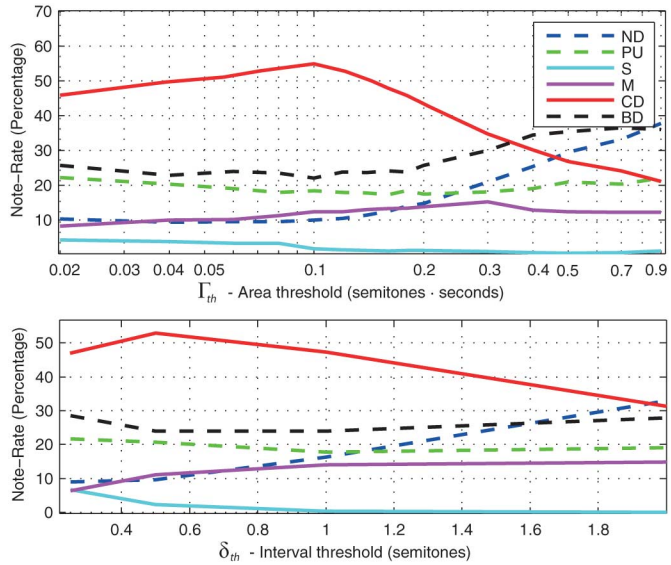

Fig. 13. Illustration of the influence of the system parameters δ_{th} and Γ_{th} on the pitch-based segmentation process. The optimal performance is achieved when the rate of correctly detected notes (CD) is maximum, and the rates of the rest of errors (ND, PU, S, M and BD) are minimum. (Top: area threshold $\Gamma_{th} = 0.1$ semitones $\times$ second. Bottom: interval threshold $\delta_{th} = 0.5$ semitones).

scribed in Section VII-A, the transcription scheme develop by Gomez and Bonada in [4] and the HMM based approach described in Section VII-B.

The first measure is the *Raw pitch accuracy* and the following two are *Voicing recall* and *Voicing false alarm*, the rest correspond to the Note-rate (NR) and the Frame-rate (FrR) measures for each category: Non-detected, Spurious, Split, Merged, Correctly detected and Badly detected. The results have been obtained using our scheme with the parameters described in previous sections, specifically: median filtering of the F0 curve, maximum chroma gap between consecutive frames in a chroma contour $\rho_{th} = 1$ semitone, interval threshold to perform note segmentation $\delta_{th} = 0.5$ semitones and hysteresis with cumulative pitch deviation (area) threshold $\Gamma_{th} = 0.1$ semitones $\times$ seconds.

As shown in Fig. 12, the proposed system developed for singing transcription outperforms the baseline approach, and attains similar results to previous state of the art schemes. Regarding the pitch accuracy, which is directly related to the correct estimation of notes' pitch, our approach performs better than the rest of approaches. In the case of voicing, both our approach and Gómez & Bonada have similar performances. In addition, when compared with the HMM-based approach, our approach and Gómez & Bonada have a better voicing recall, but a worse voicing false alarm. This is so because the voicing estimation is more restrictive in the HMM-based method. Note that, in spite of this fact, the rate of spurious notes is slightly higher in the HMM-based method.

Regarding the Note-rate and Frame-rate of correctly detected notes, the performances of all the state-of-the-art systems are similar between them (and better than the baseline, as expected). However, we found statistically significant differences between the HMM-based approach and Gómez & Bonada in terms of CD note-rate performance (frame-rate score differences are not significant). On the other hand, note the good behavior of the baseline method with respect to the other methods regarding the rate of merged (M) notes. This fact can be considered a drawback of the hysteresis cycle introduced by our system. We have observed that this issue is especially noticeable when consecutive vowels are analyzed or in the presence of voiced consonants (e.g. 'lalala'), in this case, all the notes are often merged. However, our approach is robust against vibrato or other type of oscillations around a constant pitch (see description in Section IV-B). The statistical significance of all the mentioned differences has been verified using Student's t-test.

D. Influence of the Parameters on the System Performance

We have studied the influence of three main parameters on the behavior of the system: interval threshold δ_{th} (see eq. (5)), cumulative pitch deviation (area) threshold Γ_{th} and, α (see eq. (6)). For the case of δ_{th} and Γ_{th}, we have analyzed the evolution of each evaluation measure in the note-rate category along each parameter. An illustration of the results obtained is shown in Fig. 13. It can be observed that the highest CD note-rate is obtained for a confidence interval $\delta_{th} = 0.5$ semitones and an area threshold $\Gamma_{th} = 0.1$ semitones $\times$ second.

Note that our system tends to merge notes rather than split them. However, for low values of δ_{th} and Γ_{th}, the number of split notes increases due to the implicit trade-off between merged notes (indicated by measure M) and split notes (measure S) of our approach.

