Singing Information Processing: Techniques and Applications

Emilio Molina Martínez

Tesis Doctoral / PhD Thesis

Programa de Doctorado en Ingeniería de Telecomunicación
Escuela Técnica Superior de Ingeniería de Telecomunicación
Universidad de Málaga, 2017

Tutor
Lorenzo José Tardón García

Directores
Lorenzo José Tardón García
Ana María Barbancho Pérez

Los directores de esta tesis doctoral, Dr. Lorenzo José Tardón García y Dra. Ana María Barbancho Pérez, acreditan que habiendo revisado esta versión de la tesis, es apta para ser entregada al tribunal autorizado.

Dr. Lorenzo José Tardón García

Dra. Ana María Barbancho Pérez

Málaga, a 24 de abril de 2017

Abstract

Singing is an essential component of music in all human cultures around the world, since it is an incredibly natural way of musical expression. Consequently, digital processing of singing has a major impact on society from the viewpoints of industry, culture and science. However, unlike speech processing, singing processing is a rather immature research field, and many challenges associated to it are not solved yet for real-world purposes. In such context, this dissertation contributes with a varied set of novel techniques and applications related to singing information processing, together with a review of the background related to each of them.

First, we analyze the importance of pitch tracking in query-by-singing-humming, since this relationship had not been deeply studied in the past. For this analysis, a comparative study of state-of-the-art pitch trackers is carried out. The achieved results show that [Boersma, 1993] (with a not-obvious parameters tuning) and [Mauch, 2014], have a great performance for query-by-singing-humming due to the smoothness of the resulting F0 contour.

In addition, a novel singing transcription algorithm based on a hysteresis process on the pitch-time curve is proposed, together with an evaluation framework for singing transcription. The interest of our singing transcription algorithm is that it achieves state-of-the-art error rates using a simple approach. The proposed evaluation framework, on the other hand, is a powerful resource for future researchers in singing transcription, and it is a valuable step forward towards a better definition of the problem and a better evaluation of the proposed solutions.

Moreover, this thesis also presents a method for singing skill evaluation. It uses dynamic time warping to align the user's performance and a reference in order to provide a score for pitch intonation and rhythm accuracy. The evaluation shows a high correlation between the scores provided by our system and the scores provided by a group of expert musicians.

Besides, we present a method to produce realistic intensity variations in singing voice. The proposed approach is based on a parametric model of spectral envelope, and it improves the perceived realism of intensity variation when compared with other commercial software, such as Melodyne and Vocaloid. The drawback of the chosen approach is that it requires manual intervention, but the achieved results provide relevant insights towards realistic automatic intensity transformation in singing voice for real-world purposes.

Finally, we propose a novel method to reduce the dissonance of isolated recorded chords. It is based on multiple F0 analysis, and a frequency shifting of sinusoidal components to produce an in-tune sound. The evaluation has been performed by a set of trained musicians, showing a clear improvement of the perceived consonance after the proposed transformation.

Resumen

La voz cantada es una componente esencial de la música en todas las culturas del mundo, ya que se trata de una forma increíblemente natural de expresión musical. En consecuencia, el procesado automático de voz cantada tiene un gran impacto desde la perspectiva de la industria, la cultura y la ciencia. En este contexto, esta Tesis contribuye con un conjunto variado de técnicas y aplicaciones relacionadas con el procesado de voz cantada, así como con un repaso del estado del arte asociado en cada caso.

En primer lugar, se han comparado varios de los mejores estimadores de tono conocidos para el caso de uso de recuperación por tarareo. Los resultados demuestran que [Boersma, 1993] (con un ajuste no obvio de parámetros) y [Mauch, 2014], tienen un muy buen comportamiento en dicho caso de uso dada la suavidad de los contornos de tono extraídos.

Además, se propone un novedoso sistema de transcripción de voz cantada basada en un proceso de histéresis definido en tiempo y frecuencia, así como una herramienta para evaluación de voz cantada en Matlab. El interés del método propuesto es que consigue tasas de error cercanas al estado del arte con un método muy sencillo. La herramienta de evaluación propuesta, por otro lado, es un recurso útil para definir mejor el problema, y para evaluar mejor las soluciones propuestas por futuros investigadores.

En esta Tesis también se presenta un método para evaluación automática de la interpretación vocal. Usa alineamiento temporal dinámico para alinear la interpretación del usuario con una referencia, proporcionando de esta forma una puntuación de precisión de afinación y de ritmo. La evaluación del sistema muestra una alta correlación entre las puntuaciones dadas por el sistema, y las puntuaciones anotadas por un grupo de músicos expertos.

Por otro lado, se presenta un método para el cambio realista de intensidad de voz cantada. Esta transformación se basa en un modelo paramétrico de la envolvente espectral, y mejora sustancialmente la percepción de realismo al compararlo con software comerciales como Melodyne o Vocaloid. El inconveniente del enfoque propuesto es que requiere intervención manual, pero los resultados conseguidos arrojan importantes conclusiones hacia la modificación automática de intensidad con resultados realistas.

Por último, se propone un método para la corrección de disonancias en acordes aislados. Se basa en un análisis de múltiples F0, y un desplazamiento de la frecuencia de su componente sinusoidal. La evaluación la ha realizado un grupo de músicos entrenados, y muestra un claro incremento de la consonancia percibida después de la transformación propuesta.

Acknowledgments

Throughout my last years, many people have been around me, what makes me feel happy and lucky. Some of them have positively influenced in my PhD in one way or another, and I would like to mention them in order to let them know.

First, I would like to mention ATIC research group in University of Málaga. It has been my home for three years, and we have shared a lot of experiences. I would like to thank Lorenzo, Isabel and Ana María for believing in me and trusting me from the very beginning, since it has been a great source of motivation throughout these years. Thanks also to Carles, a great workmate and friend, for our everyday lunch in law department, and for everything we have shared. Thanks also to Alejandro, Jesús, Panos, Najera... for those conversations and coffees at the lab. Thanks to all students that participated in my music production workshop, it was a really nice experience.

Along all these years, I have also been in contact with my previous colleges from MTG in Universitat Pompeu Fabra in Barcelona. We have shared a lot of moments, knowledge, code, etc. So many people that have inspired me somehow: Juanjo, Jan, María, Marius, John, Tim, Emilia... Thanks for being around me.

My family has also a lot to do with this story. My parents always showed me that passion and work should be together, and their support and love have been essential for me. They have worked hard for making my life easy, so thanks.

Finally, thanks to Mabel, my soulmate. You have always encouraged me to do my best in every step of my life because you understand what it means for me, and this PhD is a great example of it. You are great, that's why I love you. Thanks!

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Singing is an essential component of music in all human cultures around the world, since it is an incredibly natural way of musical expression. In fact, the expression of feelings through singing is considered to be far older than the expression of thoughts through speech [Jespersen, 1922], and it is agreed to be present even in other animal species [Wallin and Merker, 2001]. In the case of western music, the role of singing voice throughout the history has varied. During the medieval period and the Renaissance, vocal music was especially popular in religious contexts. After 17th century, the birth of opera leads to a new context for singing voice, in which virtuoso solo singers are accompanied by an orchestra in a theatrical context. In 20th century, recording technologies and audio amplification contributed to the appearance of non-operatic, speech-like singing styles and new expressive resources (e.g. whispering). Nowadays, singing has a clear leading role in most modern music styles (e.g. pop).

Consequently, digital processing of singing has a major impact on society from the viewpoints of industry, culture and science due to its countless applications. For instance, Music Information Retrieval (MIR) systems can greatly benefit from singing analysis since vocals convey highly relevant information about the audio content (expressiveness, singer, style, lyrics, etc). In addition, singing is an accessible and intuitive way of human-machine interaction, so it is particularly suitable for games, composition tools, query-by-humming-singing systems or educational applications. In the context of education, singing is essential for the development of general music skills [Welch et al., 1988]. Moreover, singing typically provides important clues about our music cognition, so singing analysis can be also useful from a scientific perspective to better understand our mental processes.

However, unlike speech processing, singing processing is a rather immature research field, and the challenges associated to it are often underestimated. In many singing-related research topics, there is a lack of good evaluation tools (datasets, software...),

and most of the classical problems are far to be solved for real-world purposes: note transcription (not even in a monophonic context [Gómez et al., 2013]), realistic timbre modifications [Molina et al., 2014c], lyric transcription and synchronization [Goto, 2014], etc. Indeed, many approaches that work for specific musical instruments, are not suitable for singing voice (e.g. note transcription [Gómez et al., 2013]). The difficulty of singing processing resides in the high variability of singing signals, as they are strongly affected by a sort of aspects: singer (gender, timbre, training, age...), music style (e.g. rap is completely different from opera), lyrics (e.g. determining the note-segmentation strategy), etc. To sum up, there is a clear need of further research to overcome such singing-related challenges.
Fortunately, the research community is increasingly interested in singing analysis and processing [Mauch et al., 2015b]. Indeed, the area of research called *Singing Information Processing* has been recently defined [Goto et al., 2010], and every year, new valuable approaches and resources are available (e.g. Tony tool[1] for note-wise annotation).

Scientific Context: Singing Information Processing

The area of research called *Singing Information Processing* was initially proposed by [Goto et al., 2010], and it is defined as "music information processing for singing voices". More recently, [Goto, 2014] presents a review of this research area through a collection of organized applications (which are summarized in Figure 1.1), some of which are described in following paragraphs.
One of the classical research problems is *singing synthesis* [Cook, 1991]. This topic has been actively studied in the second half of 1980s and throughout 1990s [Cook, 1996]. More recently, corpus-based synthesis methods based on the concatenation of samples have been proposed [Bonada and Serra, 2007] [Schwarz, 2007]. One of the most successful applications of corpus-based synthesis is Vocaloid [Kenmochi and Ohshita, 2007], which has become very popular (especially in Japan).
Lyric transcription and synchronization, on the other hand, aim to give computer the ability to understand lyrics in singing voice. This challenging problem can be seen as the singing version of automatic speech recognition (ASR), which is a classic research problem, and it is considered unsolved for generic singing with accompaniment. If the text of the lyrics is known in advance, the problem is called lyrics synchronization. Research into lyric synchronization can be divided into two categories: that using no forced-alignment (e.g. [Kan et al., 2008]), and that using forced alignment (e.g. [Fujihara et al., 2011]).
Some other applications are based on voice timbre analysis and processing. Vocal timbre is an essential element of singing, since it conveys information about the

[1] http://isophonics.net/tony

singer, the vocal quality, expressive aspects, etc. Many applications have been proposed based on vocal timbre processing: voice conversion [Toda et al., 2007], singer identification [Zhang, 2003], emotion recognition [Kanato et al., 2014], etc.
In addition, some music information retrieval systems are directly based on singing voice, such as query-by-singing-humming applications, which use short singing or humming excerpts as a search key in a collection of songs. A variety of successfull methods for QBH have been proposed, mostly based on a mix F0 contour and note-wise matching [Wang et al., 2008] [Li et al., 2008], but also based on lyrics matching [Wang et al., 2010] [Suzuki et al., 2007] or MFCC and formants matching [Duda et al., 2007]. A less frequent variant of music information retrieval based on singing voice is based on "voice percussion" [Nakano et al., 2005].
Singing transcription is a relevant and challenging research problem that refers to the automatic conversion of a recorded singing signal into a symbolic representation (e.g. a MIDI file) by applying signal-processing methods [Ryynänen, 2006]. It can be used as an intermediate stage for QBSH [Pardo et al., 2004], for singing assessment [Dittmar et al., 2010], or directly applied to computational tools for musicians (e.g. ScoreCloud[2]).
In addition, some successful systems are commercially available for pitch contour modification (e.g. Melodyne[3] or Auto-tune[4]). This kind of systems are massively used nowadays in recording studios to correct intonation errors of singers.
Finally, automatic singing skill evaluation, or automatic singing assessment, has been addressed in a variety of works [Rossiter and Howard, 1996] [Howard et al., 2004] [Saino et al., 2006] (see [Molina, 2012] for a review). In general, all these systems focus on intonation assessment with visually attractive real-time feedback. Songs2See [Grollmisch et al., 2011] is a recent and representative example of the state of the art. These type of applications have been applied in two main fields: entertainment (mainly games) and education. Perhaps, the most famous game related to automatic singing skill evaluation is Singstar[5], which has become popular in the last years.

Topics Addressed in this thesis

Given the relevance, and the growing interest of *Singing Information Processing* (as mentioned previously in this Section), this thesis addresses several specific topics related to such a broad field, which are described in the following paragraphs.
We analyze the importance of pitch tracking in query-by-singing-humming, since this relationship has not been deeply studied in the past. In this thesis, the term

[2] http://scorecloud.com
[3] www.celemony.com
[4] www.antarestech.com
[5] www.singstar.com

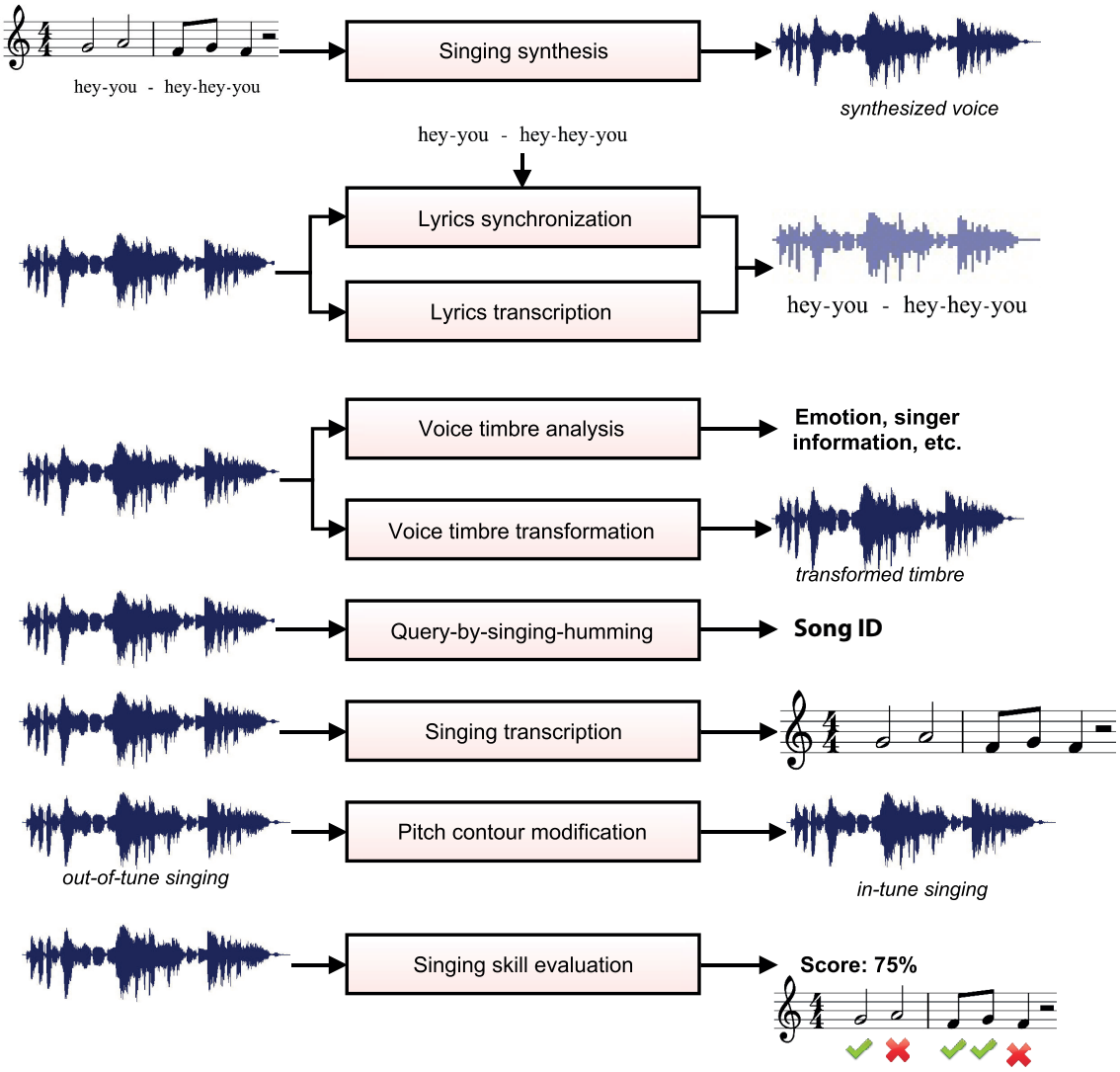

Figure 1.1: Schema of *Singing Information Processing* applications, showing the kind of input and output in each case (audio signal, symbolic, etc.).

pitch is not used to refer to the perceptual feature, but to the fundamental frequency (F0) of a signal; therefore, the terms *pitch* and F0 are used indistinctly. For such analysis, we carry out a comparative study of state-of-the-art pitch trackers in the context of query-by-singing-humming. This study is described in more detail in [Molina et al., 2014d] (Section 3.1).
In addition, a novel singing transcription algorithm based on a hysteresis process on the pitch-time curve is proposed (published in [Molina et al., 2015]), together with an evaluation framework for singing transcription (published in [Molina et al., 2014b]). These contributions are summarized in Section 3.2.
Moreover, a method for singing skill evaluation (or singing assessment) is presented (see Section 3.3). This method has been published in [Molina et al., 2013], and it uses dynamic time warping to align the user's performance and a reference in order to provide a score for pitch intonation and rhythm accuracy.
Besides, we present a study about the changes in spectral envelope when vocal intensity varies, together with a method to produce realistic intensity variations in singing voice. This method has been published in [Molina et al., 2014c], and it is summarized in Section 3.4.
Finally, in relation with the problem of pitch contour modification, we propose a novel method to reduce the dissonance of recorded chords (vocal or instrumental) by processing its sinusoidal component. It is based on a multiple F0 analysis, and a frequency shifting of sinusoidal component to produce an in-tune output sound. It has been published in [Molina et al., 2014a], and it is summarized in Section 3.5.

1.1 Research Goals

The research goals of this PhD involve both, techniques and applications, related to the field of Singing Information Processing. These goals are:

- Review the state-of-the-art of the research field *Singing Information Processing*. For it, the most relevant research problems, and the key references for each of them must be identified and understood. This review must be especially deep in the main topics addressed by this thesis: F0 and note tracking, automatic singing assessment and voice timbre processing.

- Develop a singing transcription method with state-of-the-art performance. This challenging goal can be broken down into several sub-goals:

  - Define a clear research methodology to address the problem of singing transcription, since the literature does not provide a clear one. This sub-goal involves deciding what kind of annotated data is needed, what

evaluation metrics are relevant and what are the available state-of-the-art methods to compare with.

- Gather a dataset of monophonic singing audio with note-level annotations.
- Gather, or implement, state-of-the-art singing transcription methods to compare with.
- Build a publicly available evaluation framework tool for singing transcription.
- Investigate and develop a novel method for automatic note transcription in singing voice.

- Investigate and develop a novel system for automatic singing assessment based on pitch contour and note-wise comparison with respect to a target reference. This goal also involves gathering singing performances with annotations by music teachers, which will be used for evaluation.

The previous goals have many aspects in common, since they mainly involve audio analysis techniques. However, the knowledge about singing information processing achieved along our investigation has also led us to two extra goals involving sound transformation:

- Investigate and develop a system to model timbre changes produced in singing voice when intensity varies. This goal also involves developing a software tool to visualize and annotate the spectral envelope of a dataset of sung vowels, which will be used for investigation.

- Investigate and develop a system to process out-of-tune vocal chords in order to make them sound in-tune. This goal also involves gathering an evaluation dataset and carrying out a listening test with musicians to assess the performance of the proposed method.

1.2 Thesis Outline

This thesis consists of four main chapters. In Chapter 1, we introduce the motivation and scientific context, together with the research goals and the outline of this thesis. Chapter 2 presents an overview on several research areas that are relevant for this thesis. These areas are: singing voice production (Section 2.1), pitch estimation (Section 2.2), singing transcription (Section 2.3), dynamic time warping (Section 2.4), automatic singing assessment (Section 2.5), timbre processing (Section 2.6), and spectral modeling synthesis (Section 2.7). In Chapter 3 we summarize

the results of this thesis organized by topic: comparative analysis of F0 trackers for query-by-singing-humming (Section 3.1), singing transcription (Section 3.2), automatic singing assessment (Section 3.3), timbre analysis and processing (Section 3.4) and dissonance reduction in polyphonic audio (Section 3.5). Chapter 4 draws some general conclusions about the various aspects covered in previous chapters (Section 4.1), presents an enumeration of all scientific and technical contributions of this thesis (Section 4.2), and presents some suggestions for future research (Section 4.3). Finally, Appendix A enumerate all relevant web resources mentioned along this thesis, and Appendix B includes the published papers in the context of this thesis.

CHAPTER 2

Background and Related Work

In this chapter, we provide the background and previous work related to the various contributions of this thesis. In Figure 2.1 the relationship between such contributions (see Section 1.4) and the topics covered in this chapter is described.

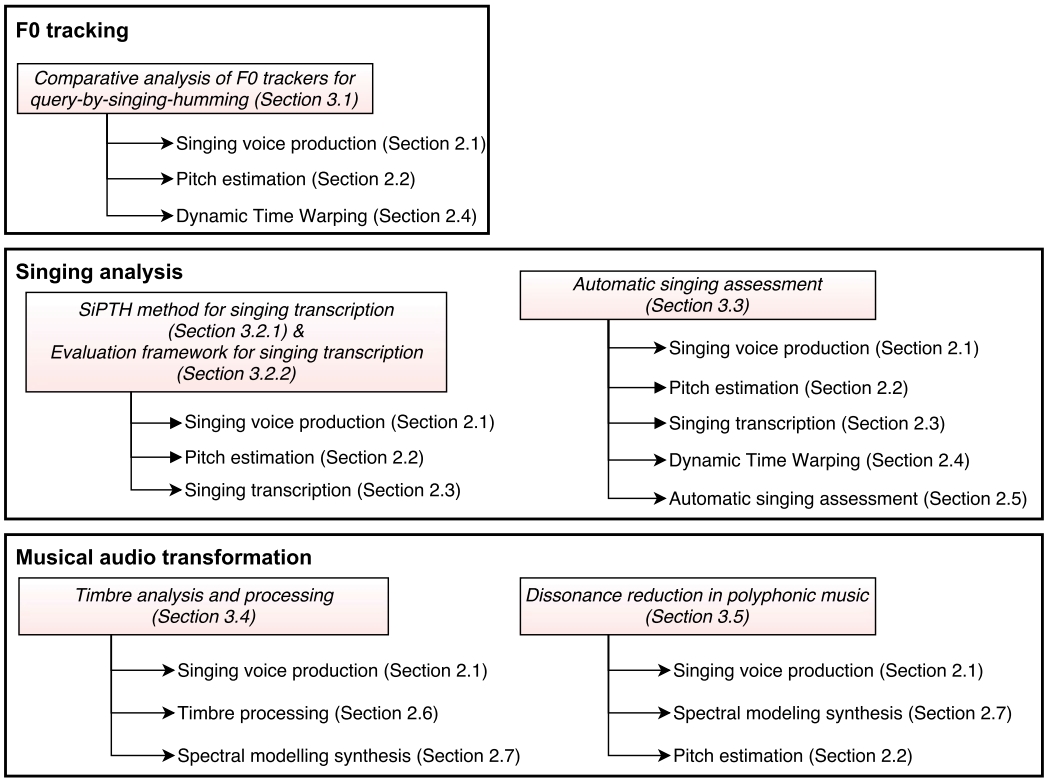

Figure 2.1: Relationship between the contributions of this thesis (see Section 4.2) and the background presented in this chapter.