Finally, also the effect of the parameter α has been studied. In this case, the influence of α on the global performance has not been found to be as important as the parameters previously considered. However, we found that $\alpha \in [0.2, 0.4]$ produces the highest correctly detected note (CD) rate, with no differences if the parameter is maintained within this range, in the experiments performed. Conversely, if $\alpha < 0.2$ or $\alpha > 0.4$, the system accuracy (CD) slightly decreases. In our case, we have chosen the central value of the interval: $\alpha = 0.3$.

VIII. Conclusions

The SiPTH system for singing note segmentation and labeling has been presented. This scheme uses the Yin algorithm [21] with specific parameters and a post-processing stage to extract three different curves: pitch, power and aperiodicity. This information is used to perform a first segmentation by estimating stable chroma contours. The concept of chroma contour is introduced in this paper as an octave-independent version of the pith contour.

A simple set of descriptors is computed from each stable chroma contour to distinguish between voiced/unvoiced regions. The voicing F-measure attained by the proposed approach on a varied set of recordings is around 97%.

After the voiced regions have been identified, a novel interval-based segmentation method has been applied to define note segments. Note changes are identified when strong and/or sustained pitch deviations are found. Thus, a pitch-time hysteresis effect has been considered to avoid the detection of weak and/or short pitch variations as false note changes.

A detailed evaluation methodology has been proposed which involves an original algorithm to recognize the different types of transcription errors. The proposed error measures can be considered an extension of the ones proposed by Ryynänen in [11] and they have been inspired by the evaluation methodology proposed in the MIREX contest for onset detection [43]. The evaluation methodology proposed is more complete than previous ones and it can be applied to further singing transcription systems to thoroughly study their performance at note and frame level.

After comparing the results obtained by the proposed scheme against the performance of a baseline scheme defined in this manuscript, a transcription algorithm developed by Gomez and Bonada [4] and a HMM-based method inspired by [3], [11], it can be concluded that the system developed introduces remarkable improvements with respect to the baseline, especially in the correctly transcribed Note-rate, Frame-rate and raw pitch accuracy measures. Also, our system achieves similar performance to the one attained by the HMM-based scheme implemented and to the algorithm presented in [4], while using a totally different strategy. On the other hand, further research is needed to improve the rate of merged notes, which is higher than with the baseline approach mainly because of note changes detected on vowels or voiced consonants.

Acknowledgment

The authors are grateful to E. Gomez for providing the results of the scheme developed in [4] for comparison.

References

[1] J. Plantinga and L. J. Trainor, "Memory for melody: Infants use a relative pitch code," *Cognition*, vol. 98, no. 1, pp. 1–11, 2005.

[2] M. Lesaffre, M. Leman, B. De Baets, and J. Martens, "Methodological considerations concerning manual annotation of musical audio in function of algorithm development," in *Proc. 5th Int. Conf. Music Inf. Retrieval ISMIR*, 2004, pp. 64–71.

[3] M. Ryynänen, "Singing transcription," in *Signal Processing Methods for Music Transcription*, A. Klapuri and M. Davy, Eds. New York, NY, USA: Springer Science + Business Media LLC, 2006, pp. 361–390.

[4] E. Gómez and J. Bonada, "Towards computer-assisted flamenco transcription: An experimental comparison of automatic transcription algorithms as applied to a cappella singing," *Comput. Music J.*, vol. 37, no. 2, pp. 73–90, 2013.

[5] B. Pardo, J. Shifrin, and W. Birmingham, "Name that tune: A pilot study in finding a melody from a sung query," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 55, no. 4, pp. 283–300, 2004.

[6] C. De La Bandera, A. M. Barbancho, L. J. Tardón, S. Sammartino, and I. Barbancho, "Humming method for content-based music information retrieval," in *Proc. 12th Int. Soc. Music Inf. Retrieval Conf. ISMIR*, 2011.

[7] D. M. Howard, G. Welch, J. Brereton, E. Himonides, M. Decosta, J. Williams, and A. Howard, "WinSingad: A real-time display for the singing studio," *Logopedics Phoniatrics Vocology*, vol. 29, no. 3, pp. 135–144, 2004.

[8] C. Dittmar, H. Gromann, E. Cano, S. Grollmisch, H. M. Lukashevich, and J. Abeer, "Songs2see and globalmusic2one: Two applied research projects in music information retrieval at Fraunhofer IDMT," in *Proc. 7th Int. Conf. Exploring Music Contents (CMMR'10)*, S. Ystad, M. Aramaki, R. Kronland-Martinet, and K. Jensen, Eds. New York, NY, USA: Springer, 2010, pp. 259–272, vol. 6684 of Lecture Notes in Computer Science.

[9] "Singstar game, by Sony Computer Entertainment Europe," [Online]. Available: http://www.singstar.com/ 2004

[10] V. Bharathi, A. A. Abraham, and R. Ramya, "Vocal pitch detection for musical transcription," in *Proc. Int. Conf. Signal Process. Commun. Comput. Network. Technol. ICSCCN*, 2011, pp. 724–726.

[11] M. Ryynänen and A. Klapuri, "Modelling of note events for singing transcription," in *Proc. ISCA Tutorial Res. Workshop Statist. Percept. Audio Process. SAPA*, Jeju, Korea, Oct. 2004.

[12] E. Molina, I. Barbancho, E. Gomez, A. Barbancho, and L. Tardon, "Fundamental frequency alignment vs. note-based melodic similarity for singing voice assessment," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2013, pp. 744–748.

[13] R. J. McNab, L. A. Smith, and I. H. Witten, "Signal processing for melody transcription," in *Proc. 19th Australasian Comput. Sci. Conf.*, 1996, vol. 18, no. 4, pp. 301–307.

[14] G. Haus and E. Pollastri, "An audio front end for query-by-humming systems," in *Proc. 2nd Int. Symp. Music Inf. Retrieval (ISMIR)*, 2001, pp. 65–72.

[15] L. P. Clarisse, J. P. Martens, M. Lesaffre, B. D. Baets, H. D. Meyer, and M. Leman, "An auditory model based transcriber of singing sequences," in *Proc. 3rd Int. Conf. Music Inf. Retrieval ISMIR*, 2002, pp. 116–123.

[16] T. De Mulder, J.-P. Martens, M. Lesaffre, M. Leman, B. De Baets, and H. De Meyer, "An auditory model based transcriber of vocal queries," in *Proc. 4th Int. Conf. Music Inf. Retrieval ISMIR*, 2003.

[17] W. Krige, T. Herbst, and T. Niesler, "Explicit transition modelling for automatic singing transcription," *J. New Music Res.*, vol. 37, no. 4, pp. 311–324, 2008.

[18] J. J. Mestres, J. B. Sanjaume, M. De Boer, and A. L. Mira, "Audio recording analysis and rating," U.S. Patent 8,158,871, Apr. 17, 2012.

[19] S. Vaseghi, *Advanced signal processing and digital noise reduction*. New York, NY, USA: Wiley, 1996, vol. 46.

[20] T. Tolonen and M. Karjalainen, "A computationally efficient multipitch analysis model," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 6, pp. 708–716, Nov. 2000.

[21] A. De Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Amer.*, vol. 111, no. 4, p. 1917, 2002.

[22] I. Mayergoyz, "Mathematical models of hysteresis," *IEEE Trans. Magnetics*, vol. MAG-22, no. 5, pp. 603–608, Sep. 1986.

[23] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, Nov. 2009.

[24] E. Gómez, A. Klapuri, and B. Meudic, "Melody description and extraction in the context of music content processing," *J. New Music Res.*, vol. 32, no. 1, pp. 23–40, 2003.

[25] W. Hess, *Pitch determination of speech signals*. Berlin, Germany: Springer Verlag, 1983.

[26] T. Viitaniemi, A. Klapuri, and A. Eronen, "A probabilistic model for the transcription of single-voice melodies," in *Proc. Finnish Signal Process. Symp. (FINSIG'03)*, 2003, pp. 59–63.

[27] G. E. Poliner, D. P. W. Ellis, A. F. Ehmann, E. Gomez, S. Streich, and B. Ong, "Melody transcription from music audio: Approaches and evaluation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 4, pp. 1247–1256, May 2007.