This chapter is organized as follows: Section 2.1 provides a general overview about voice production, since these scientific concepts are relevant for the rest of sections of this dissertation. Section 2.2 describes the state-of-the-art in pitch estimation, for both monophonic and polyphonic contexts. In Section 2.3, we present a review on note-level transcription of singing voice. Section 2.4 describes the technique of Dynamic Time Warping (DTW), since it is the base of our contribution on automatic singing assessment. Then, Section 2.5 presents an overview about the state-of-the-art on automatic singing assessment. In Section 2.6, a general overview about voice timbre processing is presented (source-filter model, spectral envelope extraction, formant analysis and timbre-related features). Finally, Section 2.7 presents spectral modeling synthesis (SMS) technique, which is implemented in our methods for timbre processing, and for polyphonic transformations.

2.1 Singing Voice Production

In this section, we describe some general aspects about the anatomy of the singing voice (Section 2.1.1), and we present some important differences between speech and singing (Section 2.1.2). This background is necessary to understand many acoustical characteristics of the singing voice signal, which is the main object of study in this thesis.

2.1.1 Anatomy of the Human Voice

According to [Sundberg, 1987], the singing voice can be defined as "the sounds produced by the voice organ and arranged in adequate musical sounding sequences". The voice organ includes the lungs, the larynx, the pharynx, the nose and the mouth (see Figure 2.2.a).
The main function of the lungs (in speech and singing) is to produce an excess of air pressure, which generates an airstream [Sundberg, 1977]. The air passes through the glottis, a space at the base of the larynx between the two vocal folds. The front end of each vocal fold is attached to the thyroid cartilage, or Adam's apple. The back end of each is attached to one of the two arytenoid cartilages, which are mobile, moving to separate the folds (for breathing), to bring them together and to stretch them. The vocal folds are at the bottom of the tube-shaped larynx, which fits into the pharynx, the wider cavity that leads from the mouth to the esophagus. When the airstream is periodically chopped by the oscillation of the vocal folds, an acoustic signal is produced (called voice source). The roof of the pharynx is the velum, or soft palate, which in turn is the door to the nasal cavity. When the velum is raised, the passage to the nose is closed and air moves out through the mouth.
The larynx, the pharynx and the mouth together constitute the vocal tract, which

acts as a resonant chamber. The shape of the tract is determined by the positions of the articulators: the lips, the jaw, the tongue and the larynx, and they shape acoustically the voice source. The frequencies enhanced by the vocal tract are called formants. The final step is the acoustic radiation through the lips.

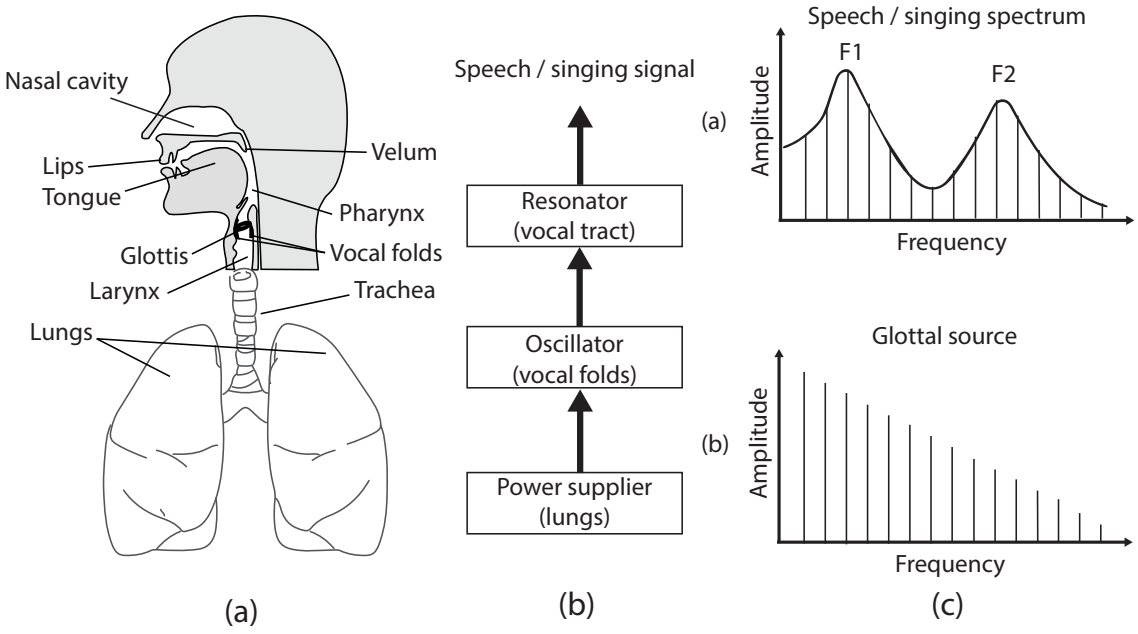

Figure 2.2: (a) Anatomy of the voice organ. (b) Simplified model of the voice organ. (c) Spectrum of the voice source (or glottal source), and spectrum of speech/singing voice after vocal tract filtering (note the effect of formants on the acoustic shaping of the sound).

As shown in Figure 2.2.b, the voice production process can be modeled with three major units: the power supplier (the lungs), an oscillator (the vocal folds) and a resonator (the vocal tract). The power supplier directly affects the energy of the sound produced. The oscillator produces a complex tone (voice source) at certain frequency, whose partials decrease uniformly with frequency at the rate of about 12 decibels per octave (Figure 2.2.c). This slope is more step in soft speech, though. Finally, this signal is shaped by the resonator, which can be modeled as an all-pole filter (see Section 2.6.2.1).

A deeper description of voice production principles can be found in Sundberg's and Titze's well-known works [Sundberg, 1977] [Sundberg, 1987] [Titze, 2000].

2.1.2 Singing vs. Speech

Speech is the most common use of human voice in all cultures, and therefore most of the research studies about the human voice in the literature are related to it. The case of singing is commonly viewed as a special case of speech, but there are some profound differences between them (summarized in [Cook, 1991] and [Kim, 2003]). In this section, we enumerate the most relevant differences between speech and singing:

- Voiced / Unvoiced ratio: In singing voice, around a 90% of the sounds produced are voiced, whereas in speech the ratio of voiced sounds is around a 60%.

- Stability of pitch: In speech, pitch is not generally stable, and it usually consist of chirps up or down within each phoneme or word. In singing, pitch is typically stable within each note, although certain expressive resources may produce predictable and controlled pitch deviations, such as vibrato or pitch bends. In Figure 2.3, we show a good example of this difference.

- Range of Pitch: In speech, the range of pitch is determined by the speaker comfort and emotional state, and it is typically narrower than the singing range of pitch, which is determined by physiology and training.

- Dynamic range: The dynamic range of singing is greater than that of typical speech. Greater flow rates and greater excursions of the vocal folds imply that the singing system is likely to operate in higher orders of non-linearity.

- Singer formant: In opera singing styles, singers tend to group the third, fourth and sometimes fifth formants together around 3 KHz for increased resonance. Opera solo singers use the singer's formant in order to be heard above the instruments. In other music styles, such as jazz or rock, the singer's formant is not used. Note that vocal resonances are also highly important in non-opera styles, but they are not grouped into the singer's formant in the same way as in opera singers.

- Singer's vowel modification: When singing, the vowel sounds may mutate a function of pitch for comfort, projection and/or intelligibility. Some modifications in the sound is an artifact of wider harmonic spacing under the vocal tract filtering spectrum envelope, rather than spectral envelope change [Cook, 1999].

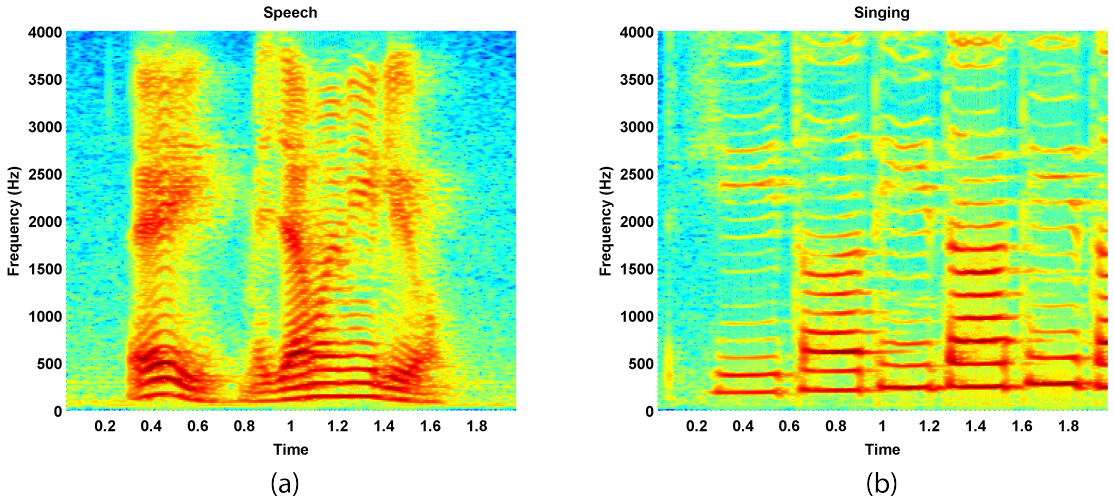

Figure 2.3: (a) Spectrogram of speech (b) Spectrogram of pop singing. Both cases have been produced by the same male voice.

2.2 Pitch Estimation

In this section, we present an overview on pitch estimation. Note that, as mentioned in Chapter 1, in this thesis the term *pitch* and fundamental frequency (F0) are used indistinctly. This Chapter covers three main research problems related to pitch estimation: monophonic F0 estimation (Section 2.2.1), melody extraction (Section 2.2.2) and multiple F0 estimation (Section 2.2.3). The specific description of each problem is provided in each of the following sections.

2.2.1 Monophonic F0 Estimation

Monophonic F0 estimation refers to the problem of estimating the F0 contour of a signal containing one single melody without accompaniment. This is a classic problem in MIR research, and has been addressed from many different perspectives in the last decades [Gómez et al., 2003b]. Monophonic F0 estimation is typically computed frame by frame to provide the curve of instantaneous F0 along time. Depending on the way frames are processed, there are two major approaches: time-domain algorithms (Section 2.2.1.1), which directly process the waveform of the signal, and frequency-domain algorithms (Section 2.2.1.2), which work in the spectral domain. In addition, some monophonic F0 estimation methods greatly improve their accuracy by introducing a time tracking stage that smooths the frame-wise F0 estimation (some relevant approaches are described in Section 2.2.1.3). Finally, in

Section 2.2.1.4 we describe current algorithms to solve *voicing* problem, that is commonly needed in monophonic F0 estimation to avoid reporting noisy F0 estimations in unvoiced regions (e.g. silence).

2.2.1.1 Time-domain Algorithms

In most approaches, F0 candidates are computed frame by frame in order to define a contour along time. In some cases, the F0 candidate with highest strength is selected as the F0 value for each frame. In other cases, F0 candidates are tracked along time in order to provide a more accurate estimation. Depending on the way F0 candidates are estimated for one frame, two main categories of algorithms can be identified: time domain algorithms and frequency domain algorithms.
Time domain algorithms try to find the periodicity of the input signal directly from the waveform. Most relevant time-domain approaches for pitch estimation are based on the autocorrelation operator and its variants [Rabiner, 1977]. The autocorrelation method has inspired a variety of successful algorithms [Boersma, 1993] [Shimamura and Kobayashi, 2001] [De Cheveigné and Kawahara, 2002], among which Yin algorithm is especially relevant.
Yin algorithm was developed by [De Cheveigné and Kawahara, 2002], and it is still today the basis of modern state-of-the-art algorithms for F0 estimation [Mauch, 2014]. This algorithm resembles the autocorrelation method [Rabiner and Schafer, 1978], but it introduces relevant improvements that make it more robust and accurate. Specifically, the autocorrelation function is replaced by the cumulative mean normalized difference function $d'_t(\tau)$, which peaks at the optimal local period with lower error rates than the traditional autocorrelation function. The cumulative mean normalized difference function $d'_t(\tau)$ is based on the squared difference function $d_t(\tau)$, which is defined as follows:

$$d_t(\tau) = \sum_{j=t}^{t+W} (x_j - x_{j+\tau})^2 \tag{2.1}$$

where: τ = Integer lag variable such that $\tau \in [0, W)$
t = Time index
W = Window size
x_τ = 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:

$$d'_t(\tau) = \begin{cases} 1 & \tau = 0 \\ \frac{d_t(\tau)}{\frac{1}{\tau}\sum_{j=1}^{\tau} d_t(j)} & \text{otherwise} \end{cases} \tag{2.2}$$

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 F0 $= f_s/\tau_p$, where f_s is the sampling rate. The aperiodicity measure ap, also called voicing parameter [Krige et al., 2008], 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.
Apart from autocorrelation-based approaches, the literature reports other time-domain algorithms for F0 estimation, such as zero-crossing rate [Kedem, 1986] (the simplest one) or parallel processing [Gold and Rabiner, 1969]. See [Gómez et al., 2003b] for a review.

2.2.1.2 Frequency-domain Algorithms

These algorithms search for the fundamental frequency from spectral information of the signal, using the STFT or other kind of transformation.
Many different algorithms for F0 estimation in the frequency-domain have been proposed for decades. In the late 60s, Noll proposed several algorithms based on this approach: the use of the cepstrum for pitch estimation, since it peaks at the period of the signal under certain circumstances [Noll, 1967]; and a method based on harmonic product spectrum, which was based on the computation of the common divisor of its harmonic sequence [Noll, 1969]. Some years later, in 1987, Lahat et al. proposed a method based on the spectrum autocorrelation, which derived from the observation that a periodic but non-sinusoidal signal has a periodic magnitude spectrum, the period of which is the fundamental frequency [Lahat et al., 1987].
On the other hand, some other successful frequency-domain approaches are based on the idea of *harmonic matching*. This idea consists of comparing the harmonic positions of a predicted F0 and the actual positions of the harmonics in the signal. One of the most successful implementations is the *Two-way mismatch* (TWM) algorithm preseted by [Maher and Beauchamp, 1994]. In TWM algorithm, for each fundamental frequency candidate, mismatches between the harmonics generated and the measured partials frequencies are averaged over a fixed subset of the available partials. A weighting scheme is used to make the procedure robust to the presence of noise or absence of certain partials in the spectral data. The discrepancy between the measured and predicted sequences of harmonic partials is referred as the mismatch error.
A more recent frequency-domain approach is SWIPE method, proposed by Camacho in 2007 [Camacho and Harris, 2008]. This algorithm estimates the pitch as the fundamental frequency of the sawtooth waveform whose spectrum best matches the spectrum of the input signal. The algorithm proved to outperform other well-known F0 estimation algorithms, and it is used in the F0 estimation stage of some

state-of-the-art query-by-humming systems [Li et al., 2013].

However, despite the big amount of frequency-domain approaches proposed during decades, the current state-of-the-art in F0 estimation for monophonic signals is mainly based on the time-domain [Boersma, 1993] [Talkin, 1995] [De Cheveigné and Kawahara, 2002] [Mauch, 2014].

2.2.1.3 Tracking Stage

In F0 estimation context, *tracking* consists of connecting the most convenient F0 candidates from every frame in order to create a smooth and representative F0 contour. Nowadays, most state-of-the-art methods for F0 estimation use some kind of tracking strategy.

One of the most relevant tracking methods for monophonic F0 estimation is [Boersma, 1993], since nowadays it is still used with success in several contexts [Molina et al., 2014d]. It defines a local strength for each F0 candidate at each frame by using a large set of parameters: *time step*, *pitch floor*, *number of candidates*, *silence threshold*, *voicing threshold*, *octave cost*, *octave-jump cost*, *voiced / unvoiced transition cost*, *pitch ceiling*. As proved by [Molina et al., 2014d], this method significantly improves its performance for query-by-singing-humming when its parameters are adapted to the input signal. The optimal path between F0 candidates is solved using dynamic programming. A similar approach is also proposed in [Talkin, 1995], and nowadays it is also widely used with success.

Recently, in 2014, Matthias Mauch has proposed pYIN [Mauch, 2014], which adds an HMM-based tracking stage to the well-known Yin algorithm [De Cheveigné and Kawahara, 2002] in order to find a smooth path over the F0 candidates found by Yin. This combination leads to excellent results in the context of query-by-singing-humming, specially in the case of highly degraded singing signals.

2.2.1.4 Voicing

The process of detecting voiced sounds (when vocal folds vibrate) in singing or speech is called *voicing*. Since fundamental frequency only makes sense in periodic sounds (voiced sounds), the voicing process is needed to obtain a representative and clean F0 contour from a speech or singing signal. Some approaches estimate voiced sounds using a wide variety of descriptors: the RMS [Haus and Pollastri, 2001], the instantaneous aperiodicity measure [Ryynänen, 2006], the evidence of pitch [Salamon and Gómez, 2012], or the zero crossing rate (ZCR) combined with the RMS [Rabiner, 1977]. In other cases, *unvoiced state* acts just like one more candidate F0 within a tracking stage [Boersma, 1993] [Mauch, 2014].

2.2.2 Melody Extraction

Melody extraction refers to the problem of F0 estimation of a single predominant pitched source from polyphonic music signals with a lead voice or instrument [Salamon2013]. This problem is directly related to some aspects of singing voice processing, and it is more challenging than monophonic F0 estimation.
In the context of polyphonic audio, monophonic F0 estimators do not perform well because of the presence of more than one pitch simultaneously. As a consequence, melody extraction methods are generally based on the concept of *pitch salience*, which is a function that represents the salience of each F0 within a frame. This function is computed in various steps:

1. **Preprocessing**: Some approaches apply a preprocessing to the signal: bandpass filtering [Goto, 2004], equal loudness filtering [Salamon and Gómez, 2012], enhancement of lead voice through source separation [Hsu and Jang, 2010] [Yeh et al., 2012].

2. **Spectral transform:** Next, the signal is windowed into frames and a transform function is applied to obtain a spectral representation of each frame. Different type of transformations can be applied: Short-Time Fourier Transform (STFT) [Ryynänen and Klapuri, 2008] [Salamon and Gómez, 2012], multirate filterbank [Goto, 2004], constant-Q transform [Cancela, 2008], multi-resolution FFT [Dressler, 2006] [Hsu and Jang, 2010] [Yeh et al., 2012], etc.

3. **Peaks extraction:** After applying the transform, most approaches only use the spectral peaks for further processing.

4. **Salience computation:** At the core of salience based algorithms lies the multipitch representation, i.e. the salience function [Klapuri, 2008]. The peaks of this function are taken as possible candidates for the melody, which are further processed in the next stages. Different methods can be used to obtain a salience function: most approaches use some form of harmonic summation, by which the salience of a certain pitch is calculated as the weighted sum of the amplitude of its harmonic frequencies [Cancela, 2008] [Hsu and Jang, 2010] [Ryynänen and Klapuri, 2008] [Salamon and Gómez, 2012] [Yeh et al., 2012]. Other approaches include two-way mismatch [Maher and Beauchamp, 1994] computed by [Rao and Rao, 2010], summary autocorrelation used by [Paiva et al., 2006] and pairwise analysis of spectral peaks as done by [Dressler, 2011].

Once the pitch salience function is available for each frame, then a tracking strategy is applied to find a smooth and representative F0 contour (as in the case of monophonic pitch trackers).

On the other hand, other approaches for melody extraction use source separation to isolate the leading voice from the accompaniment [Durrieu, 2010] [Tachibana et al., 2010], and has gained popularity in recent years following the advances in audio source separation research.

2.2.3 Multi-F0 Estimation

Multiple F0 estimation aims at identifying all pitched sounds that might be present simultaneously in an audio signal. It is different from *melody extraction* problem, because in this case we are interested not only in the lead voice, but in identifying all possible pitched sounds. As described in [Ibañez, 2010], existing approaches have been classified into:

- **Salience methods**: They try to emphasize the underlying fundamental frequencies by applying signal processing transformations to the input signal [Tolonen and Karjalainen, 2000] [Peeters, 2006] [Zhou et al., 2009] [Zhou and Mattavelli, 2007].

- **Iterative cancellation methods:** They estimate the most prominent f0, subtracting it from the mixture and repeating the process for the residual signal until a termination criterion [Klapuri, 2003] [Klapuri, 2005] [Klapuri, 2008].

- **Joint estimation methods:** They evaluate a set of possible hypotheses, consisting of F0 combinations, to select the best one without corrupting the residual at each iteration [Yeh, 2008] [Barbancho et al., 2010].

- **Supervised learning methods:** They attempt to assign a class to a musical pitch, and it applies trained classifiers (such as support vector machines or neural networks) to detect the presence of each pitch [Marolt, 2004a] [Marolt, 2004b] [Poliner et al., 2007] [Zhou, 2006].

- **Unsupervised learning methods:** They are based on non-negative matrix factorization (NMF), which approximates a non-negative matrix X as a product of two non-negative matrices W and H, in such a way that the reconstruction error is minimized: $X \approx WH$, where X represents the spectral data, H corresponds to the spectral models (basis functions), and W are the weights. This methodology is suitable for instruments with a fixed spectral

profile, such as piano sounds. Some examples of this approach are [Smaragdis and Brown, 2003] [Cont, 2006] [Raczyński and Ono, 2007] [Virtanen, 2007].

- **Matching pursuit methods:** The Matching Pursuit (MP) algorithm from [Mallat, 1993] approximates a solution for decomposing a signal into linear functions (or atoms) that are selected from a dictionary. Some works based on this approach are [Cañadas-Quesada et al., 2008] [Gribonval and Bacry, 2003] [Leveau et al., 2008].

- **Statistical modeling:** The statistical approach formulates the problem within a Bayesian framework. Bayesian statistical methods provide a complete paradigm for both statistical inference and decision making under uncertainty. Some methods performing a statistical modeling of the musical information are [Cemgil et al., 2006] [Goto, 2000], [Kameoka et al., 2007].

- **Blackboard systems:** A blackboard system integrates various forms of knowledge or information for solving complicated problems. In general, a blackboard system for Auditory Scene Analysis consists of a three-level process: low-level signal processing (peak extraction, transients, etc.), mid-level grouping (clustering of events being harmonically related, or having common onsets, etc.), and high-level stream forming (considering features such as key, scale or tempo). Examples of these methods are [Bello and Sandler, 2000] [Martin, 1996] [Ellis, 1996] [Plumbley et al., 2002].

2.3 Singing Transcription

Singing transcription, in the context of this thesis, 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" [Ryynänen, 2006]. In other words, singing transcription refers to the automatic conversion of a recorded singing signal into a symbolic representation (e.g. a MIDI file) by applying signal-processing methods. In Figure 2.4, a representative example of singing transcription using two state-of-the-art methods (pYIN [Mauch et al., 2015a] and Melotranscript[1]) is shown.
The applications of note-level singing transcription are countless. One of its renowned applications is query-by-singing-humming [Pardo et al., 2004], since many state-of-the-art approaches [Doreso, 2013] [Li et al., 2013] combine note-level and frame-

[1]https://www.samplesumo.com/melody-transcription

level matching to improve their performance. Other applications are singing tutors [Dittmar et al., 2010], computer games (e.g. Singstar[2]), tools for musicians (e.g. ScoreCloud[3]), etc.

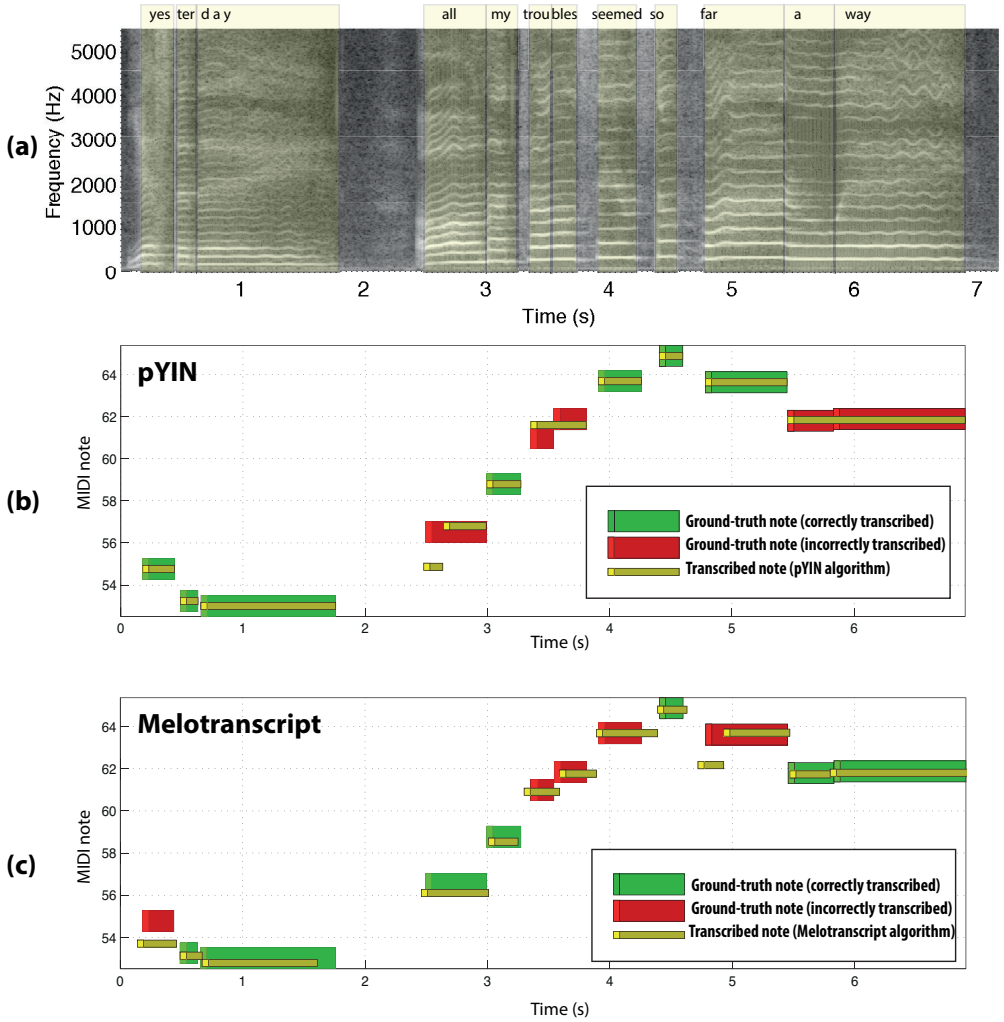

Figure 2.4: Excerpt of "Yesterday" (*The Beatles*) sung by a male amateur singer. (a) Spectrogram and syllable segmentation. (b) Transcription using pYIN. (c) Transcription using Melotranscript. The errors made by these transcribers are representative of the behavior of state-of-the-art singing transcribers with real-world audio.

Singing transcription is usually associated with melody transcription task (also called note tracking), which is more general problem because it also applies to

[2]http://www.singstar.com/
[3]http://scorecloud.com/

musical instruments. However, singing transcription is a task not only related with melody transcription, but also with 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 [Gómez et al., 2013]. As a consequence, many difficulties appear for obtaining correct F0 estimations, detecting note transitions (onsets and offsets) and labelling notes in terms of pitch or duration. These difficulties are verified when comparing state-of-the-art systems for audio onset detection (task related to note segmentation and required for automatic transcription), which yield an average F-measure (a statistical measure of accuracy, from 0 to 1) around 0.78 according to the 2010 edition of the Music Information Retrieval Evaluation eXchange (MIREX[4]). This F-measure is obtained for a mixed dataset of 85 files, but if we just consider the 5 tested singing voice excerpts, the maximum F-measure is 0.47. This suggests that state-of-the-art systems for singing voice transcription are not accurate enough to be used in an unsupervised way, even in a monophonic context.

In the literature, one can find various approaches for singing transcription (see [Molina et al., 2014b] for a comparative evaluation). In following sections, the most relevant state-of-the-art methods are described and organized into handcrafted approaches (Section 2.3.1), and probabilistic approaches (Section 2.3.2).

2.3.1 Handcrafted Approaches

A simple but commonly referenced approach was proposed by [McNab et al., 1996], and it relies on several handcrafted pitch-based and energy-based segmentation methods. Specifically it uses a "island-building" strategy, which groups areas with stable pitch values, followed by a segmentation stage that detects sudden amplitude or pitch changes. These segments are then assigned discrete note frequencies using a tuning adaptation strategy to deal with untrained singers with no stable tonal reference. Later, [Haus and Pollastri, 2001] used a similar approach with some refined rules to deal with intonation mistakes.

On the other hand, [Clarisse et al., 2002] contributed with an auditory model for pitch estimation, followed by a segmentation stage based on loudness, voicing and pitch variation. This approach led to later improved systems such as [De Mulder et al., 2003] [De Mulder et al., 2004], whose latest evolution is *Melotranscript*[1], provided by SampleSumo company.

[4]http://www.music-ir.org/mirex

2.3.2 Probabilistic Approaches

Other approaches use Hidden Markov Models (HMM) to detect note-events in singing voice [Viitaniemi et al., 2003] [Ryynänen, 2006] [Krige et al., 2008] [Mauch et al., 2015a]. These systems are directly inspired by the classical HMM-based approach for speech recognition [Young et al., 2009], where phonemes (or words) are modeled with separately trained left-to-right HMMs (acoustic model), which are connected in a larger probabilistic system determining the transitions between acoustic units (linguistic model).

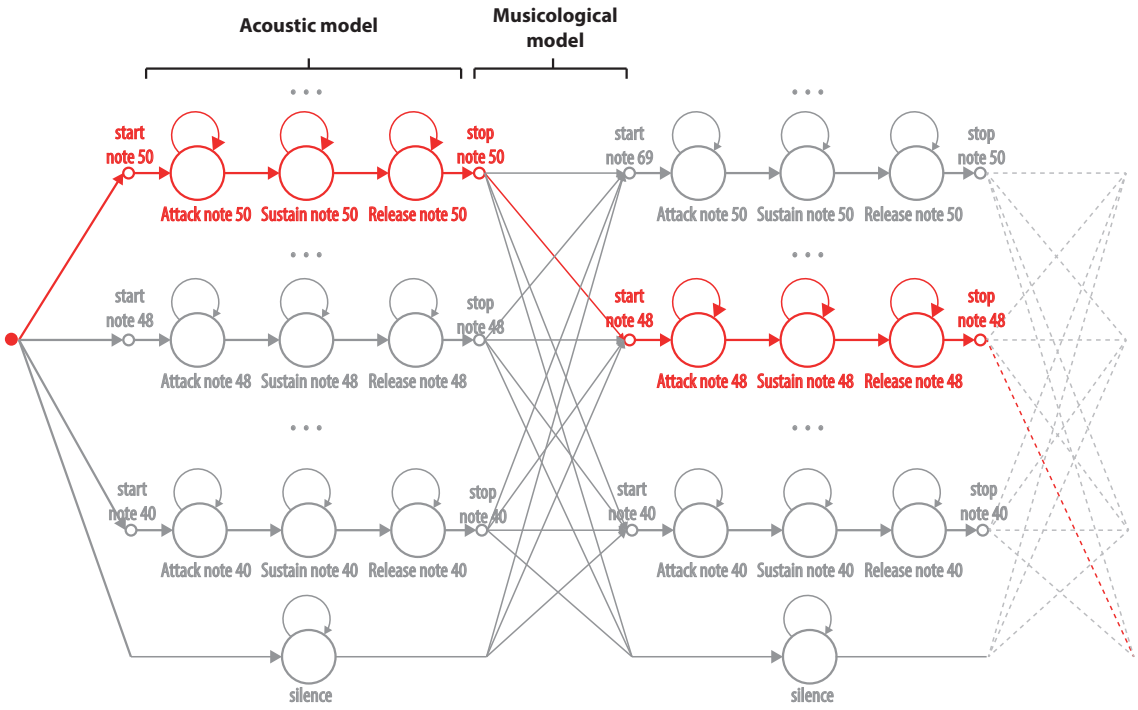

Figure 2.5: Trellis diagram of HMM-based approach proposed by [Ryynänen, 2006]. The acoustic model represents the evolution of features within the same note, and the musicological model represents the transitions probabilities between notes. In red, an example of Viterbi-decoded path is shown (corresponding to two consecutive notes).

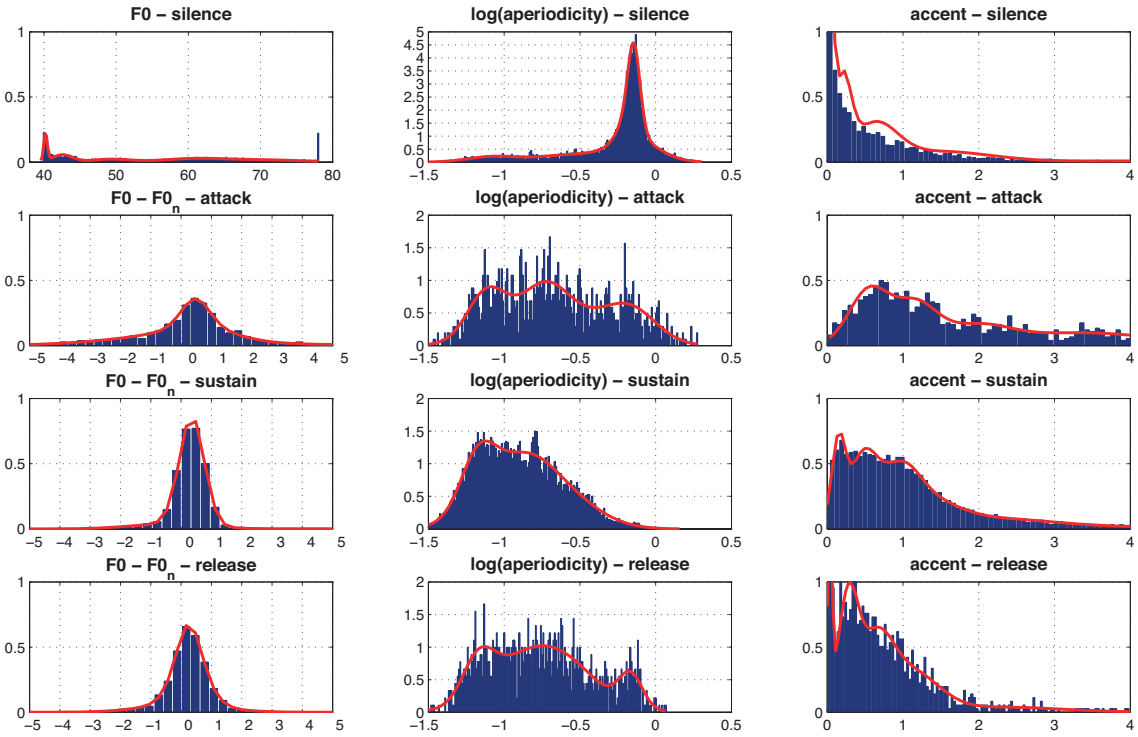

Figure 2.6: Distribution of observable features (and its Gaussian Mixture Modeling shown with red line) for each state of the HMM diagram shown in Figure 2.5.

In singing, the acoustic units are associated to musical notes, and the observable features are pitch, energy, voicing, accent, etc. In the case of [Viitaniemi et al., 2003], one state per note is used, whereas [Ryynänen, 2006] [Krige et al., 2008] and [Mauch et al., 2015a] consider several consecutive states per note (typically corresponding to attack, sustain, release or silence). The final sequence of notes corresponds to the path of states with maximum likelihood, which can be decoded using Viterbi algorithm (see [Rabiner, 1989] for a tutorial about it). This note sequence decoding process typically relies on a musicological model using key information [Viitaniemi et al., 2003] [Ryynänen, 2006], or other kind of heuristics to favor reasonable intervals while singing [Mauch et al., 2015a]. In Figure 2.5, an example trellis diagram of this HMM-based approach is illustrated. In Figure 2.6, the distribution of some features are shown for each state of such HMM-based scheme. These features have been implemented as described in [Ryynänen and Klapuri, 2004], and they have distributions for different stages of the note: silence, attack, sustain and release.
A different probabilistic approach for singing transcription is proposed by [Gómez et al., 2013]. It does not relies on hidden Markov models; instead, it performs a short note transcription by maximizing a likelihood function using low-level features (e.g. pitch, voicing or stability), and then it consolidates them into longer notes using an

iterative process.

2.4 Dynamic Time Warping

Dynamic Time Warping (DTW) is an algorithm to find an optimal alignment between two similar temporal sequences that may vary in time or speed. Some early works about DTW are [Vintsyuk, 1968] [Sakoe and Chiba, 1971] [Hiroaki, 1978], where dynamic programming algorithms are proposed for pattern matching in speech recognition. For a comprehensive tutorial about DTW in the context of music information retrieval, see [Müller, 2007]. On the other hand, a ready-to-use implementation of DTW in Matlab can be found in [Ellis, 2003].

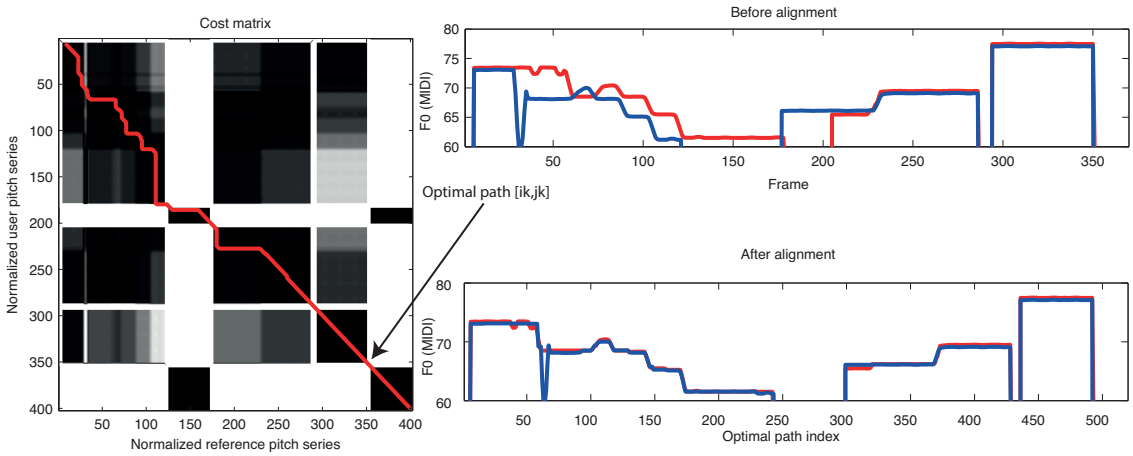

Figure 2.7: Example of use of DTW for pitch contour alignment.

As described in [Müller, 2007], the objective of DTW is to compare two (time-dependent) sequences $X := (x_1, x_2, \ldots, x_N)$ of length $N \in \mathbb{N}$ and $Y := (y_1, y_2, \ldots, y_M)$ of length $M \in \mathbb{M}$. These sequence may be discrete signals (time-series) or, more generally, feature sequences sampled at equidistant points in time. In the following, we fix a *feature space* denoted by $\mathcal{F}$. Then, $x_n, y_m \in \mathcal{F}$ for $n \in [1 : N]$ and $m \in [1 : M]$. To compare two different features $x, y \in \mathcal{F}$, one needs a *local cost measure*, sometimes also referred to as *local distance measure*, which is defined to be a function:

$$c : \mathcal{F} \times \mathcal{F} \to \mathbb{R}_{\geq 0} \tag{2.3}$$

Typically, $c(x, y)$ is small (low cost) if x and y are similar to each other, and otherwise $c(x, y)$ is large (high cost). Evaluating the local cost measure for each pair of elements of the sequences X and Y, one obtains the *cost matrix* $C \in \mathbb{R}^{N \times M}$ defined

by $C(n, m) := c(x_n, y_m)$. Then the goal is to find an alignment between X and Y having minimal overall cost.

Definition 1. *An* (N, M)-warping path *(or simply referred to as warping path if* N *and* M *are clear from the context) is a sequence* $p = (p_1, \ldots, p_L)$ *with* $p_l = (n_l, m_l) \in [1 : N] \times [1 : M]$ *for* $l \in [1 : L]$ *satisfying the following three conditions.*

(i) Boundary condition: $p_1 = (1, 1)$ *and* $p_L = (N, M)$.

(ii) Monotonicity condition: $n_1 \leq n_2 \leq \cdots \leq n_L$ *and* $m_1 \leq m_2 \leq \cdots \leq m_L$.

(iii) Step size condition: $p_{l+1} - p_l \in \{(1, 0), (0, 1), (1, 1)\}$ *for* $l \in [1 : L - 1]$.

The *total cost* $c_p(X, Y)$ of a warping path p between X and Y with respect to the local cost measure c is defined as:

$$c_p(X, Y) := \sum_{l=1}^{L} c(x_{n_l}, y_{m_l}) \tag{2.4}$$

Furthermore, an *optimal warping path* between X and Y is a warping path $p*$ having minimal total cost among all possible warping paths. The *DTW distance* $\mathrm{DTW}(X, Y)$ between X and Y is then defined as the total cost of $p*$. The computational cost of classical DTW is $O(M \times N)$, although extensive research have been performed on how to accelerate DTW computations (e.g. [Salvador and Chan, 2007] [Al-Naymat et al., 2009]).

Regarding the applications of DTW, during the 70's it was a trendy approach to perform speech recognition [Hiroaki, 1978], but it was displaced during the 80's due to the appearance of HMM based methods [Rabiner, 1989]. Nowadays, however, DTW is being proposed as a promising approach for many applications ([Anguera, 2012] claims the existence of a *DTW's new youth*). In the field of speech, it is used for query-by-example spoke term detection [Anguera and Ferrarons, 2013], unsupervised training of acoustic models [Jansen and Church, 2011], zero resources spoken term discovery [Jansen et al., 2010], etc. In addition to speech, DTW has found numerous applications in a wide range of fields including data mining, information retrieval, bioinformatics, chemical engineering, signal processing, robotics, or computer graphics; see, e. g., [Keogh and Ratanamahatana, 2004] and the references therein. In the field of music information retrieval, DTW plays in important role for synchronizing music data streams [Dixon and Widmer, 2005] [Hu et al., 2003] [Müller et al., 2004] [Müller et al., 2006] [Soulez et al., 2008]. In Figure 2.7, we show an example of use of DTW for pitch contour alignments, as used in our approach for automatic singing assessment (Section 3.3). DTW has also been used in the field of computer animation to analyze and align motion data [Bruderlin and Williams, 1995] [Giese and Poggio, 2000] [Hsu et al., 2005] [Kovar and Gleicher, 2003] [Müller and Röder, 2006].

2.5 Automatic Singing Assessment

Automatic singing assessment refers to the task of automatically analyzing a music performance in order to score it, and to provide meaningful feedback about it. In the literature, this task has been also referred as singing skill evaluation [Nakano et al., 2009], solfège evaluation [Schramm et al., 2015] or performance scoring [Mayor et al., 2006]. In this section, we present some previous approaches for automatic performance assessment (Section 2.5.1), together with a musicological analysis about the topic (Section 2.5.2).

2.5.1 Existing Systems for Automatic Assessment

Automatic singing assessment has been mainly applied to two fields: entertainment (Section 2.5.1.1) and education (Section 2.5.1.2). In most cases these two aspects are tied, but in the case of education there is a clearer aim at improving the musical skills of the user. As an exception, [Nichols et al., 2012] does not use automatic singing assessment for entertainment nor education, but for a music information retrieval system able to automatically discover talented singers in Youtube videos.

2.5.1.1 Entertainment

In last years, many musical games based on automatic performance rating have become successful (e.g. Guitar Hero[5], Rockband[6], etc.). In the case of singing voice, the main approach is a karaoke-style game with automatic intonation rating. Some examples of these games are Singstar[7] and Ultrastar[8]. These systems usually perform a relatively simple analysis of singing voice, and usually assess just pitch accuracy by comparing user's pitch contour with a reference. Other approaches are song-independent (e.g. Skore[9] or [Nakano et al., 2009]), and they analyse some features as pitch stability, vibrato, etc. in order to grade the user performace.
In general, these systems do not use formal music notation, and they are aimed at engaging the user without focusing on the proper development of music skills.

2.5.1.2 Education

Existing systems with educational purposes typically lead to complex and ambitious approaches. These systems should be able to provide a meaningful feedback in

[5] www.guitarhero.com
[6] www.rockband4.com
[7] www.singstar.com
[8] http://ultrastardx.sourceforge.net
[9] http://www.bmat.com/products/skore-en/

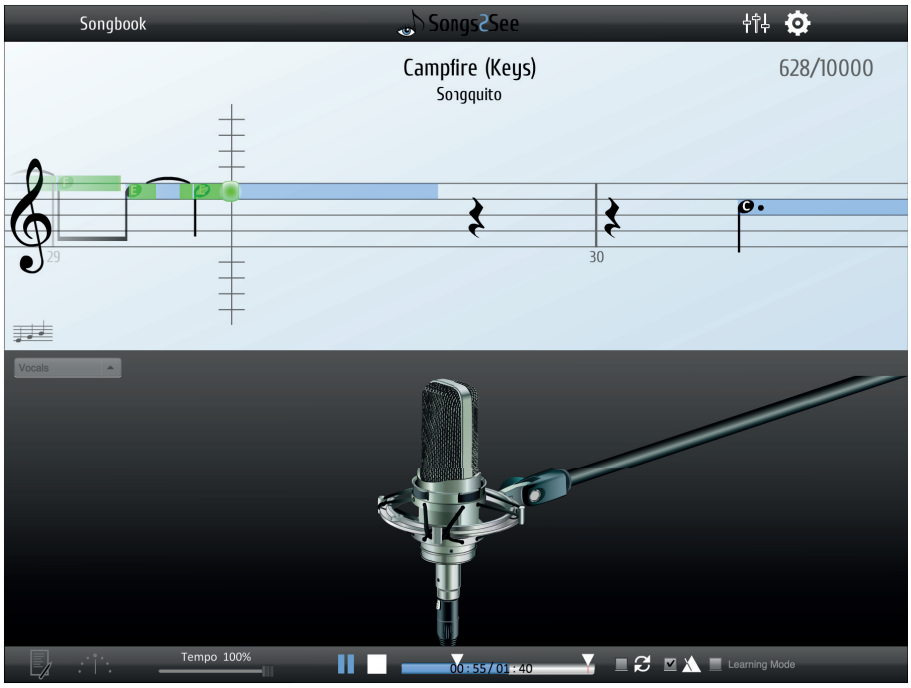

Figure 2.8: Screenshot of Songs2See web application [Dittmar et al., 2010].

order not only to engage the user, but also to incrementally improve his or her music skills. A representative example of educational tool for singing learning is Songs2See [Dittmar et al., 2010] (a screenshot is shown in Figure 2.8), which uses music notation and note-based evaluation.
Other kind of systems rather focus on providing real-time feedback to the user, instead of scoring the whole user performance. Examples are [Rossiter and Howard, 1996], [Howard et al., 2004] or Sing&See[10]. The main aim of these approaches is helping the user to better understand their mistakes through a real-time visualization of some parameters of their voice (e.g. pitch [Howard et al., 2004], vibrato [Nakano et al., 2007], etc).