[28] L. Rabiner and R. Schafer, *Digital processing of speech signals*, ser. Prentice-Hall Series in Signal Processing No. 7. Englewood Cliffs, NJ, USA: Prentice-Hall, 1978.
[29] A. De Cheveigné, Matlab implementation of YIN algorithm [Online]. Available: http://audition.ens.fr/adc/sw/yin.zip Feb. 2012
[30] L. Rabiner, M. Sambur, and C. Schmidt, "Applications of a nonlinear smoothing algorithm to speech processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-23, no. 6, pp. 552–557, Dec. 1975.
[31] J. Rouat, Y. C. Liu, and D. Morissette, "A pitch determination and voiced/unvoiced decision algorithm for noisy speech," *Speech Commun.*, vol. 21, no. 3, pp. 191–207, 1997.
[32] J. Salamon and E. Gómez, "Melody extraction from polyphonic music signals using pitch contour characteristics," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 6, pp. 1759–1770, Aug. 2012.
[33] Y. Xu and X. Sun, "Maximum speed of pitch change and how it may relate to speech," *J. Acoust. Soc. Amer.*, vol. 111, p. 1399, 2002.
[34] "Documentation about class J48 of WEKA tool," [Online]. Available: http://weka.sourceforge.net/doc/weka/classifiers/trees/J48.html Feb. 2012.
[35] J. Quinlan, *C4. 5: programs for machine learning*. Burlington, MA, USA: Morgan Kaufmann, 1993, vol. 1.
[36] Y. Mansour, "Pessimistic decision tree pruning based on tree size," in *Proc. 14th Int. Conf. Mach. Learn.*, 1997, pp. 195–201.
[37] E. Pollastri, "Some considerations about processing singing voice for music retrieval," in *Proc. 3rd Int. Conf. Music Inf. Retrieval ISMIR*, 2002.
[38] J. Bednar and T. Watt, "Alpha-trimmed means and their relationship to median filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, no. 1, pp. 145–153, Feb. 1984.
[39] E. Benetos and S. Dixon, "Polyphonic music transcription using note onset and offset detection," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process. (ICASSP)*, 2011, pp. 37–40.
[40] M. M. Association *et al.*, "MIDI 1.0 detailed specification," The Int. MIDI Association, Los Angeles, CA, USA, 1998.
[41] K. Schutte, *MIDI toolkit for Matlab*, 2012 [Online]. Available: http://www.kenschutte.com/midi
[42] J. Salamon, J. Serra, and E. Gómez, "Tonal representations for music retrieval: From version identification to query-by-humming," *Int. J. Multimedia Inf. Retrieval*, pp. 1–14, 2013.
[43] J. S. Downie, *MIREX contest website*, 2013 [Online]. Available: http://www.music-ir.org/mirex
[44] A. P. Klapuri, A. J. Eronen, and J. T. Astola, "Analysis of the meter of acoustic musical signals," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 1, pp. 342–355, Jan. 2006.

Emilio Molina received his degree in telecommunications engineering from the University of Málaga, Spain, in 2011. In 2012, he obtained the Professional Degree of classic piano from the Conservatori del Liceu, Barcelona, Spain, and his M.Sc. in sound and music computing from the Universitat Pompeu Fabra, Barcelona, Spain, in 2013. He was awarded with the Best Final Year Project award from University of Málaga in 2007 and he was nominated as finalist for the Best Final Year Project Award by the Official National Telecommunications Engineering Board in 2013. Currently, he is a Ph.D. candidate at the Application of Information and Communication Technologies Research Group. His main research topic is the automatic analysis and processing of audio signals and applications.

Lorenzo J. Tardón received his degree in telecommunications engineering from the University of Valladolid, Spain, in 1995 and his Ph.D. degree from the Polytechnic University of Madrid, Spain, in 1999. In 1999 he worked for ISDEFE on air traffic control systems at the Madrid-Barajas Airport and for Lucent Microelectronics on systems management. Since November 1999, he has been with the Department of Communications Engineering, University of Málaga, Spain. He is currently the head of the Application of Information and Communications Technologies (ATIC) Research Group. He has been the main researcher of different projects on audio and music analysis. He is a member of several international journal committees on communications and signal processing. In 2011, he was awarded the Premio Málaga de Investigación by the Academies Bellas Artes de San Telmo and Malagueña de Ciencias. His research interests include serious games, audio signal processing, digital image processing, and pattern analysis and recognition.

Ana M. Barbancho received her degree in telecommunications engineering and her Ph.D. degree from University of Málaga, Spain, in 2000 and 2006, respectively. In 2001, she received her degree in solfeo teaching from the Málaga Conservatoire of Music. Since 2000, she has been with the Department of Communications Engineering, University of Málaga, as an Assistant and then Associate Professor. Her research interests include musical acoustics, digital signal processing, and mobile communications. Dr. Barbancho was awarded with the Second National University Prize to the Best Scholar 1999/2000 by the Spanish Ministry of Education in 2000 and with the Extraordinary Ph.D. Thesis Prize by ETSI Telecomunicación of University of Málaga in 2007.

Isabel Barbancho (SM'10) received her degree in telecommunications engineering and her Ph.D. degree from the University of Málaga, Spain, in 1993 and 1998, respectively, and her degree in piano teaching from the Málaga Conservatoire of Music in 1994. Since 1994, she has been with the Department of Communications Engineering, as an Assistant and then Associate Professor. During 2013, she was a Visiting Scholar at the University of Victoria, Victoria, BC, Canada. She has been the main researcher on several research projects on polyphonic transcription, optical music recognition, music information retrieval, and intelligent content management. Her research interests include musical acoustics, signal processing, multimedia applications, audio content analysis, and serious games. Dr. Barbancho received the Severo Ochoa Award in Science and Technology, Ateneo de Málaga-UMA in 2009 and the Premio Málaga de Investigación 2011 Award from the Academies Bellas Artes de San Telmo and Malagueña de Ciencias.