2.5.2 Musicological Perspective

The assessment of a musical performance is commonly affected by many subjective factors, even in the case of expert musicians' judgments. Certain aspects such as the context, the evaluator's mood, or even the physical appearance of the performer can strongly change the perceived quality of the same performance [Griffiths and Davidson, 2006]. As a consequence, automatic assessment of user performance is

[10] www.singandsee.com

a really challenging problem. However, under certain conditions, some objective aspects can be analyzed in order to model the expert's judgment.
Previous researchers have studied the reliability of judgments in music performance evaluation [Wapnick and Ekholm, 1997] [Ekholm et al., 1998] [Bergee, 2003] [Nakano et al., 2006], with some relevant results for the purposes of this thesis. In such studies, different musicians were asked to grade a certain number of performers according to different aspects, with the aim to study how similar the different judgments were. In [Wapnick and Ekholm, 1997], the case of solo voice evaluation has been addressed through a set of experiments, with a focus on technique aspects: appropriate vibrato, color/warmth, diction, dynamic range, efficient breath management, evenness of registration, flexibility, freedom in vocal range, intensity, intonation accuracy, legato line, resonance/ring and overall score. Among these aspects, the ones presenting a higher reliability were intonation accuracy, appropriate vibrato, resonance/ring and the overall score. In [Bergee, 2003], the rhythm/tempo aspects are also considered, and the conclusions are quite similar. Since intonation, vibrato, timbre (resonances) and overall score seems to be more objective aspects than the others (according to the reliability analysis), these aspects are good candidate features to build automatic assessment systems.

2.6 Timbre Processing

In this section, we present a review of techniques and approaches for voice timbre processing. In Section 2.6.1 the source-filter model is presented. Then, the most common approaches for spectral envelope extraction are described in Section 2.6.2. In Section 2.6.3 we review some concepts related to formant analysis. Finally, in Section 2.6.4 we describe a set of features for timbre processing that have been successfully applied in state-of-the-art speech and singing applications.

2.6.1 Source-Filter Model

As described in Section 2.1.1, human voice can be modelled as an excitation (produced by the vocal chords), shaped by some resonators (vocal tract). This excitation-resonance model is also known as source-filter model [Zölzer, 2011]. Although this generic concept is present in a varied set of synthesis models (e.g. Klatt synthesizer [Klatt, 1980]), the term *source-filter processing* in the literature generally refers to a specific scheme for sound processing based on a time-frequency representation, where the spectral envelope and the source signal are separately processed frame by frame. In Figure 2.9, we show the block diagram of the source-filter processing scheme. In such scheme, the *Spectral Envelope Estimation* block is the core of the system, and much effort in the literature is devoted to achieve good spectral enve-

lope estimators (see Section 2.6.2). Note that, despite source-filter model fits well the acoustic mechanism of voice production, the inverse-filtered excitation signal $e_1(n)$ does not totally correspond to the glottal source, since the glottal source is a low-pass signal, whereas $e_1(n)$ is perfectly white.

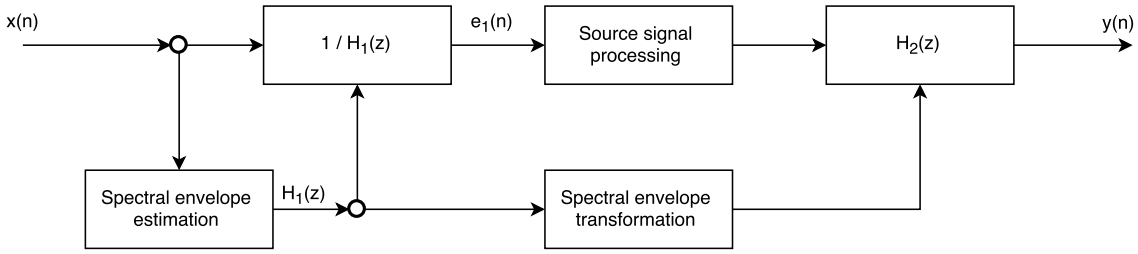

Figure 2.9: Schema of source-filter processing: $x(n)$ = input speech / singing signal, $H_1(z)$= original spectral envelope filter, $e_1(n)$=source signal (white in frequency), $H_2(z)$=transformed spectral envelope filter, $y(n)$=output transformed speech / singing signal

Source-filter processing is useful to perform any kind of voice transformation in which the spectral envelope must be separately processed. For instance, proper pitch shifting in singing voice must avoid formants to be uncontrollably shifted along pitch. In this case, the use of source-filter processing has been successful to perform pitch shifting with formants preservation [Röbel and Rodet, 2005]. Additionally, source-filter processing also allows to perform spectral morphings, or certain effects based on spectral envelope processing (such as formants shifting without modifying the pitch).

2.6.2 Spectral Envelope Extraction

The term spectral envelope denotes a smooth function that passes through the prominent spectral peaks [Röbel and Rodet, 2005]. However, there exists no technical or mathematical definition for it, and what is desired depends to some extend on the signal. In the case of speech and singing, the desired spectral envelope is the actual acoustic response of the vocal tract producing the target sound. Two classic approaches for this problem are linear predictive coding (LPC) (Section 2.6.2.1), and cepstrum-based methods (Section 2.6.2.2). A relevant variant of cepstrum-based methods is *true envelope algorithm* [Imai and Abe, 1979], which estimates the spectral envelope using an iterative approach (Section 2.6.2.3). For a comprehensive review on spectral envelope estimation see [Zölzer, 2011].

2.6.2.1 LPC-based Methods

Linear Predictive Coding (LPC) is used to efficiently find the coefficients of an all-pole filter that fits the magnitude spectrum of an stationary input signal $x(n)$ [Makhoul, 1975]. This model works well for voice, since the all-pole filter is a good approximation of the acoustic response of vocal resonances.
In LPC the current input signal $x(n)$ is approximated by a linear combination of past samples of it. The prediction of $x(n)$ is computed using an FIR filter by:

$$\hat{x}(n) = \sum_{k=1}^{p} a_k x(n-k) \tag{2.5}$$

where: p = Prediction order
a_k = Prediction coefficients

The difference between the original input signal $x(n)$ and its prediction $\hat{x}(n)$ is called residual or prediction error, and it is evaluated by:

$$e(n) = x(n) - \hat{x}(n) = x(n) - \sum_{k=1}^{p} a_k x(n-k) \tag{2.6}$$

If LPC estimation is done with a high enough prediction order, $e(n)$ tends to be flat in frequency. The z-transform of $e(n)$, $E(z)$, can be then related to the concept of spectral envelope:

$$\hat{X}(z) = X(z) \sum_{k=1}^{p} a_k z^{-k} = X(z)P(z) \tag{2.7}$$

$$E(z) = X(z)\big[1 - P(z)\big] \tag{2.8}$$

$$E(z) = X(z)A(z) \tag{2.9}$$

where: $P(z)$ = Prediction filter
$A(z)$ = Prediction filter error

Since $E(z)$ tends to be flat in frequency, $A(z)$ models the inverse of the spectral envelop. The inverse filter of $A(z)$ is an IIR all-pole filter $H(z)$, which is called *synthesis filter* or *LPC filter*, and represents the spectral envelope of the signal:

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 - P(z)} \tag{2.10}$$

$$H(z) = \frac{1}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots a_N z^{-N}} \tag{2.11}$$

Since LPC coefficients a_k are hard to interpret and manipulate, $H(z)$ is commonly expressed using its poles:

$$H(z) = \frac{1}{(1 - p_1 z^{-1})(1 - p_2 z^{-2}) \dots (1 - p_N z^{-N})} \tag{2.12}$$

The literature reports some voice processing methods based on manipulating these poles [Slifka and Anderson, 1995] [Morris and Clements, 2002]. In Figure 2.10 we show the estimated $H(z)$ for a short window of a speech signal with different LPC orders.

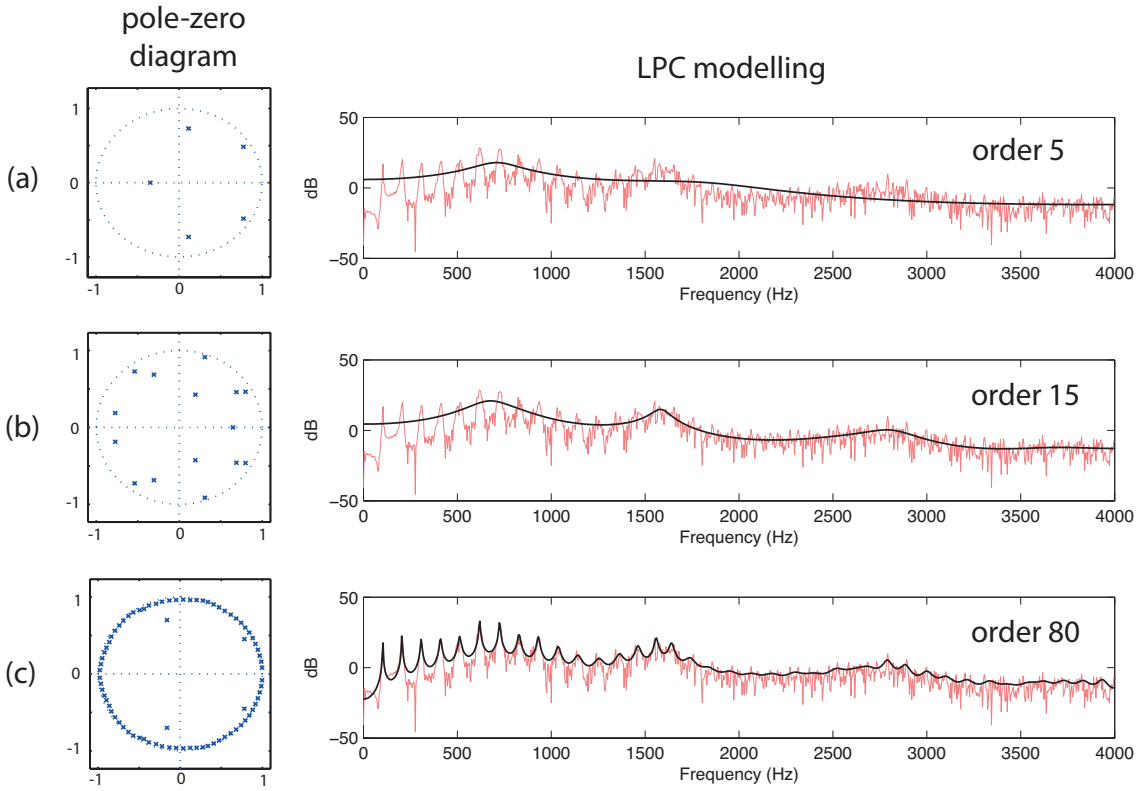

Figure 2.10: LPC modeling for a short window of speech using different orders. In left side, we show the poles-zeros diagram, and in right side we present the estimated spectral envelope. Note the importance of choosing a good LPC order to avoid undesirable under- or over-fitting to the magnitude spectrum.

Regarding the computation of LPC filter coefficients, there exists three main methods: autocorrelation, covariance (both described in [Markel and Gray, 1976]) and Burg [Gray and Wong, 1980], among which the autocorrelation method is the most common one. In the autocorrelation method, the prediction error energy

$$E_p = E\{e^2(n)\} \tag{2.13}$$

of a windowed excerpt of the signal is minimized by setting the partial derivatives to zero:

$$\partial E_p / \partial a_i = 0 \tag{2.14}$$

This system of equations can be expressed in a compact way using the autocorrelation operator $r_{xx}(i)$, finally leading to the so-called *normal equations*:

$$\sum_{k=1}^{p} a_k r_{xx}(i-k) = r_{xx}(i) \tag{2.15}$$

which can be efficiently solved using Levinson-Durbin recursion [Levinson, 1947]. The autocorrelation method is implemented in `lpc` function of the Signal Processing Toolbox of MATLAB[11].

LPC has been used to perform formant analysis [Snell and Milinazzo, 1993], music/speech/noise segmentation [Muñoz-Expósito et al., 2005], speaker modification using poles warping [Slifka and Anderson, 1995], etc. Other systems based on LPC make use of a different representation of filter coefficients called Line Spectral Frequencies (LSF) or Line Spectral Pairs (LSP) [McLoughlin, 2008], which is more appropriate to perform spectral interpolations. However, LPC-based approaches have a sort of drawbacks that motivate research on alternatives. Specifically, the optimal order p of the LPC filter is hard to obtain, and it directly affects the usefulness of the estimated spectral envelope. If the order p is too low, the resulting envelope may fit poorly the spectrum of the signal. In contrast, if p is too high, there may be a problem of overfitting. Moreover, even if the optimal order were known, it contains systematic errors due to the fact that the harmonic spectrum sub-samples the spectral envelope. These problems are especially manifested in voiced and high pitched signals.

2.6.2.2 Cepstrum-based Methods

The *real cepstrum* (commonly called simply *cepstrum*) is the result of taking the Inverse Discrete Fourier Transform (IDFT) of the log magnitude of the DFT of a signal:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \tag{2.16}$$

$$c[n] = \sum_{n=0}^{N-1} \log\big(|X[k]|\big) e^{j\frac{2\pi}{N}kn} \tag{2.17}$$

[11] http://es.mathworks.com/help/signal/ref/lpc.html

In this thesis, we apply the term *cepstrum* to refer to the *real cepstrum*, and it must not be confused with the *complex cepstrum* or the *power cepstrum*, that are different transformations [Childers et al., 1977].
Roughly, the cepstrum can be seen as the "spectrum of the spectrum", so low n values in $c[n]$ are related to smooth variations in the log magnitude of the spectrum, and high n values to rapid variations. These n values are commonly called *quefrencies*, in order to differentiate them from standard frequency concept. If we keep only the cepstral coefficients with low quefrencies, and we apply an extra DFT transformation to $c[n]$, we obtain a smoothed representation of the original log magnitude, i.e. the spectral envelope. In Figure 2.11 we show the computational steps to estimate the spectral envelope of a signal using the cepstrum.

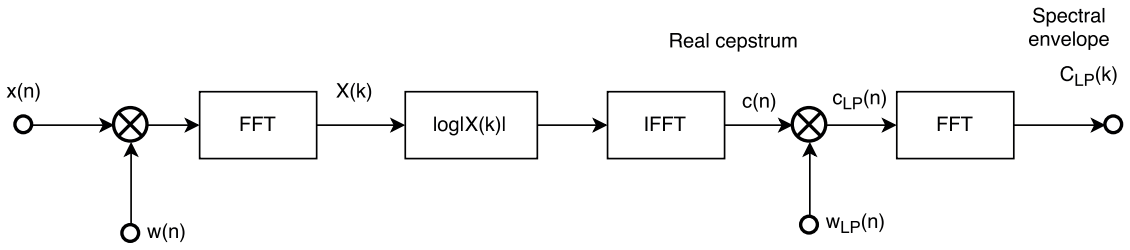

Figure 2.11: Block diagram of spectral envelope estimation using cepstral smoothing

Regarding the limitations of cepstrum-based spectral envelope estimation, they are similar to the case of LPC: the behavior of the spectral envelope depends on the chosen cut-off quefrency (cepstral order), similarly as in the case of LPC with the filter order. Again, as in the case of LPC, the spectral envelope is poorly estimated for high-pitched sounds due to the heavy sub-samping of the spectrum. Besides, using cepstrum, the spectral envelope does not exactly fit the spectral peaks, as shown in the cepstral smoothing of the spectrum at iteration 1 in Figure 2.12.

2.6.2.3 True Envelope

The *true envelope* estimator has been proposed originally in 1979 [Imai and Abe, 1979], and it is based on cepstral smoothing of the amplitude using an iterative procedure. Let $X[k]$ the K-point DFT of the signal frame $x[n]$ and $E_i[k]$ the spectral envelope resulting of a cepstral smoothing at iteration i. The algorithm then iteratively updates the smoothing input spectrum $A_i[k]$ with the maximum of the original spectrum and the current cepstral representation:

$$A_i[k] = \max(\log(|X[k]|), E_{i-1}[k]) \tag{2.18}$$

and apply the cepstral smoothing to $A_i[k]$ to obtain $E_i[k]$. The procedure is initialized setting $A_0[k] = \log(|X[k]|)$, and starting the cepstral smoothing to obtain

$E_0[k]$. The estimated envelope will steadily grow. The algorithm stops if for all k the relation $A_i[k] < E_i[k]+\theta$. In Figure 2.12 we show the obtained spectral envelope for several iterations of this algorithm for a voiced speech frame.

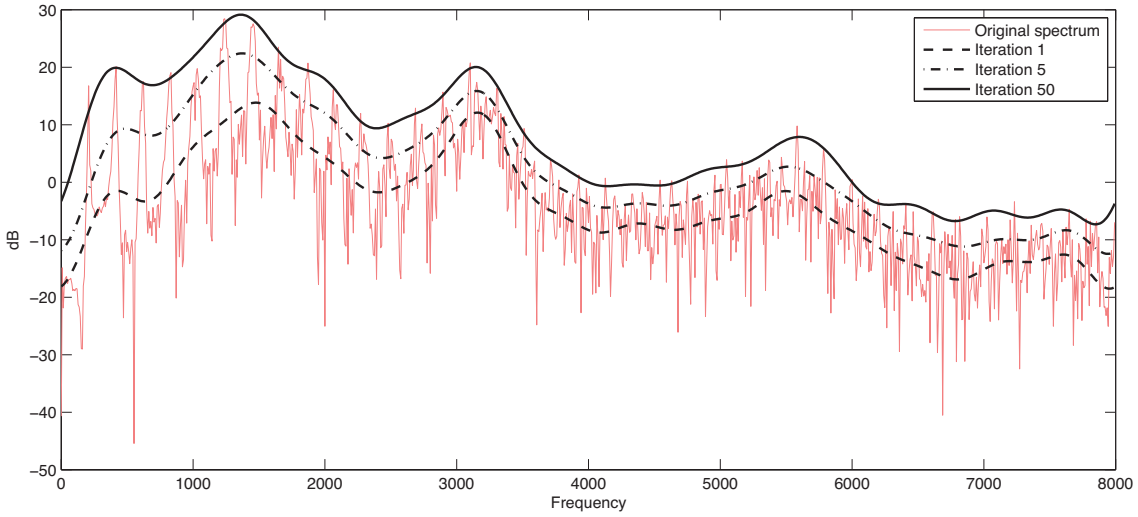

Figure 2.12: Spectral envelope obtained with the *True Envelope* algorithm for several iterations. Note that, at iteration 1, the spectral envelope is a simple cepstral smoothing of the $\log(|X[k]|)$.

For more information about the true envelope estimator, see [Zölzer, 2011] and [Villavicencio et al., 2006].

2.6.3 Formant Analysis

The Acoustical Society of America defines a *formant* as: "a range of frequencies [of a complex sound] in which there is an absolute or relative maximum in the sound spectrum" [ANSI, 2004]. In speech science and phonetics, however, a formant is sometimes used to mean an acoustic resonance of the human vocal tract [Titze, 2000]. Thus, in the literature, formant can mean either a resonance or the spectral maximum that the resonance produces. In this thesis, we use the term *formant* to mean an acoustic resonance (as in [Titze, 2000]).

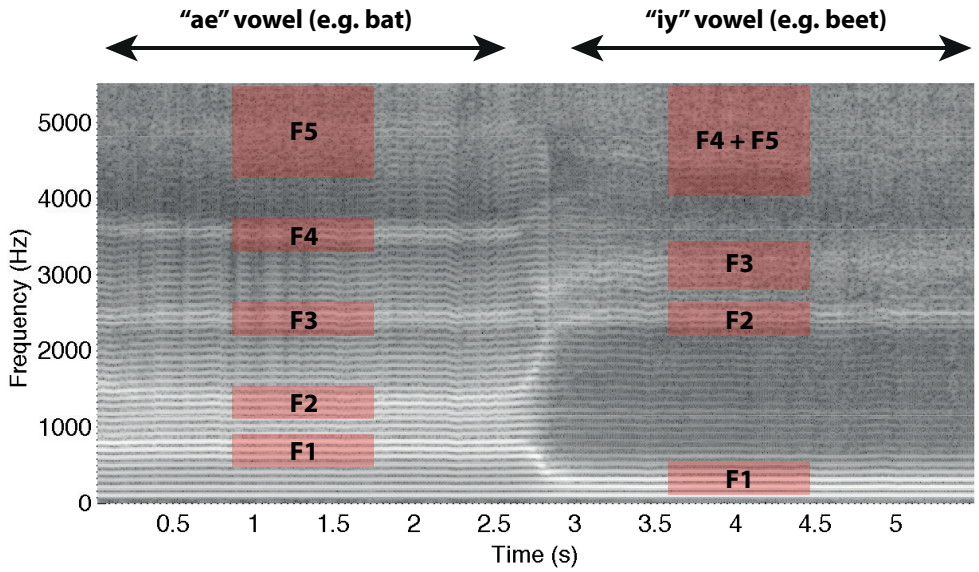

Figure 2.13: Spectrogram of two consecutive low-pitched vowels uttered by a young man: *"ae"* (e.g. bat) and *"iy"* (e.g. beet). During the whole utterance, formants F1, F2 and F3 are clear and easy to track, whereas F4 and F5 seem to be merged in "iy" vowel.

The frequencies and bandwidths of formants are primarily dependent upon the shape of the vocal tract, which is determined by the position of the articulators (tongue, lips, jaw,etc.). In continuous speech or singing, the formant frequencies vary as the articulators change position. Typically, no more than five formants are considered in speech or singing analysis, whose frequencies are labeled as F1-F5 (see Figure 2.13). The first two formants F1 and F2 are frequently used as a compact representation of the vowel space (see Figure 2.14), whereas F3, F4 and F5 are rather related to the voice timbre. In the specific case of opera singing, F3 and F4 (and sometimes also F5) are commonly grouped to create the "singer formant" [Sundberg, 2001], which produces a significant boost around 3kHz to be heard above the orchestra.

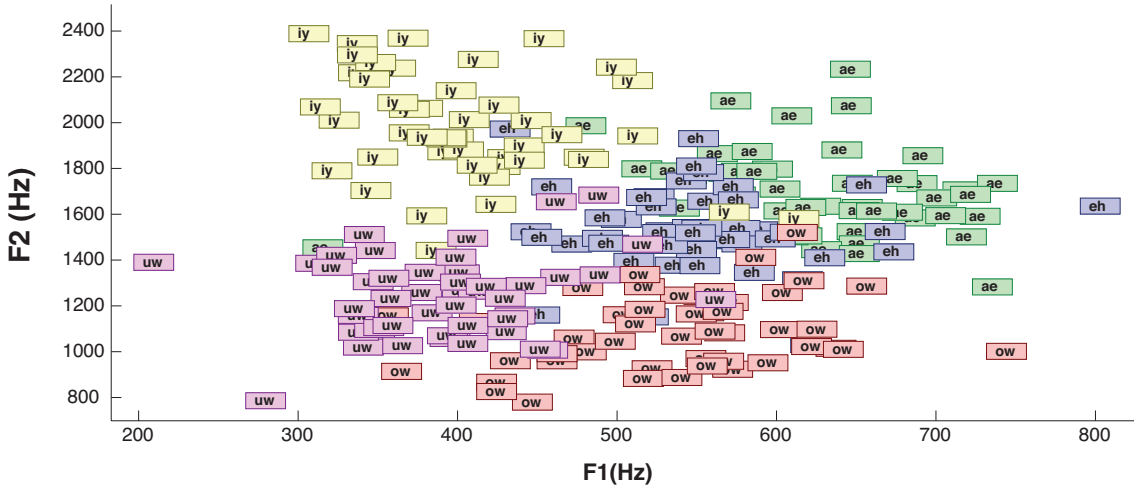

Figure 2.14: Distribution of first and second formants frequencies (F1 and F2) for 46 different phones spoken by several male speakers, taken five different phonemes categories: *"ae"* (e.g. bat), *"ew"* (e.g. bet), *"iy"* (e.g. beet), *"ow"* (e.g. boat), *"uw"* (e.g. boot). See TIMIT dataset documentation for more details [Garofolo, 1993]. The plotted formants frequencies are manual annotations obtained from VTR dataset [Deng et al., 2006].

The literature reports many different approaches for automatic formant analysis. The classical method is based on all-pole modeling using LPC analysis, and it consists in associating the poles positions with the actual formants frequencies [Snell and Milinazzo, 1993]. This approach is the one suggested by Matlab documentation for formants extraction[12]. Some variants of this approach propose improvements in LPC modeling [Alku et al., 2013], or the use of phase in order to estimate formants [Bozkurt et al., 2004]. Other approaches also include a tracking stage, frequently based on dynamic programming and/or probabilistic models [Xia and Espy-Wilson, 2000], in order to obtain more accurate formants trajectories. Finally, the literature also report methods based on less common strategies: multiband energy demodulation [Potamianos and Maragos, 1996], auditory preprocessing and bayesian estimation [Gläser et al., 2010], particle filters [Zheng and Hasegawa-Johnson, 2004], etc.

Automatic formant analysis has been widely applied in phonetic studies [Carlsson and Sundberg, 1992] [Busby and Plant, 1995], speech and singing synthesis [Borges et al., 2008] [Bonada and Serra, 2007], as well as in some approaches for classic speech-related problems such as automatic speech recognition[Holmes et al., 1997] or speaker verification[Becker et al., 2008]. However, state-of-the-art approaches for speech recognition are not based on formant-based features, but in MFCC [Baker

[12] www.mathworks.com/help/signal/ref/lpc.html

et al., 2009], PLP [Hermansky, 1990] or in deep neural networks [Hinton et al., 2012]. The reason is that automatic formant tracking techniques can introduce important errors when two formants are merged (e.g. F1 and F2 in "uw" vowels, or F2 and F3 in "iy" vowels), or when a formant does not produce a spectral prominence in the spectrum [Deng et al., 2006]. As a conclusion, formant-based features are a compact representation of speech and singing signals, and they are interestingly related to physical aspects of the vocal tract, but in practice their estimation is not reliable enough for many real-world problems related to speech and singing.

2.6.4 Features for Timbre Processing

In this section we present more timbre-related features commonly used for speech and singing analysis. First, we describe the well-loved MFCC (Section 2.6.4.1), which have been successful in many different problems related to speech and singing. In Section 2.6.4.1 we describe PLP and RASTA-PLP, which are speaker-invariant features. Then, in Section 2.6.4.3 we present two relevant time-domain features for speech and singing processing: zero crossing rate and 4Hz modulation. In Section 2.6.4.4 we present some frequency domain features which are commonly used in many audio analysis problems. Finally, we draft some ideas about unsupervised feature learning in Section 2.6.4.5.

2.6.4.1 Mel-Frequency Cepstral Coefficients (MFCC)

Perhaps, the most versatile feature for timbre analysis are Mel-Frequency Cepstral Coefficients (MFCC) [Logan, 2000]. MFCCs are used in state-of-the-art solutions for classic problems such as speech recognition [Baker et al., 2009] or speaker identification [Hasan et al., 2004]. They are computed in 5 steps (see [Ellis, 2005] for a Matlab implementation):

1. Take the Fourier Transform of a frame.

2. Map the powers of the spectrum obtained above onto the Mel scale, using triangular overlapping windows.

3. Take the logs of the powers at each of the Mel frequencies.

4. Take the discrete cosine transform of the list of Mel log powers, as if it were a signal.

5. The MFCCs are the amplitudes of the resulting spectrum. In many speech-related applications, only the coefficients 2 to 13 are kept.

MFCCs are a compact representation of a sound frame (using typically 12 coefficients named as $C1$-$C12$), and they convey highly discriminatory information for phonetic and timbre analysis. In addition, MFCC coefficients are fairly uncorrelated, and therefore they can be used in simple statistical models. In many applications, the temporal difference of MFCCs (known as ΔMFCC), and the temporal difference of ΔMFCC (known as $\Delta\Delta$MFCC) are also considered [Young et al., 2009].

2.6.4.2 PLP and RASTA-PLP

Perceptual Linear Prediction (PLP) was originally proposed by Hynek Hermansky in 1990 as a way of warping spectra to minimize the differences between speakers while preserving the important speech information [Hermansky, 1990]. This technique uses three concepts from the psychoacoustics of hearing to derive an estimate of the auditory spectrum: (1) the critical-band spectral resolution, (2) the equal-loudness curve, and (3) the intensity-loudness power law. The auditory spectrum is then approximated by an autoregressive all-pole model.
RASTA is a separate technique that applies a band-pass filter to the energy in each frequency subband in order to smooth over short-term noise variations and to remove any constant offset resulting from static spectral coloration in the speech channel e.g. from a telephone line [Hermansky and Morgan, 1994]. A Matlab implementation of this technique can be found in [Ellis, 2005].

2.6.4.3 Time-domain Features

In this section we describe two simple, but highly relevant, time-domain features for speech and singing analysis: zero crossing rate and 4Hz modulation.

- **Zero Crossing Rate:** As defined in [Kedem, 1986], the Zero Crossing Rate (ZCR) is the rate of sign-changes along the signal, i.e., the rate at which the signal changes from positive to negative or back. This measure highly correlates with the spectral center of mass or Spectral Centroid of the input signal (see Section 4.3.3).

- **4Hz Modulation:** The 4Hz Modulation Energy Peak is a characteristic feature of speech signals due to a near 4Hz syllabic rate of english language. It is computed by decomposing the original waveform into 20 [Karneback, 2001] or 40 [Scheirer, 1998] mel-frequency bands, depending on the accuracy. The energy of each band is extracted and a second band pass filter centered at 4 Hz is applied to each one of the bands.

2.6.4.4 Frequency-domain Features

Along with MFCCs (already described in Section 2.6.4.1), many frequency-domain features can be used to perform timbre analysis of audio signals. In this section, we provide a brief description of each one (for more information see [Guaus, 2009]).

- **Spectral centroid:** The Spectral Centroid is defined as the balancing point of the spectral power distribution [Scheirer, 1998]. It is related to the perceived brightness of a sound.

- **Spectral flatness:** The Spectral Flatness is defined as the ratio of the geometric mean to the arithmetic mean of the power spectral density components in each critical band for the input signal. In general, a low spectral flatness reveals a tone-like signal, whereas a high spectral flatness indicates a signal that is completely noise-like.

- **Spectral flux:** The Spectral Flux is also known as Delta Spectrum Magnitude, and it measures the local temporal variations of the sound. See [Tzanetakis and Cook, 2002] for a formal definition.

- **Spectral roll-off:** Spectral rolloff point is defined as the Nth percentile of the power spectral distribution, where N is usually 85% or 95%. The rolloff point indicate the frequencies below which the N% of the magnitude distribution is concentrated. This measure is useful in distinguishing voiced speech from unvoiced: unvoiced speech has a high proportion of energy contained in the high-frequency range of the spectrum, where most of the energy for voiced speech and music is contained in lower bands.

2.6.4.5 Unsupervised Feature Learning

A detailed review about the use of deep learning for audio processing is out of the scope of this thesis. However, due to its increasing success in audio processing [Hinton et al., 2012], in this section we draft some ideas about deep learning applied to audio processing.
Since recently, the state-of-the-art in some classic problems related to speech or music is based on *deep neural networks* (e.g. automatic speech recognition [Hinton et al., 2012] or onset detection [Böck et al., 2012]). A deep neural network (DNN) is an artificial neural network with multiple hidden layers of units between the input and output layers.
The use of neural networks for sound-related problems was popular during the 80s, but they were abandoned during the 90s because other machine learning algorithms worked better (such as Support Vector Machines [Burges, 1998]). Over the last

few years, advances in both machine learning algorithms and computer hardware have led to more efficients methods for training DNNs, and this fact has produced a promising revival of neural networks in sound-related problems.

Generally, in the case of deep learning applied to audio processing, the input of the DNN is a matrix comprised by the values of a set of filterbanks along several frames [Dahl et al., 2010] [Schlüter and Osendorfer, 2011] [Hinton et al., 2012] [Böck et al., 2012]. The output of these DNN is usually a reduced set of values, which represents the input information in a meaningful way for the task it has been trained for. These values are then used as features for more complex tasks, such as: phone recognition [Dahl et al., 2010], music/speech discrimination [Schlüter and Sonnleitner, 2012], music similarity estimation [Schlüter and Osendorfer, 2011], etc.

A relevant, comprehensive and updated overview on deep learning can be found in [LeCun et al., 2015].

2.7 Spectral Modeling Synthesis

Spectral Modeling Synthesis (SMS) technique was firstly proposed by [Serra, 1989], although it has evolved during the last decades with some variants of it [Serra and Smith, 2014]. In its original idea, SMS technique models time-varying spectra as (1) a collection of sinusoids controlled through time by piecewise linear amplitude and frequency envelopes (deterministic part), and (2) a time-varying filtered noise component (stochastic part). The analysis procedure first extracts the sinusoidal trajectories by tracking peaks in a sequence of short-time Fourier transforms. These peaks are then removed by spectral subtraction. The remaining "noise floor" is then modeled as white noise through a time-varying filter. A piecewise linear approximation to the upper spectral envelope of the noise is computed for each successive spectrum, and the stochastic part is synthesized by means of the overlap-add technique. This signal model is also called *deterministic plus stochastic*.

Recent tutorials about SMS [Serra and Smith, 2014] propose some variants of this model, each one of which is suitable for a different purpose. In this section, we focus on four of them: sinusoidal plus residual model (SpR) in Section 2.7.1, harmonic plus residual model (HpR) in Section 2.7.2, sinusoidal plus stochastic model (SpS) in Section 2.7.3 and harmonic plus stochastic model (HpS) in Section 2.7.4. In Section 2.7.5, some details about the practical implementation of SMS models is presented.

2.7.1 Sinusoidal Plus Residual Model (SpR)

This model assumes the signal can be modeled as a sum of sinusoids plus a residual component:

$$y[n] = \sum_{r=1}^{R} A_r[n] \cos(2\pi f_r[n] n + \phi_r) + x_{residual}[n] = y_{sinusoidal}[n] + x_{residual}[n] \quad (2.19)$$

where: R = number of sinusoidal components
$A_r[n]$ = instantaneous amplitude of sinusoid r
$f_r[n]$ = instantaneous frequency of sinusoid r
ϕ_r = initial phase of sinusoid r

However, the standard implementation of this model works in frequency domain[13], so a spectral expression might be closer to the actual workflow:

$$Y_l[k] = \sum_{r=1}^{R} A_{(r,l)} W[k - \hat{f}_{(r,l)}] e^{j\phi_{(r,l)}} + X_{residual\,l}[k] = Y_{sinusoidal\,l}[k] + X_{residual\,l}[k] \quad (2.20)$$

where: $W[k]$ = spectrum of analysis window
l = frame number
R_l = number of sinusoidal components in frame l
$A_{(r,l)}$ = amplitude of sinusoid r in frame l
$\hat{f}_{(r,l)}$ = normalized frequency of sinusoid r in frame l
$\phi_{(r,l)}$ = phase of sinusoid r in frame l
$Y_{sinusoidal\,l}[k]$ = sinusoidal component spectrum in frame l
$X_{residual\,l}[k]$ = residual component spectrum in frame l

The parameters of this model are obtained by a frame-wise peak-picking process on the magnitude spectrum. Typically, all local maxima above a threshold in the magnitude spectrum are considered peaks. In Figure 2.15, the use of SpR model with a sax sound is shown.

This model is useful to perform transformations affecting just the sinusoidal component of the sound (e.g. pitch shifting). It has been used in our approach for automatic dissonance reduction in polyphonic music (see Section 3.5).

[13] https://github.com/MTG/sms-tools

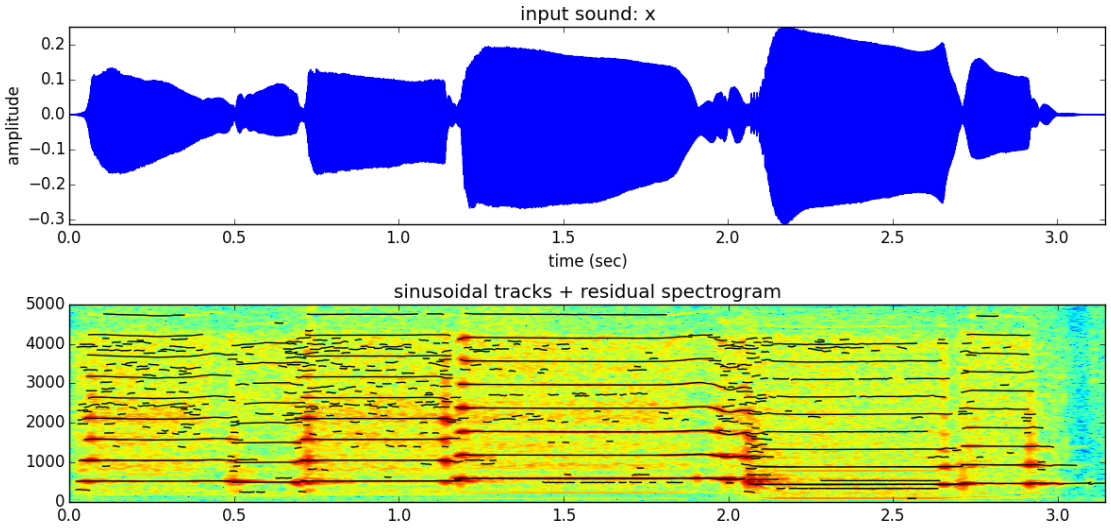

Figure 2.15: Example of sinusoidal plus residual modeling of an audio signal. This figure has been generated using the source code and audio samples provided in `https://github.com/MTG/sms-tools`. In this case, we have chosen SpR default parameters and the audio sample *sax-phrase-short.wav*.

In Figure 2.18 we show a comprehensive blocks diagram of the models presented along this section.

2.7.2 Harmonic Plus Residual Model (HpR)

In this model, the fundamental frequency f_0 of the signal is estimated in order to select only the harmonic peaks of the sound. The harmonics are selected by choosing the closest spectral peaks to positions $[f_0, 2f_0, 3f_0, \dots]$ and discarding the rest of them. The residual component contains the non-harmonic component.

This model is suitable for processing monophonic harmonic sounds (e.g. singing voice) when the residual component is not affected. For instance, this model allows to manipulate the f_0 contour of the input signal. However, it does not allow to modify the duration of the signal, since the residual component, which should be modified as well, can not be altered. In Figure 2.16, a sax sound has been modeled with a HpR model.

In the block diagram of Figure 2.18, the blocks *F0 detection* and *Harmonic detection* are used to perform the harmonic modeling of the input signal.

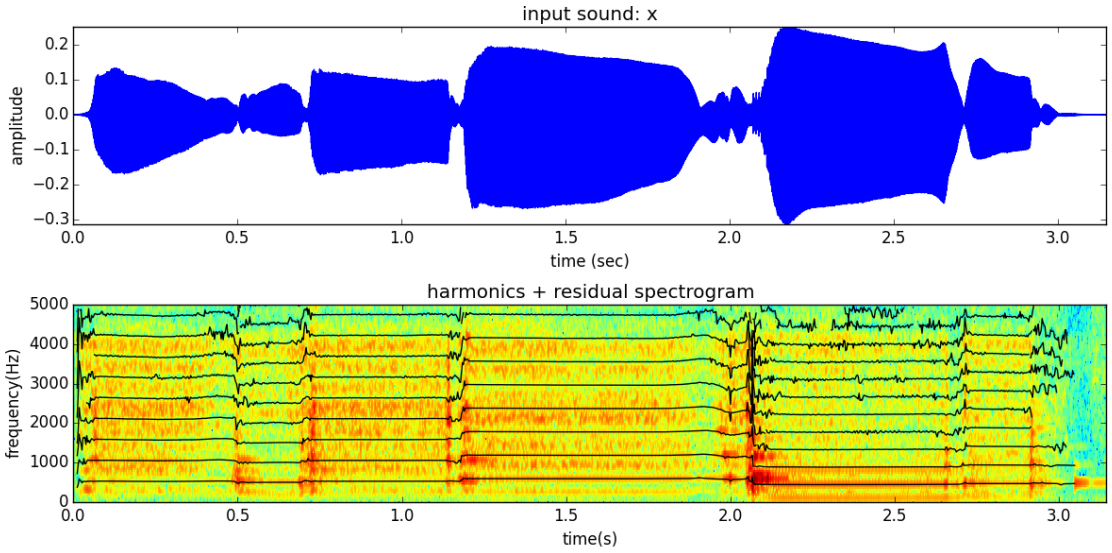

Figure 2.16: Example of harmonic plus residual modeling of an audio signal. As in Figure 2.15 this figure has been generated using the source code and audio samples provided in `https://github.com/MTG/sms-tools`. In this case, we have chosen HpR default parameters and the audio sample *sax-phrase-short.wav*.

2.7.3 Sinusoidal Plus Stochastic Model (SpS)

This model similar to the *sinusoidal plus residual*, but in this case a stochastic modeling of the residual component is performed. The stochastic modeling consists of white noise ($u[n]$), filtered by the same spectral envelope as the input signal (modeled by an impulse response $h[n]$).

$$y[n] = \sum_{r=1}^{R} A_r[n]\cos(2\pi f_r[n]n+\phi_r)+y_{stochastic}[n] = y_{sinusoidal}[n]+y_{stochastic}[n] \quad (2.21)$$

where: R = number of sinusoidal components
$A_r[n]$ = instantaneous amplitude of sinusoid r
$f_r[n]$ = instantaneous frequency of sinusoid r
ϕ_r = initial phase of sinusoid r

The stochastic component can be defined as:

$$y_{stochastic}[n] = \sum_{k=0}^{N-1} u[k]h[n-k] \quad (2.22)$$

where: $u[n]$ = white noise
$h[n]$ = impulse response of residual approximation
$y_{stochastic}[n]$ = stochastic component

Again, the same model can be expressed in frequency domain:

$$Y_l[k] = \sum_{r=1}^{R_l} A_{(r,l)} W[k - \hat{f}_{(r,l)}] e^{j\phi_{(r,l)}} + Y_{stochasticl}[k] = Y_{sinusoidall}[k] + Y_{stochasticl}[k] \tag{2.23}$$

where: l = frame number
$W[k]$ = spectrum of analysis window
R_l = number of sinusoidal components in frame l
$A_{(r,l)}$ = amplitude of sinusoid r in frame l
$\hat{f}_{(r,l)}$ = normalized frequency of sinusoid r in frame l
$\phi_{(r,l)}$ = phase sinusoid r in frame l

In frequency domain, the stochastic component is defined as:

$$Y_{stochasticl}[k] = |\tilde{X}_{residual_l}[k]| e^{j\angle U[k]} \tag{2.24}$$

where: $|\tilde{X}_{residual_l}[k]|$ = spectral envelope of residual in frame l
$\angle U[k]$ = spectral phases of noise (random)
l = frame number

This model is useful to perform transformations affecting both the sinusoidal and the stochastic components of the sound. For instance, time stretching requires modifying the duration of both components, and this can be performed using such stochastic modeling. Our system for realistic intensity variation of singing voice (Section 3.4) is based on the SpS model, because the sinusoidal and the stochastic components are separately modified.

The disadvantage of the SpS model with respect to the SpR model is that it is only suitable for sounds with a pure stochastic residual component (e.g. breathy sounds). If applied to other kind of signals, some artifacts can be perceived due to the stochastic modeling of a non-stochastic signal. In Figure 2.17, we show the stochastic modeling of a ocean sound, which is suitable for this kind of modeling due to its clearly stochastic nature.

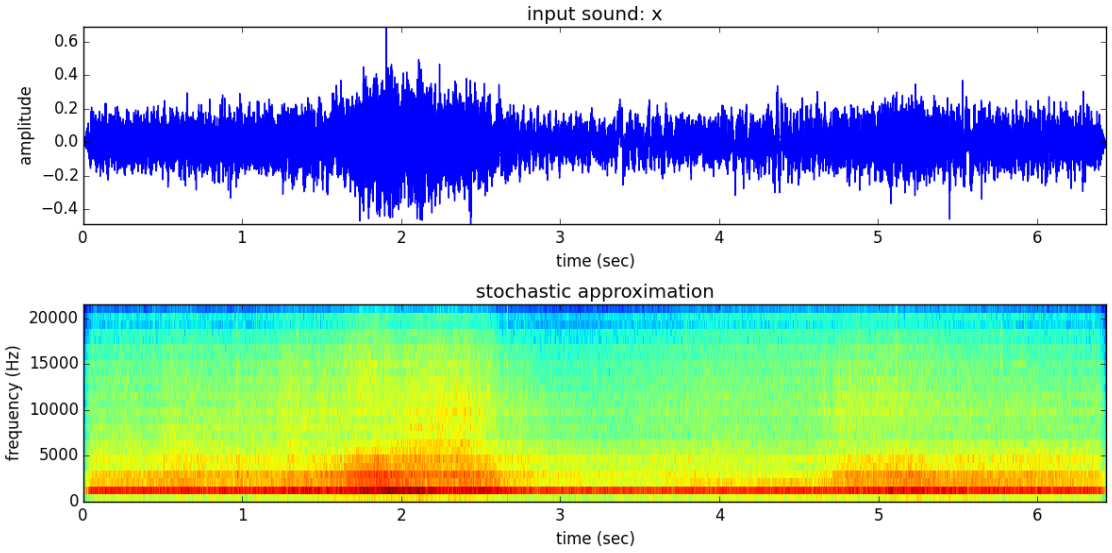

Figure 2.17: Example of stochastic modeling. As in Figure 2.15 this figure has been generated using the source code and audio samples provided in `https://github.com/MTG/sms-tools`. In this case, we have chosen stochastic model default parameters and the audio sample *ocean.wav*.

2.7.4 Harmonic Plus Stochastic Model (HpS)

This model is a variant of SpS model where, again, the fundamental frequency of the signal is estimated and only the harmonic peaks of the sound are considered. This model is only suitable for monophonic harmonic sounds with a pure stochastic residual component (e.g. a flute), but in exchange, it provides a rich source of transformations.
By using this model, many different transformations can be applied: to manipulate breathiness, to remove/introduce vibrato, to apply pitch-shifting or time-stretching, etc.

2.7.5 Implementation

The practical implementation works frame by frame, and takes several steps.

1. The frame $x_l[n]$ is windowed by a window function $w_l[n]$ (e.g. hann) in time domain to produce $xw_l[n]$.

2. The FFT of $xw_l[n]$ is computed to produce the spectrum $XW_l[k]$.

3. A peak picking process is applied to the frame spectrum. Any local maximum with magnitude value above a threshold t is considered a peak, and each peak is described with amplitude A, frequency f and phase p.

4. In case of harmonic model: the $f0$ of the frame is computed and the harmonics of the signal are chosen among the detected peaks. The rest of peaks are discarded from the harmonic component.

5. Any transformation of the peaks can be done at this point (e.g. pitch shifting or timbre processing).

6. The sinusoidal or harmonic spectrum of the signal is generated from the list of peaks to produce $Y_{\text{sinusoidal}_l}[n]$ or $Y_{\text{harmonic}_l}[n]$.

7. The sinusoidal/harmonic spectrum is subtracted to the spectrum of the original signal in order to produce the residual component $X_{\text{residual}_l}[n]$.

8. In the case of stochastic model: The spectral envelope of the residual component is approximated. Any transformation to such spectral envelope can be performed at this point. A random phase spectrum is applied to it to produce the stochastic component $Y_{\text{stochastic}_l}[n]$.

9. Both components, residual/stochastic and sinusoidal/harmonic, are summed and the inverse FFT is applied to it in order to resynthesize the output frame $yw_l[n]$.

The frames are then overlapped and added to compute the final waveform $y[n]$:

$$y[n] = \sum_{l=0}^{l=L} yw_l[n - lH] \tag{2.25}$$

where: $yw_l[n]$ = Windowed output for frame l
L = Number of frames
H = Hop size

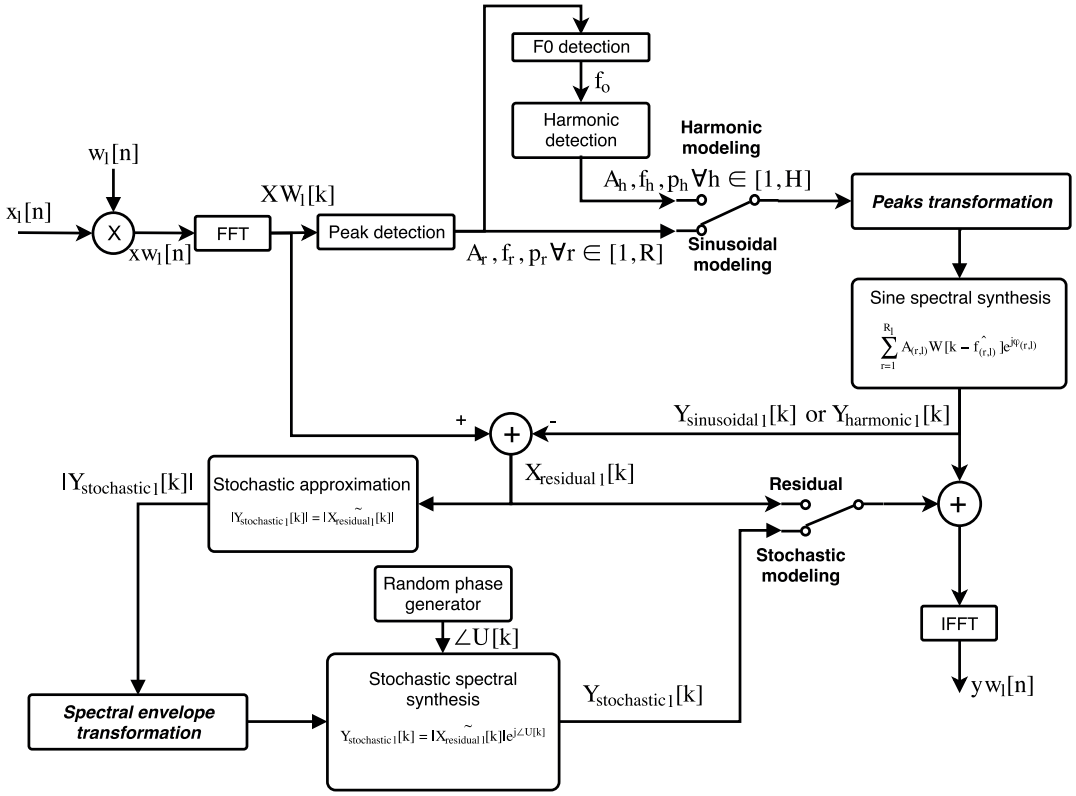

Figure 2.18: Block diagram of spectral modeling synthesis technique for one frame. Two switches are included to comprehend the four described models: sinusoidal plus residual, harmonic plus residual, sinusoidal plus stochastic and harmonic plus stochastic. Note that blocks *Peaks transformation* and *Spectral envelope transformation* are customizable and allow a large range of transformations to the original sound (pitch shifting, time stretching, timbre processing, etc.).

At this point, the most relevant background about topics addressed in this thesis has been presented. Specifically, we have presented an introduction about singing voice production (Section 2.1); a review on pitch estimation (Section 2.2); the state-of-the-art on singing transcription (Section 2.3); some basic concepts on Dynamic Time Warping (Section 2.4); a review on automatic singing assessment (Section 2.5); a review on timbre processing (Section 2.6), and a description of Spectral Modeling Synthesis (Section 2.7). In next chapter, a global summary of results achieved in this thesis is presented.

CHAPTER 3

Global Summary of Results

In Chapter 1, context and goals and this thesis were introduced, and in Chapter 2 a comprehensive review of the state-of-the-art in the topics related to this thesis was presented. In this chapter, we summarize the results and achievements of this thesis. The results achieved in this thesis are classified into the following topics:

Comparative analysis of F0 trackers for query-by-singing-humming (Section 3.1): This section presents a detailed comparative analysis between state-of-the-art pitch trackers for the specific context of query-by-singing-humming. It is a summary of [Molina et al., 2014d]:

- Molina, E., Tardón, L. J., Barbancho, I., and Barbancho, A. M. (2014). The importance of F0 tracking in query-by-singing-humming. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 277-282, Taipei (Taiwan).

Singing transcription (Section 3.2): This section presents an evaluation framework for singing transcription and a novel approach for note transcription of singing voice. It summarizes publications [Molina et al., 2014b] and [Molina et al., 2015]:

- Molina, E., Barbancho, A. M., Tardón, L. J., and Barbancho, I. (2014). Evaluation framework for automatic singing transcription. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 567-572, Taipei (Taiwan).

- Molina, E., Tardón, L. J., Barbancho, A. M., and Barbancho, I. (2015). SiPTH: Singing transcription based on hysteresis defined on the pitch-time curve. *IEEE/ACM Transactions on Acoustics, Speech and Language Processing*, 23(2):252-263.

Automatic singing assessment (Section 3.3): This section presents two variants of a novel approach for singing skill assessment: one approach based on pitch contour similarity, and another one based on note-based similarity. It is a summary of [Molina et al., 2013]:

- Molina, E., Barbancho, I., Gómez, E., Barbancho, A. M., and Tardón, L. J. (2014). Fundamental frequency alignment vs. note-based melodic similarity for singing voice assessment. In *Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, pages 744-748, Vancouver (Canada).

Timbre analysis and processing (Section 3.4): This section presents a parametric spectral-envelope model for singing voice, together with an approach for synthesizing realistic intensity variations based on this model. It summarizes [Molina et al., 2014c]:

- Molina, E., Barbancho, I., Barbancho, A. M., and Tardón, L. J. (2014). Parametric model of spectral envelope to synthesize realistic intensity variations in singing voice. In *Proceedings of the 39th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, pages 634-638, Florence (Italy).

Dissonance reduction in polyphonic audio (Section 3.5): Finally, this section presents a method for dissonance reduction in polyphonic music (applicable also to choir music). It is a summary of [Molina et al., 2014a]:

- Molina, E., Barbancho, A. M., Tardón, L. J., and Barbancho, I. (2014). Dissonance reduction in polyphonic music using harmonic reorganization. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 22(2):325-334.

In each of the results presented in this chapter, the methodology used to achieve the specific research goals is briefly described. This methodology, in general terms consists of: (i) a review of the state-of-the-art and background knowledge about the topic, (ii) an approach proposal to address the related problem, (iii) the evaluation of the proposed approach and a comparison with respect to other state-of-the-art approaches, and (iv) a discussion of the results and conclusions.

3.1 Comparative Analysis of F0 Trackers for Query-by-Singing-Humming

In this section a comparative study of several state-of-the-art F0 trackers applied to the context of query-by-singing-humming (QBSH) is presented. This study has been carried out using the well known, freely available, MIR-QBSH dataset[1] in different conditions of added pub-style noise and smartphone-style distortion [Mauch and Ewert, 2013]. For audio-to-MIDI melodic matching, we have used two state-of-the-art systems and a simple, easily reproducible baseline method. For evaluation, we measured the QBSH performance for 189 different combinations of F0 tracker, noise/distortion conditions and matcher. In Figure 3.1, the scheme of our study is shown. Additionally, the overall accuracy of the F0 transcriptions (as defined in MIREX[2]) was also measured. In the results, we found that F0 tracking overall accuracy correlates with QBSH performance, but it does not totally measure the suitability of a pitch vector for QBSH. In addition, we also found clear differences in robustness to F0 transcription errors between different matchers.

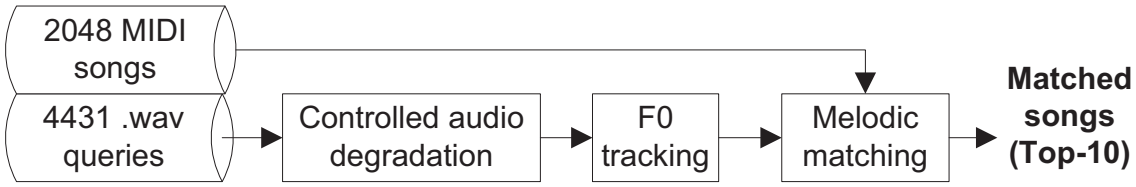

Figure 3.1: Overall scheme of our study in the context of query-by-singing-humming

This section is a summary of the content presented in [Molina et al., 2014d]:

Molina, E., Tardón, L. J., Barbancho, I., and Barbancho, A. M. (2014). The importance of F0 tracking in query-by-singing-humming. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 277-282, Taipei (Taiwan).

Specifically, this summary does not include information about the parameters chosen for F0 tracking, and the technical description of the baseline melodic matcher has been simplified.

[1] http://mirlab.org/dataSet/public/
[2] http://www.music-ir.org/mirex

3.1.1 Algorithms Evaluated

In this section, the algorithms considered for F0 tracking and for melody matching are enumerated.

3.1.1.1 F0 Trackers

Eight different F0 trackers have been considered in our comparative study:

- YIN [De Cheveigné and Kawahara, 2002]: It resembles the idea of the autocorrelation method [Rabiner, 1977] but it uses the cumulative mean normalized difference function, which peaks at the local period with lower error rates than the traditional autocorrelation function.
- pYIN [Mauch, 2014]: It adds a HMM-based F0 tracking stage in order to find a "smooth" path through the fundamental frequency candidates obtained by Yin.
- AC-DEFAULT [Boersma, 1993] (default configuration of parameters in Praat[3]): It is based on the autocorrelation method, but it improves it by considering the effects of the window during the analysis and by including a F0 tracking stage based on dynamic programming
- AC-ADJUSTED: Same algorithm as AC-DEFAULT with a manually adjusted configuration of parameters. They have been adjusted using several random samples extracted from MIR-QBSH dataset. More details are found in [Molina et al., 2014d].
- AC-LEIWANG [Doreso, 2013]: It is based on [Boersma, 1993], but it uses a finely tuned set of parameters and a post-processing stage in order to mitigate spurious and octave errors.
- SWIPE' [Camacho and Harris, 2008]: This algorithm estimates the pitch as the fundamental frequency of the sawtooth waveform whose spectrum best matches the spectrum of the input signal.
- MELODIA-MONO [Salamon and Gómez, 2012]: This system is based on the creation and characterization of pitch contours, which are time continuous sequences of pitch candidates grouped using auditory streaming cues. Melodic and non-melodic contours are distinguished depending on the distributions of its characteristics. In this case, the parameters preset *Monophonic* of MELODIA Vamp plugin[4] has been chosen.

[3] www.fon.hum.uva.nl/praat/

[4] http://mtg.upf.edu/technologies/melodia

- MELODIA-POLY: Same algorithm as MELODIA-MONO, but with parameters adjusted to default preset *Polyphonic*.

3.1.1.2 Audio-to-MIDI Melodic Matchers

Three algorithms for melodic matching are considered: a simple baseline approach, MusicRadar and NetEase.

Description of baseline approach

The baseline approach for audio-to-MIDI matching is based in f_0 contour alignment using DTW (see Figure 3.2), and it is freely available[5] to allow reproducibility of results. For each MIDI file in the database, several f_0 contours are extracted (hopsize 0.01s) with various lengths: 5, 6, 7, 8, 9, 10 and 11 seconds (all of them from the beginning of the song). Then, they all are resampled to 50 points vectors and zero-mean normalized. On the other hand, the f_0 contour from the audio query is equally extracted, resampled to 50 points and zero-mean normalized. Then, DTW is applied to find the alignment cost between the query and each reference excerpt. Finally, top-10 song with smallest alignment cost are reported.

MusicRadar

MusicRadar [Doreso, 2013] is a state-of-the-art algorithm for melodic matching, which participated in MIREX 2013 and obtained the best accuracy in all datasets, except for the case of IOACAS[6]. It is the latest evolution of a set of systems developed by Lei Wang since 2007 [Wang et al., 2008] [Wang et al., 2010]. The system takes advantage of several matching methods to improve its accuracy. First, Earth Mover's Distance (EMD), which is note-based and fast, is adopted to eliminate most unlikely candidates. Then, Dynamic Time Warping (DTW), which is frame-based and more accurate, is executed on these surviving candidates. Finally, a weighted voting fusion strategy is employed to find the optimal match. In our study, we have used the exact melody matcher tested in MIREX 2013, provided by its original author.

NetEase

NetEase's approach [Li et al., 2013] is a state-of-the-art algorithm for melodic matching, which participated in MIREX 2013 and obtained the first position for IOACAS dataset[6], as well as relevant results in the rest of datasets. This algorithm adopts

[5] www.atic.uma.es/ismir2014qbsh/

[6] http://mirlab.org/dataSet/public/IOACAS_QBH.rar

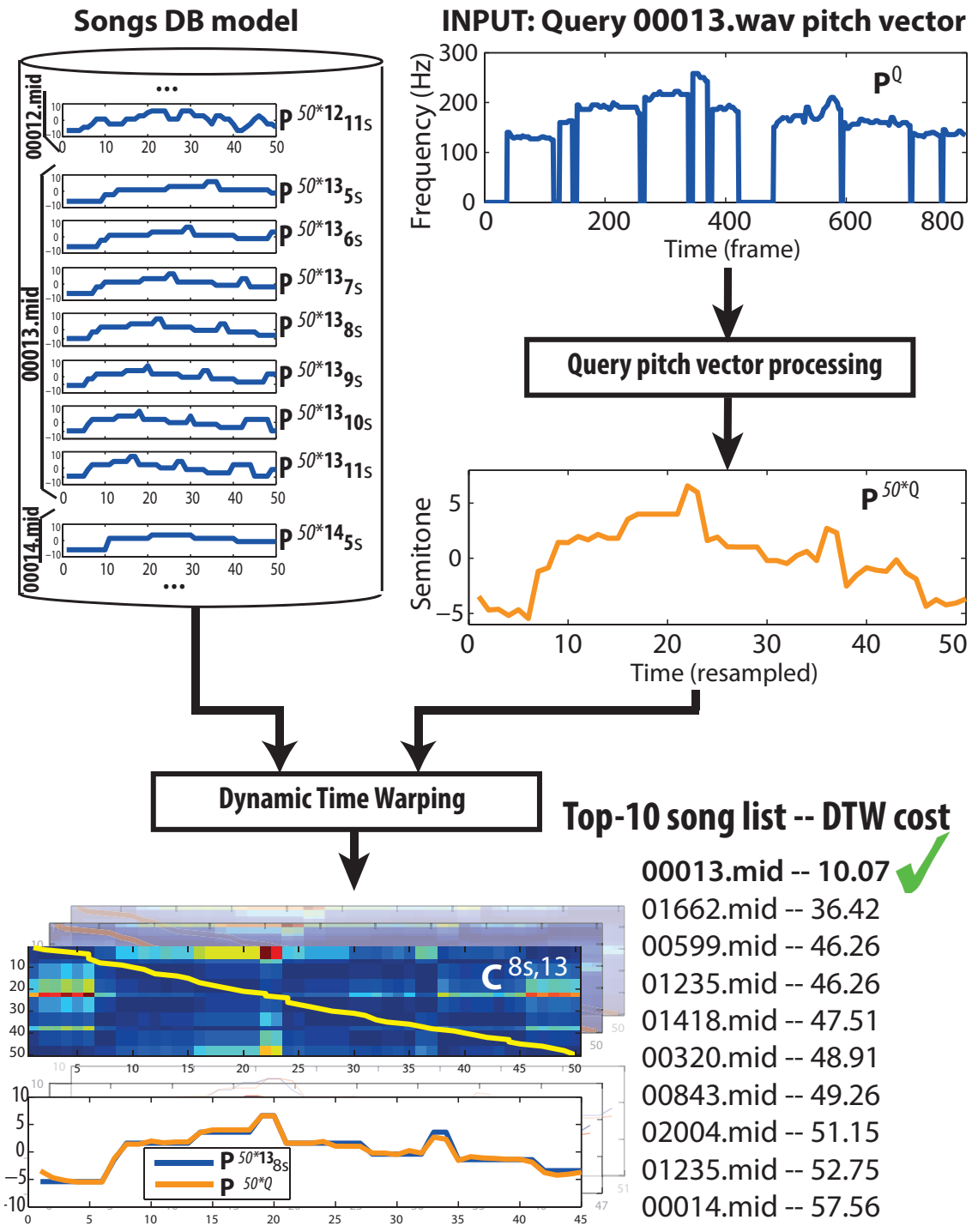

Figure 3.2: Scheme of the proposed baseline method for audio-to-MIDI melody matching

a two-stage cascaded solution based on Locality Sensitive Hashing (LSH) and accurate matching of frame-level pitch sequence. Firstly, LSH is employed to quickly filter out songs with low matching possibilities. In the second stage, Dynamic Time Warping is applied to find the M (set to 10) most matching songs from the candidate list. Again, the original authors of NetEase's approach (who also authored some older works on query-by-humming [Li et al., 2008]) collaborated in this study, so we have used the exact melody matcher tested in MIREX 2013.

3.1.2 Evaluation Strategy

In this section, we present the datasets used in our study (Section 3.1.2.1), the way in which we have combined F0 trackers and melody matchers (Section 3.1.2.2) and the chosen evaluation measures (Section 3.1.2.3).

3.1.2.1 Datasets

We have used the public corpus MIR-QBSH[1] (used in MIREX since 2005), which includes 4431 .wav queries corresponding to 48 different MIDI songs. The audio queries are 8 seconds length, and they are recorded in mono 8 bits, with a sample rate of 8kHz. In general, the audio queries are monophonic with no background noise, although some of them are slightly noisy and/or distorted. This dataset also includes a manually corrected pitch vector for each .wav query. Although these annotations are fairly reliable, they may not be totally correct, as stated in MIR-QBSH documentation.

In addition, we have used the Audio Degradation Toolbox [Mauch and Ewert, 2013] in order to recreate common environments where a QBSH system could work. Specifically, we have combined three levels of pub-style added background noise (`PubEnvironment1` sound) and smartphone-style distortion (`smartPhoneRecording` degradation), leading to a total of seven evaluation datasets: (1) Original MIR-QBSH corpus (2) 25 dB SNR (3) 25 dB SNR + smartphone distortion (4) 15 dB SNR (5) 15 dB SNR + smartphone distortion (6) 5 dB SNR (7) 5 dB SNR + smartphone distortion. Note that all these degradations have been checked in order to ensure perceptually realistic environments.

Finally, in order to replicate MIREX conditions, we have included 2000 extra MIDI songs (randomly taken from ESSEN collection[7]) to the original collection of 48 MIDI songs, leading to a songs collection of 2048 MIDI songs. Note that, although these 2000 extra songs fit the style of the original 48 songs, they do not correspond to any .wav query of MIR-QBSH dataset.

[7] www.esac-data.org/

3.1.2.2 Combinations of F0 Trackers and Melody Matchers

For each of the 7 datasets, the 4431 .wav queries have been transcribed using the 8 different F0 trackers mentioned in Section 3.1.1.1. Additionally, each dataset also includes the 4431 manually corrected pitch vectors of MIR-QBSH as a reference, leading to a total of 279153 pitch vectors. Then, all these pitch vectors have been used as input to the 3 different melody matchers mentioned in Section 3.1.1.2, leading to 930510 lists of top-10 matched songs. Finally, these results have been used to compute a set of meaningful evaluation measures.

3.1.2.3 Evaluation Measures

In this section, we present the evaluation measures used in this study:
(1) Mean overall accuracy of F0 tracking ($\overline{\mathbf{Acc_{ov}}}$): For each pitch vector we have computed an evaluation measures defined in MIREX Audio Melody Extraction task: *overall accuracy* ($\mathrm{Acc_{ov}}$) (a definition can be found in [Salamon and Gómez, 2012]). The *mean overall accuracy* is then defined as

$$\overline{\mathrm{Acc_{ov}}} = (1/N) \sum_{i=1}^{N} \mathrm{Acc_{ov}}_i \tag{3.1}$$

where: N = total number of queries considered
$\mathrm{Acc_{ov}}_i$ = overall accuracy of pitch vector for i:th query

We have selected this measure because it considers both voicing and pitch, which are important aspects in QBSH. For this measure, our ground truth consists of the manually corrected pitch vectors of the .wav queries, which are included in the original MIR-QBSH corpus.
(2) Mean Reciprocal Rank (MRR): This measure is commonly used in MIREX Query By Singing Humming task, and it is defined as

$$\mathrm{MRR} = (1/N) \sum_{i=1}^{N} r_i^{-1} \tag{3.2}$$

where: N = total numbers of queries considered
r_i = rank of the correct answer for i:th query

Note that in case r_i is higher than 10, then we set $r_i = 0$.

3.1.3 Results & Discussion

In this section, we present the obtained results and some relevant considerations about them.

3.1.3.1 $\overline{\text{Acc}_{\text{ov}}}$ and MRR for each F0 tracker - Dataset - Matcher

In Table 1, we show the $\overline{\text{Acc}_{\text{ov}}}$ and the MRR obtained for the whole dataset of 4431 .wav queries in each combination of F0 tracker-dataset-matcher (189 combinations in total). Note that these results are directly comparable to MIREX Query by Singing/Humming task[6] (MIR-QBSH dataset, also known as Jang's dataset).
As expected, the manually corrected pitch vectors produce the best MRR in most cases ($\overline{\text{Acc}_{\text{ov}}}$ is 100% because it has been taken as the ground truth for such measure). Note that, despite manual annotations are the same in all datasets, NetEase and MusicRadar matchers do not produce the exact same results in all cases. It is due to the generation of the indexing model (used to reduced the time search), which is not a totally deterministic process.
Regarding the relationship between $\overline{\text{Acc}_{\text{ov}}}$ and MRR in the rest of F0 trackers, we find a somehow contradictory result: the best $\overline{\text{Acc}_{\text{ov}}}$ does not always correspond with the best MRR. This fact may be due to two different reasons:

- The meaning of $\overline{\text{Acc}_{\text{ov}}}$ may be distorted due to annotation errors in the ground truth (as mentioned in Section 3.1.2.1), or to eventual intonation errors in the dataset. However, manual annotations produce the best MRR, what suggests that the amount of these types of errors is low.

- The measure $\overline{\text{Acc}_{\text{ov}}}$ itself is not totally representative of the suitability of a pitch vector for QBSH. Let's illustrate this fact through an example: in Figure 3.3 we show two different pitch vectors with same overall accuracy $\overline{\text{Acc}_{\text{ov}}} = 82.91\%$. However, pitch vector (a) matches the right song with rank $r_i = 1$ whereas pitch vector (b) does not matches the right song at all ($r_i \geq 11$). The reason is that $\overline{\text{Acc}_{\text{ov}}}$ do not take into account the pitch values of false positives, but in fact they are important for QBSH. Therefore, we conclude that the high MRR achieved by some F0 trackers (AC-LEIWANG when background noise is low, and PYIN for highly degraded signals), is not only due to the amount of errors made by them, but also to the type of such errors.

Additionally, we observed that, in most cases, the queries are matched either with rank $r_i = 1$ or $r_i \geq 11$ (intermediate cases such as rank $r_i = 2$ or $r_i = 3$ are less frequent). Therefore, the variance of ranks is generally high, and their distribution is not Gaussian.

F0 tracker	Clean dataset	25dB SNR	25 dB SNR + distortion	15dB SNR	15 dB SNR + distortion	5dB SNR	5 dB SNR + distortion
(A)	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.89 / 0.95	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.88 / 0.95
(B)	89 / **0.80** / **0.89** / **0.96**	89 / **0.80** / **0.89** / **0.96**	**88** / **0.80** / **0.88** / **0.95**	88 / **0.79** / **0.88** / **0.94**	84 / 0.71 / 0.86 / 0.94	78 / 0.50 / 0.73 / 0.85	67 / 0.33 / 0.57 / 0.73
(C)	**90** / 0.74 / 0.85 / 0.94	90 / 0.71 / 0.85 / 0.92	86 / 0.72 / 0.84 / 0.92	89 / 0.71 / 0.84 / 0.92	85 / 0.66 / 0.81 / 0.89	72 / 0.49 / 0.58 / 0.70	64 / 0.26 / 0.39 / 0.51
(D)	90 / 0.71 / 0.83 / 0.92	**90** / 0.74 / 0.85 / 0.93	85 / 0.74 / 0.85 / 0.94	**90** / 0.78 / 0.87 / 0.94	**85** / **0.77** / **0.87** / **0.94**	79 / **0.69** / **0.79** / **0.87**	72 / **0.58** / **0.69** / **0.81**
(E)	89 / 0.71 / 0.83 / 0.92	89 / 0.71 / 0.84 / 0.92	84 / 0.66 / 0.80 / 0.91	88 / 0.72 / 0.84 / 0.93	83 / 0.65 / 0.80 / 0.91	75 / 0.67 / 0.67 / 0.82	66 / 0.48 / 0.53 / 0.73
(F)	86 / 0.62 / 0.81 / 0.89	86 / 0.70 / 0.83 / 0.92	81 / 0.64 / 0.78 / 0.89	82 / 0.60 / 0.77 / 0.88	75 / 0.50 / 0.67 / 0.82	48 / 0.03 / 0.08 / 0.04	44 / 0.04 / 0.04 / 0.03
(G)	88 / 0.56 / 0.81 / 0.88	87 / 0.47 / 0.79 / 0.86	83 / 0.47 / 0.76 / 0.85	86 / 0.39 / 0.78 / 0.87	81 / 0.35 / 0.73 / 0.82	70 / 0.11 / 0.32 / 0.52	63 / 0.04 / 0.20 / 0.38
(H)	87 / 0.66 / 0.83 / 0.87	87 / 0.67 / 0.82 / 0.87	83 / 0.64 / 0.78 / 0.84	86 / 0.66 / 0.81 / 0.84	82 / 0.58 / 0.74 / 0.80	83 / 0.51 / 0.73 / 0.75	73 / 0.32 / 0.55 / 0.62
(I)	84 / 0.62 / 0.76 / 0.86	84 / 0.62 / 0.76 / 0.86	79 / 0.50 / 0.64 / 0.74	84 / 0.63 / 0.76 / 0.86	79 / 0.50 / 0.65 / 0.75	**83** / 0.60 / 0.73 / 0.83	**75** / 0.39 / 0.55 / 0.65

Table 3.1: F0 overall accuracy and MRR obtained for each case. F0 trackers: (A) *MANUALLY CORRECTED* (B) *AC-LEIWANG* (C) *AC-ADJUSTED* (D) *PYIN* (E) *SWIPE'* (F) *YIN* (G) *AC-DEFAULT* (H) *MELODIA-MONO* (I) *MELODIA-POLY*. The format of each cell is: $\overline{\text{Acc}_{\text{ov}}}(\%)$ / *MRR-baseline* / *MRR-NetEase* / *MRR-MusicRadar*.

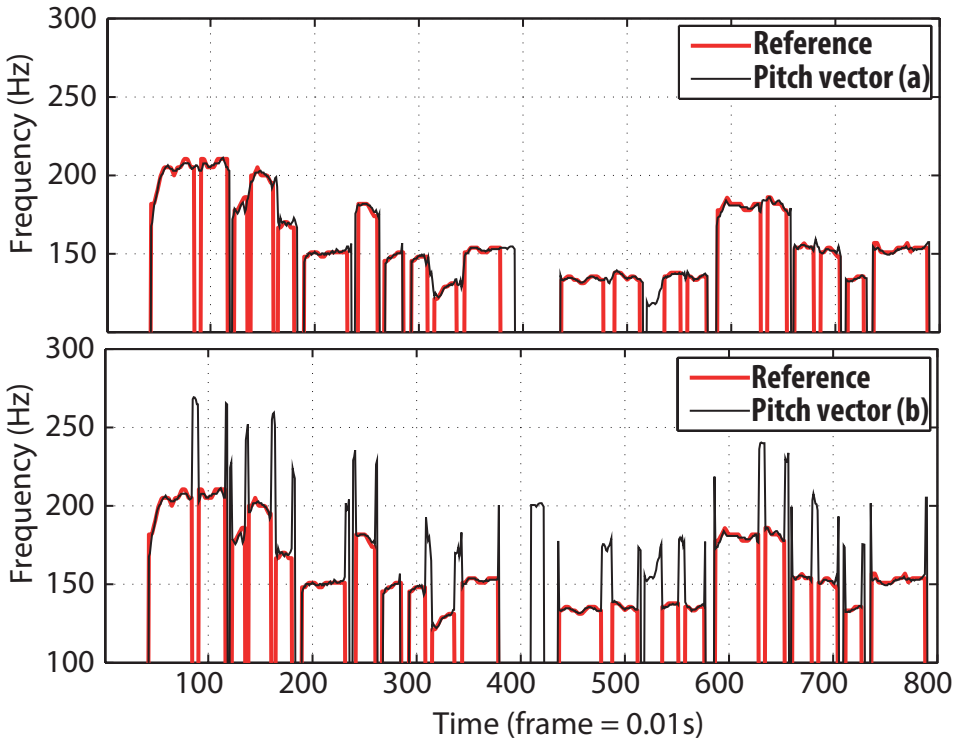

Figure 3.3: According to MIREX measures, these two pitch vectors (manually manipulated) are equally accurate: *voicing recall* = 99.6%, *voicing false-alarm* = 48.4%, *raw pitch accuracy* = 82.91%, *raw chroma accuracy* = 97.41%, *overall accuracy* = 82.91%. However, pitch vector (a) is much more suitable than pitch vector (b) for QBSH.

3.1.3.2 MRR vs. $\overline{\text{Acc}_{\text{ov}}}$ for each matcher

In order to study the robustness of each melodic matcher to F0 tracking errors, we have represented the MRR obtained by each one for different ranges of $\overline{\text{Acc}_{\text{ov}}}$ (Figure 3.4). For this experiment, we have selected only the .wav queries which produce the right answer in first rank for the three matchers considered (baseline, Music Radar and NetEase) when manually corrected pitch vectors are used (around a 70% of the dataset matches this condition). In this way, we ensure that bad singing or a wrong manual annotation is not affecting the variations of MRR in the plots. Note that, in this case, the results are not directly comparable to the ones computed in MIREX (in contrast to the results shown in Section 3.1.3.1, which are directly comparable). Regarding the obtained results (shown in Figure 3.4), we observe clear differences in the robustness to F0 estimation errors between matchers, what is coherent with the results presented in Table 3.1. The main difference is found in the baseline

matcher with respect to both NetEase and Music Radar. Given that the baseline matcher only uses DTW, whereas the other two matchers use a combination of various searching methods, we hypothesise that such combination may improve their robustness to F0 tracking errors. However, further research is needed to really test this hypothesis.

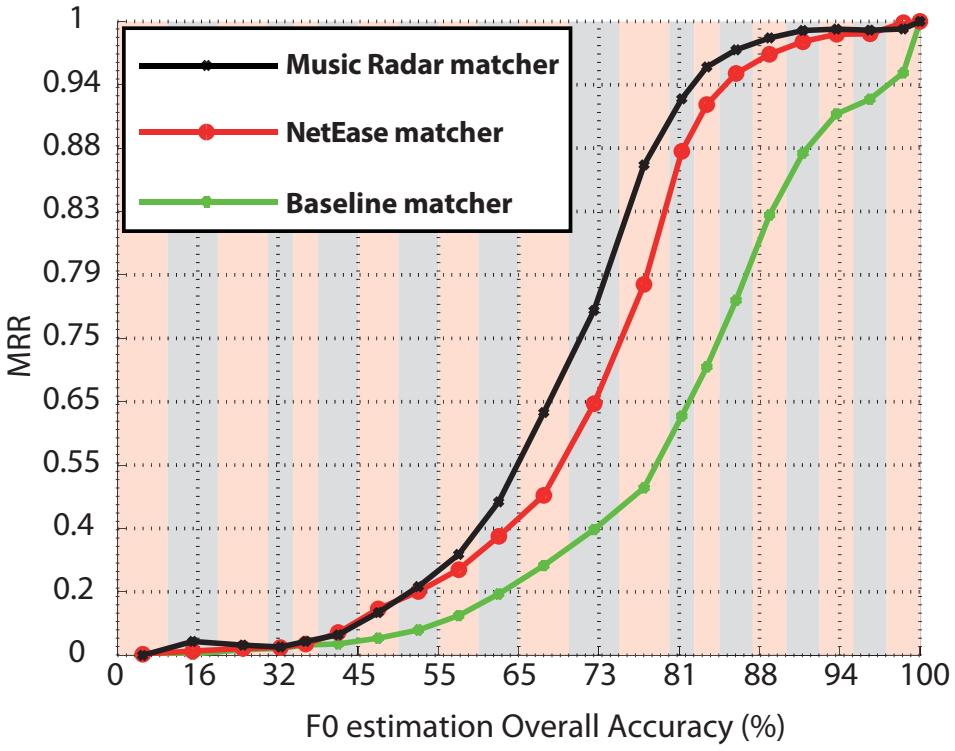

Figure 3.4: MRR obtained for each range of Overall Accuracy (each range is marked with coloured background rectangles). We have considered only the .wav queries which, using manually corrected F0 vectors, produce MRR = 1 in all matchers. Note: The axis have been manually manipulated in order to better visualize the differences between curves.

3.2 Singing Transcription

In this section, we describe our contributions related to the topic of singing transcription. As described in Section 2.3, singing transcription refers to the task of automatically generating a symbolic representation (e.g. MIDI notes) from the audio signal. In this thesis two main contributions are presented in relationship with

such topic: (1) the method SiPTH for note transcription (Section 3.2.1) and (2) an evaluation framework for singing transcription (Section 3.2.2). In Section 3.2.3, we present the results achieved by our SiPTH algorithm with respect to several state-of-the-art methods using the proposed evaluation framework.

3.2.1 SiPTH: Singing Transcription

In this section we describe a novel method for monophonic singing transcription based on hysteresis defined on the pitch-time curve, which has been published in [Molina et al., 2015]:

Molina, E., Tardón, L. J., Barbancho, A. M., and Barbancho, I. (2015). SiPTH: Singing transcription based on hysteresis defined on the pitch-time curve. *IEEE/ACM Transactions on Acoustics, Speech and Signal Processing*, 23(2):252-263.

Specifically, in this section we provide the key concepts of the SiPTH method in a simple and clear way, without providing deep details about the parameters chosen or the exact definitions of the terms used (e.g. *chroma contour*), which can be found in the paper.

Our approach implements an interval-based note segmentation through a hysteresis process on the pitch-time curve, which is obtained using Yin algorithm [De Cheveigné and Kawahara, 2002]. The exact definition of hysteresis varies from area to area and from paper to paper [Mayergoyz, 1986], 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 singing transcription task addressed and to the pitch-time hysteresis effect considered to perform note segmentation.
The selected approach for singing transcription can be summarized into the following steps:

1. **Chroma contour estimation**: First, the regions where chroma feature is stable are isolated, since they are candidates to sung notes (Figure 3.5).

2. **Voice/Unvoiced classification**: Then, each chroma contour is classified into two classes: voiced or unvoiced. This classification is performed with a previously trained tree classifier using two descriptors: aperiodicity and energy.

3. **Interval-based transcription:** Note segmentation based on pitch intervals is carried out. To this end, a dynamic averaging of the pitch curve is performed

for each growing note in order to roughly estimate its average pitch value. This dynamic average $\mathrm{F0}_A(l)$ is computed using the following expression:

$$\mathrm{F0}_A(l) = \frac{\sum_{k=l_0}^{l} \mathrm{F0}(k)}{l - l_0 + 1} \tag{3.3}$$

where l_0 is the index corresponding to either the beginning of a new note, or the beginning of a voiced region. Deviations of the instantaneous 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 (Figure 3.6).

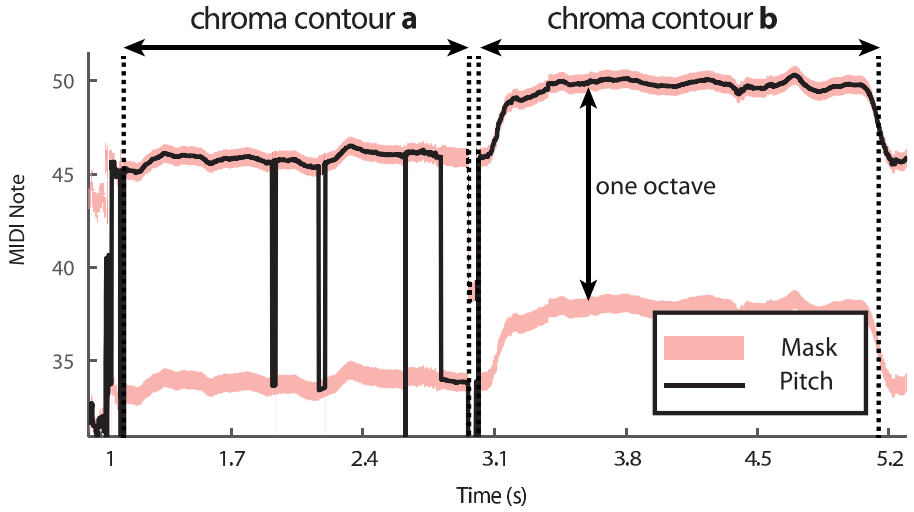

Figure 3.5: Chroma contours estimation. The black curve represents the estimated pitch value; red regions represent the mask where pitch values can vary between consecutive frames. The use of a mask that allows fast octave jumps avoids fake note changes if octave errors happen.

4. **Note labelling:** Finally, each note is labelled using three values: pitch, onset and offset. The onset and the offset instants are directly obtained from the note changes produced by the previous step. The pitch value is computed by using an energy weighted α-trimmed mean [Bednar and Watt, 1984]. In this weighted mean, the extreme (high and low) values of F0 (typically outliers) are discarded.

This method has been evaluated using the evaluation framework for singing voice presented in Section 3.2.2. In Section 3.2.3 we present the results obtained by SiPTH method in comparison with some state-of-the-art methods for singing transcription.

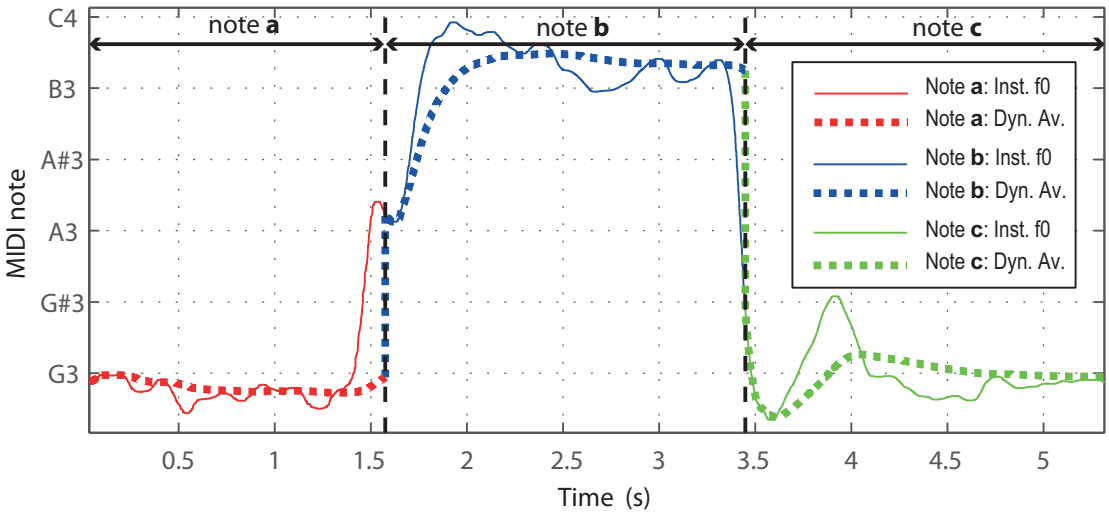

Figure 3.6: Representation of the hysteresis process for the detection of note changes. Samples are taken from real data: from $\approx G3$ to $\approx B3$ to $\approx G3$. The instantaneous F0 and the dynamic average $F0_A$ for each note are shown. Strong and/or sustained deviations of the instantaneous 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.

3.2.2 Evaluation Framework for Singing Transcription

Given the lack of standard evaluation strategies for singing transcription, in this thesis, a comprehensive evaluation framework is proposed. It consists of a cross-annotated dataset of 1154 seconds and a set of extended evaluation measures, which are integrated in a Matlab toolbox. The presented evaluation measures are based on standard MIREX note-tracking measures, but they provide extra information about the type of errors made by the singing transcriber.
This evaluation framework is freely available[8], and a detailed description of it can be found in [Molina et al., 2014b]:

Molina, E., Barbancho, A. M., Tardón, L. J., and Barbancho, I. (2014). Evaluation framework for automatic singing transcription. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Taipei (Taiwan).

[8]http://www.atic.uma.es/ismir2014singing

Specifically, in this section we skip the description of previous evaluation approaches, which are presented in the paper, and we focus on the key aspects of the proposed evaluation methodology.

3.2.2.1 Proposed Dataset

The proposed dataset consists of 38 melodies sung untrained singers (men, women and children), recorded in mono with a sample rate of 44100Hz 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 (14 traditional melodies), adult male (13 pop melodies) and adult female (11 pop melodies).

The described music collection has been manually annotated to build the ground truth. First, we have transcribed the audio recordings with a baseline algorithm, and then all the transcription errors have been corrected by an expert musician with more than 10 years of music training. Then, a second expert musician (with 7 years of music training) checked all the annotations until both musicians agreed. The transcription errors were corrected by listening to the synthesized transcription and the original audio simultaneously.

3.2.2.2 Evaluation Measures

The proposed evaluation framework has been included in a Matlab toolbox, which can be used with a GUI (Figure 3.7) or via command line. In this framework, three different definitions of correct note are proposed:

- Correct Onset, Pitch and Offset (COnPOff): This is a standard correctness criteria, since it is used in MIREX[9] (*Multiple F0 estimation and tracking* task), and it is the most restrictive one.

- Correct Onset, Pitch (COnP): This criteria is also used in MIREX, but it is less restrictive since it just considers onset and pitch, and ignores the offset value.

- Correct Onset (COn): We have also included the evaluation criteria used in MIREX *Audio Onset Detection* task.

[9] www.music-ir.org/mirex

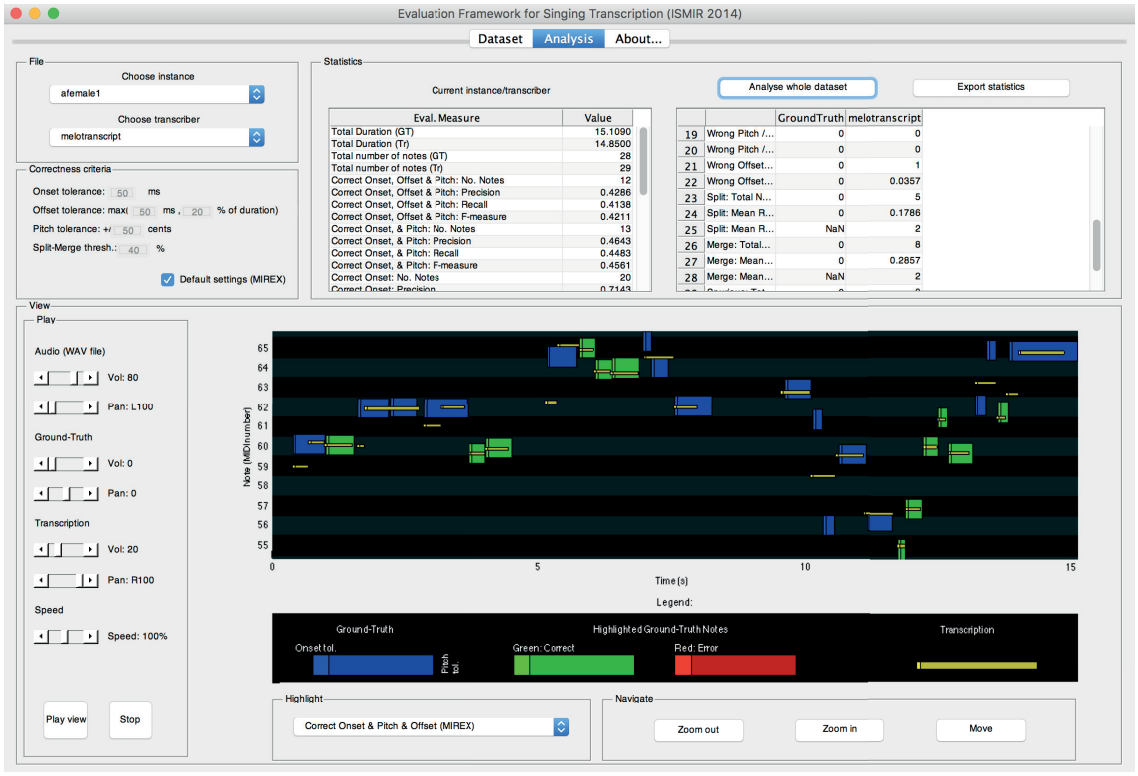

Figure 3.7: GUI for the proposed evaluation framework

Additionally, we have defined several types of errors:

- Only Bad Onset (OBOn): A note with correct pitch and offset, bad incorrect onset value.
- Only Bad Pitch (OBP): A note with correct onset and offset, but incorrect pitch value.
- Only Bad Offset (OBOff): A note with correct onset and pitch, but incorrect offset.
- Split (S): A split note refers to a note that has been incorrectly segmented into different consecutive notes.
- Merged (M): A set of consecutive notes that have been incorrectly merged into a single note.
- Spurious notes (PU): A transcribed note not corresponding to any ground-truth note.

- Non-detected notes (ND): A ground-truth note not corresponding to any transcribed note.

In Figure 3.8, these categories are illustrated through some examples.

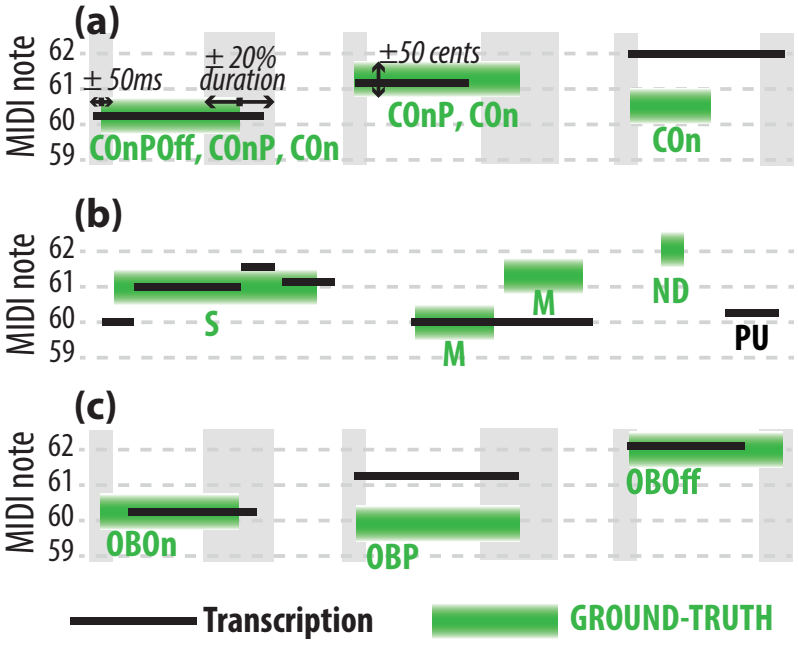

Figure 3.8: Examples of the proposed note categories

3.2.3 Results & Discussion

In this section, we provide the results of evaluating SiPTH method (Section 3.2.1) together with several state-of-the-art methods using the described evaluation framework in Section 3.2.2.

Compared Algorithms

The singing transcription algorithms considered in our evaluation are:

SiPTH [Molina et al., 2015]: Proposed method for singing transcription described in Section 3.2.1, which is based on interval-based segmentation using a hysteresis process on the pitch-time curve of the audio signal.

Gómez & Bonada [Gómez et al., 2013]: It consists of three main steps: tuning-frequency estimation, transcription into short notes, and an iterative process involving note consolidation and refinement of the tuning frequency. For the experiment, we have used a standalone binary provided by the authors of the algorithm.

Ryynänen [Ryynänen, 2008]: We have used Ryynänen's method for automatic transcription of melody, bass line and chords in polyphonic music, although we only focus on melody transcription. The used version is the latest evolution of the original HMM-based monophonic singing transcriber [Ryynänen and Klapuri, 2004], provided by the authors of the algorithm.

Melotranscript[10]: It is an improved, commercial version derived from the research initially carried out by [De Mulder et al., 2004], which is based on the use of an auditory model for singing transcription. For the experiment, we have used the demo version available in SampleSumo website.

Baseline algorithm: According to [Viitaniemi et al., 2003], 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. The proposed baseline method is based on such idea, and it uses Yin [De Cheveigné and Kawahara, 2002] to extract the F0 and aperiodicity at frame-level. A frame is classified as unvoiced if its aperiodicity is under < 0.4. Finally, all notes shorter than 100ms are discarded.

Results

In Figure 3.9 we show the results of our comparative analysis, from which several observations can be made.
The first observation is that none of the state-of-the-art singing transcribers has a great performance in global terms. Indeed, the highest value F-measure of *COnPOff* metric (correct onset, pitch and offset) is less than 0.5. Considering that *COnPOff* metric reflects the global goodness of singing transcribers, this result shows that singing transcription problem is still far to be solved for all real-world purposes. In any case, the best performing method is *Melotranscript*, followed by *SiPTH* and *Gómez and Bonada*, which have a similar performance. Finally, *Ryynänen* has a lower performance, probably due to the use of integer pitch values for the transcription (as suggested by [Mauch et al., 2015a]). This fact is also reflected by *OBP* metric (only bad pitch), which is especially high in the case of *Ryynänen* method.

[10]https://www.samplesumo.com/melody-transcription

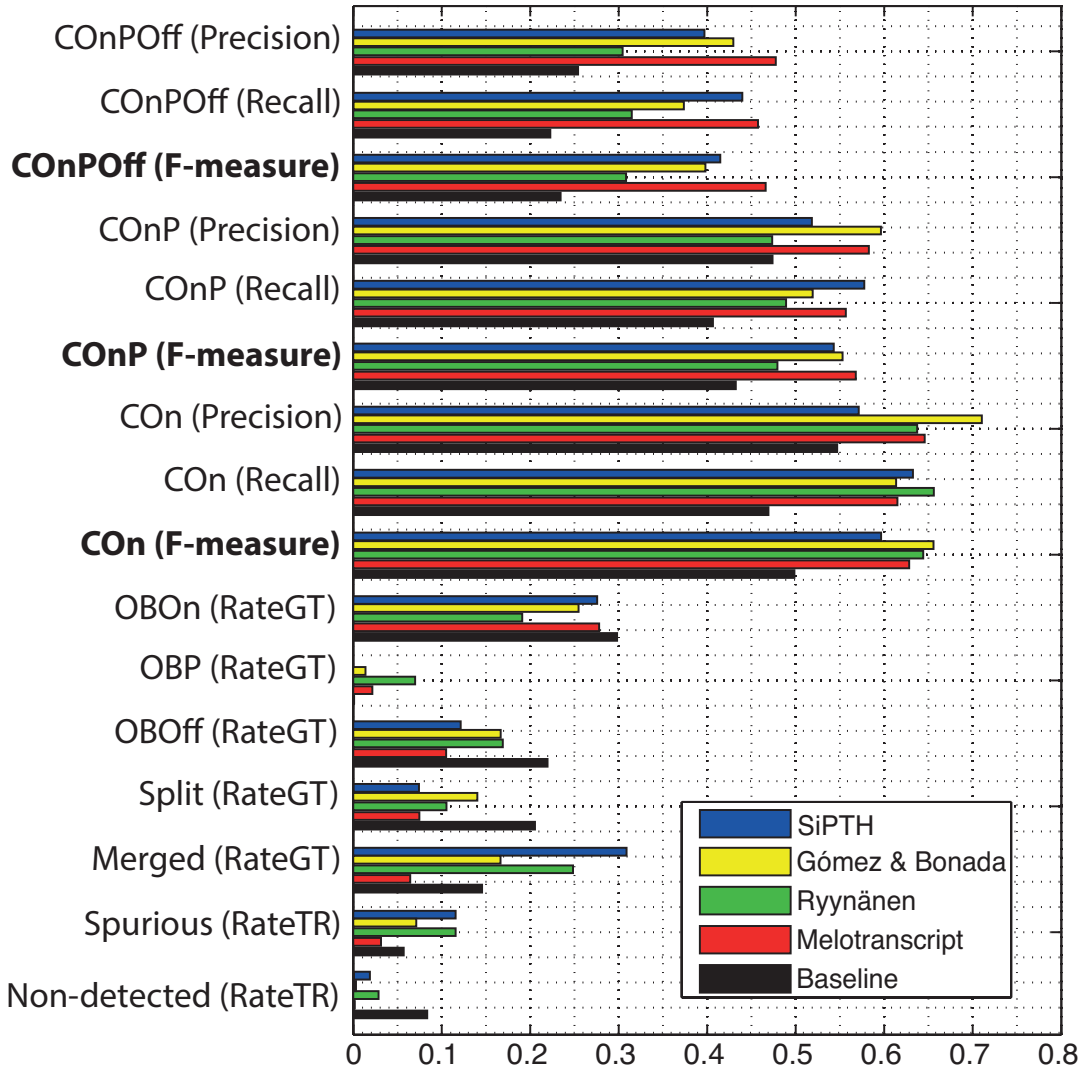

Figure 3.9: Comparison in detail of several state-of-the-art singing transcription systems using the presented evaluation framework.

In addition, the information provided by *Split* and *Merged* metrics allow us to identify the behavior of each method. Methods *SiPTH* and *Ryynänen* tend to merge notes, whereas the baseline method clearly tends to split them. On the other hand, *Gómez and Bonada* and *Melotranscript* are rather balanced in the type of errors made.
Besides, the metric *OBOn* (only bad onset), which ranges from 0.2 to 0.3 for the studied methods, let us note that an improvement in onset detection would significantly improve the global transcription accuracy. This might be due to the 50ms

tolerance for onset detection, which is quite challenging in the case of singing voice. To sum up, *Melotranscript* is the method with best performance, followed by *SiPTH* and *Gómez and Bonada*, which attain similar performance. *Ryynänen* method, however, has a lower accuracy probably due to the use of integer pitch values. All methods, however, are substantially better than the proposed *Baseline* transcriber.

3.3 Automatic Singing Assessment

In this thesis, we explore two variants of a novel approach for automatic singing assessment: frame-level similarity using f_0 curve alignment through Dynamic Time Warping (DTW), and note-level similarity using singing transcription. Both approaches provide the user with a set of intonation, rhythm and overall ratings. These ratings are obtained by measuring the similarity between the sung melody and a target performance. The approaches are evaluated by measuring the correlation between the provided ratings, and a set of ratings annotated by experts musicians. The details about this research can be found in [Molina et al., 2013]:

Molina, E., Barbancho, I., Gómez, E., Barbancho, A. M., and Tardón, L. J. (2014). Fundamental frequency alignment vs. note-based melodic siilarity for singing voice assessment. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver (Canada).

In this section, we summarize the content of this paper. Specifically, we skip the extended descriptions of the various similarity measures proposed; instead, we enumerate them and we highlight their key aspects.

3.3.1 Description of the Two Approaches

3.3.1.1 Frame-level Similarity

The f_0 contour of both the user performance and the target performance, are extracted using the Yin algorithm [De Cheveigné and Kawahara, 2002]. In these contours, unvoiced frames are indicated with $f_0 = 0$. Then, Dynamic Time Warping (DTW) [Hiroaki, 1978] is applied in order to find an optimal alignment between both f_0 contours. The cost matrix used for this alignment is

$$M_{ij} = \min\{(f_{0T}(i) - f_{0U}(j))^2, \alpha\} \tag{3.4}$$

where $f_{0T}(i)$ is the f_0 value of the target melody in the frame i, $f_{0U}(j)$ represents the f_0 value of the user's performance in the frame j, M_{ij} is the cost value and α is a constant. In our approach, we use DTW as a frame-based similarity measure, since

the total cost of the optimal path, as well as its shape, provide relevant information about the user performance. Specifically, the cost of the optimal path provides information about the pitch deviation, and its shape about the rhythmic deviation (see Figure 3.10). Consequently, we define two measures: Total Intonation Error, defined as

$$TIE = \sum_{k=1}^{K} M_{i_k j_k} \tag{3.5}$$

where: $[i_k, j_k]$ for $k \in 1 \dots K =$ optimal path

and Root Mean Squared Error, defined as

$$\varepsilon_{\text{RMS}} = \sqrt{\frac{1}{K} \sum_{k=1}^{K} {\varepsilon_k}^2} \tag{3.6}$$

where: $\varepsilon_k =$ linear regression error (of DTW optimal path, see Figure 3.10).

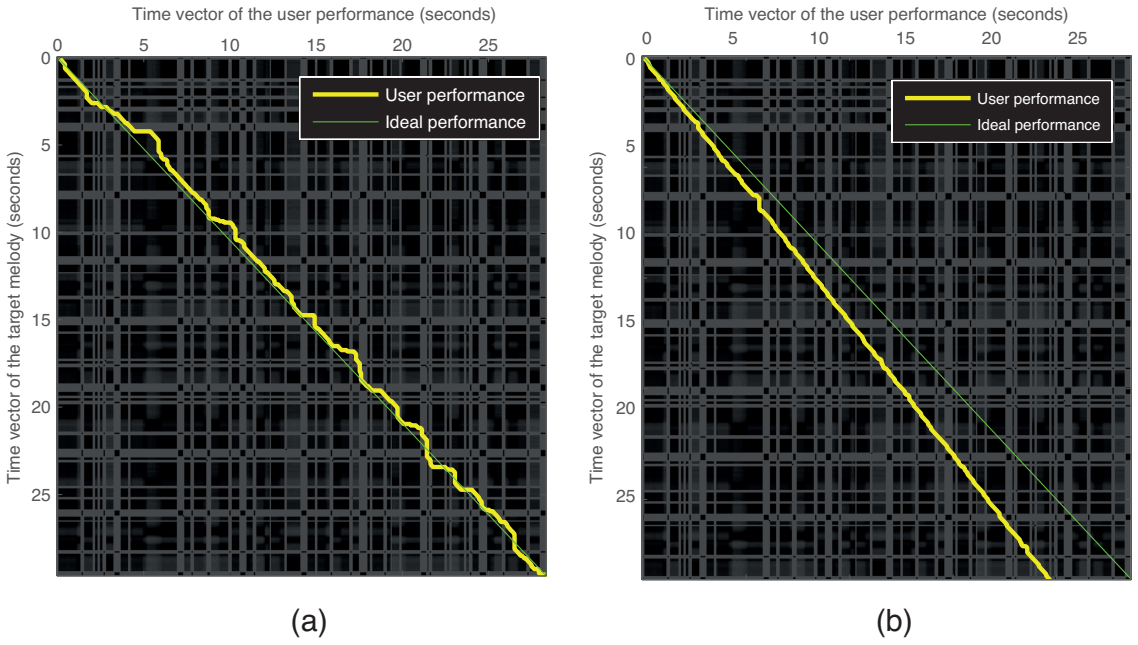

Figure 3.10: Cost matrix of the DTW, together with the path for an ideal performance (dashed line) and two different user performances. Rhythmically unstable: $\varepsilon_{\text{RMS}} = 0.36s$ (a) and rhythmically stable (different tempo): $\varepsilon_{\text{RMS}} = 0.047s$ (b).

3.3.1.2 Note-level Similarity

In this case, a note-level transcription is performed using SiPTH algorithm (see Section 3.2.1) of both user and target performances. Then, an f_0 contour alignment is performed (as in previous approach) in order to map each note from the user performance to a note from the target performance. Once this mapping is available, then several note-level similarity measures are computed: onset time deviation (ΔO), note frequency deviation (Δf) and interval deviation (ΔI).

3.3.1.3 Score Computation

The proposed system computes three different scores: intonation, rhythm and overall score. These scores are the output of three different polynomial regression functions, which use similarity measures as input features (TIE, ϵ_{RMS}, ΔO, Δf and ΔI). These regressors are trained using a dataset of 27 singing performances assessed by 4 different expert musicians. More details are provided in Section 3.3.2.

3.3.2 Evaluation

In this section, we present the ground truth built for the evaluation (Section 3.3.2.1), as well as the evaluation measures computed (Section 3.3.2.2).

3.3.2.1 Groundtruth

We combine the use of real recordings and artificially generated melodies in order to systematically control the level of intonation and rhythm deviations. The evaluation dataset is then built by introducing random pitch/rhythm variations to three different target melodies, using an harmonic plus stochastic modelling of the input signal as described in [Gómez et al., 2003a]. Three levels of random variations have been applied for both pitch and rhythm. In total, nine combinations with different degree of error are generated from each reference melody. Therefore, 27 melodies (around 22 minutes of audio) comprise the whole evaluation dataset[11].

Human judgements were collected from four trained musicians, who were asked to score from 1 to 10 the evaluation dataset in three different aspects: intonation, rhythm and overall impression. Melodies were presented in random order using headphones.

[11] Audio samples extracted from the ground truth can be found at http://www.atic.uma.es/singing

3.3.2.2 Evaluation Measures

Three different measures have been computed to evaluate the singing voice assessment system: interjudgement reliability, correlation between similarity measures and human judgements and polynomial regression error. Interjudgement reliability, proposed in [Wapnick and Ekholm, 1997], measures the correlation between human ratings. This measure aims to quantify the objectivity of the ratings. We have computed the correlation between the ratings for each pair of musicians (in total $n(n-1)/2 = 6$ pairs), and then averaged all the correlations. We have also computed the correlation coefficient for each similarity measure with respect to the different mean score given by musicians. This is a good reference about how meaningful each similarity measure is for performance assessment. A total of 27 (9 similarity measures $\times$ 3 ratings) correlation coefficients have been computed. Finally, the human criteria has been modelled in Weka through quadratic polynomial regression. The regression error quantifies the accuracy of the data fitting procedure. In this case, the evaluation dataset is the same as the training dataset. We consider the following measures from regression analysis: the correlation coefficient and the root mean squared error.

3.3.3 Results & Discussion

The mean correlation values corresponding to the interjudgement reliability measure are shown in Table 3.2. The results show that the agreement on rhythmic evaluation is lower. Nevertheless, the correlation in all cases is acceptable, and the case of intonation is specially good.

Type of score	**Mean correlation coefficient**
Intonation	0.93
Rhythm	0.82
Overall	0.90

Table 3.2: Results of interjudgement reliability

Table 3.3 shows the correlation between the different similarity measures and the human ratings. We observe a high correlation of human ratings and DTW based measures (TIE and ε_{RMS}), specially for rhythm assessment. DTW based measures do not require singing transcription, since it directly uses the low-level feature. Therefore, DTW is a simple but efficient technique for intonation and rhythm automatic assessment.

Similarity measure	Corr. with Intonation rating	Corr. with Rhythm rating	Corr. with Overall rating
TIE	0.92	0.21	0.81
ε_{RMS}	0.0012	0.81	0.52
$\overline{\Delta O}$	0.026	0.68	0.48
$\overline{\Delta O}_W$	0.037	0.68	0.48
$\overline{\Delta f}$	0.96	0.2	0.82
$\overline{\Delta f}_W$	0.89	0.23	0.82
$\overline{\Delta I}$	0.94	0.34	0.9
$\overline{\Delta I}_W$	0.87	0.35	0.87

Table 3.3: Correlation values of each similarity measure with the ratings given by trained musicians. Note: for note-level measures, we also include a weighted mean which weights each error by the note duration. TIE=Total intonation error, ε_{RMS}=Root Mean Squared Error, $\overline{\Delta O}$=Mean onset time deviation, $\overline{\Delta O}_W$=Weighted mean onset time deviation, $\overline{\Delta f}$=Mean frequency deviation, $\overline{\Delta f}_W$=Weighted mean frequency deviation, $\overline{\Delta I}$=Mean interval deviation, $\overline{\Delta I}_W$=Weighted mean interval deviation.

Type of error	Intonation	Rhythm	Overall
Correlation coefficient	0.988	0.969	0.976
Root mean squared error	0.4167	0.58	0.44

Table 3.4: Polynomial regression error

Finally, Table 3.4 shows the obtained regression errors. The optimal polynomial combination of similarity measures provides high correlation with human judgements. For intonation, the results are specially good, because the chosen similarity measures are very representative and there is a high interjudgement reliability.
As a conclusion, our experiment show that the chosen similarity measures are suitable to model the criteria of real musicians, so further research (e.g. as [Schramm et al., 2015])) is encouraged to explore the possibilities of this approach with more data in real-world use cases.

3.4 Timbre Analysis and Processing

One of the contributions of this thesis is a method to model the variations of spectral envelope along intensity in singing voice. This method is based on a parametric model of spectral envelope, whose parameters are shifted accordingly to emulate the intensity variations in singing voice. All details have been published in [Molina et al., 2014c]:

Molina, E., Barbancho, I., Barbancho, A. M., and Tardón, L. J. (2014). Parametric model of spectral envelope to synthesize realistic intensity variations in singing voice.

In *Proceedings of the 39th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, pages 634-638, Florence (Italy).

In this section, we summarize the chosen approach (Section 3.4.1) by skipping formulas or specific parameter values, we describe the evaluation methodology (Section 3.4.2), and we present the results achieved (Section 3.4.3).

3.4.1 Summary of the Approach

The proposed method consists of several steps, which are summarized in the following paragraphs:

1. Definition of a parametric model of spectral envelope: The proposed parametric model of spectral envelope is inspired by previous systems for speech / singing synthesis like [Klatt, 1980] [Bonada et al., 2001], but in our case we use 4-pole resonators instead of 2-pole resonators. These type of models synthesize the spectral envelope with several resonator filters in parallel (equivalent to the acoustic formants) with a certain overall slope (determined by the glottal source). According to our model, twelve parameters are needed to define a spectral envelope:

- Gain (Gain_{dB})
- SlopeDepth ($\text{SlopeDepth}_{\text{dB}}$)
- Frequency of the glottal formant (f_{GP})
- Bandwidth of the glottal formant (B_{GP})
- Frequency of the first formant (f_1)
- Bandwidth of the first formant (B_1)
- Frequency of the second formant (f_2)
- Bandwidth of the second formant (B_2)
- Frequency of the third formant (f_3)
- Bandwidth of the third formant (B_3)
- Frequency of the forth formant (f_4)
- Bandwidth of the forth formant (B_4)

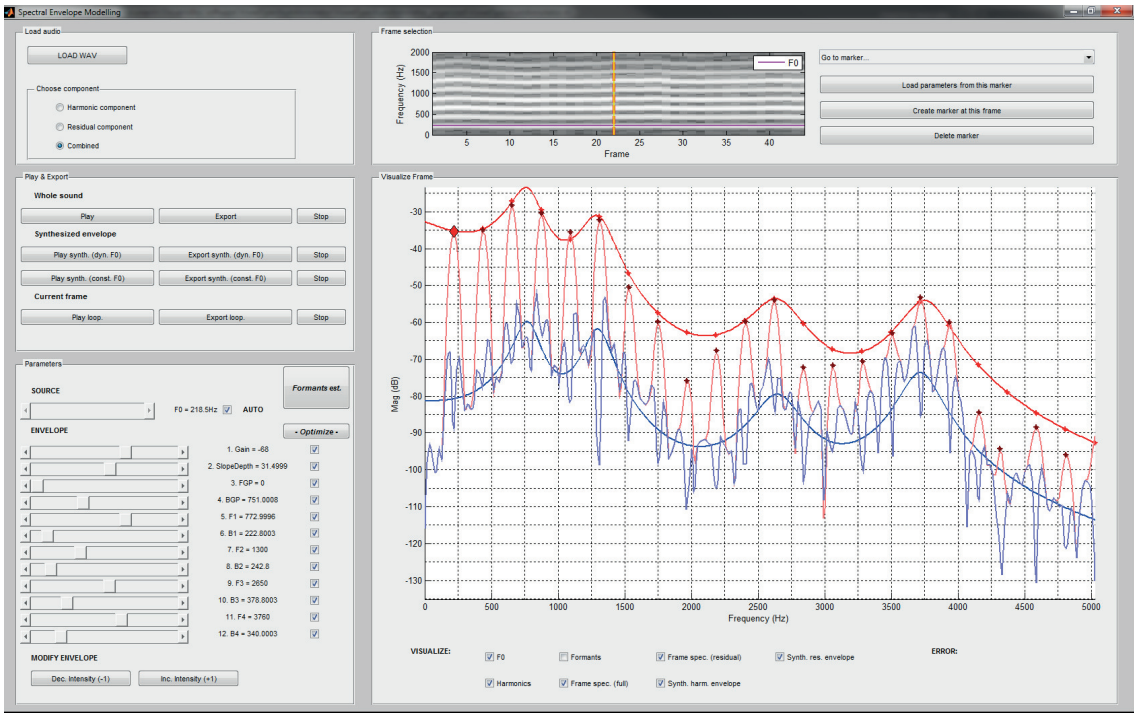

Figure 3.11: GUI for annotating spectral envelope parameters

2. Annotation of 60 sung vowels: By using a software tool specially created for this purpose (see Figure 3.11), the parameters of our model have been manually estimated for 60 sustained vowels sung by two male and two female amateur pop singers. The 60 sung notes correspond to 5 different sustained vowels (/a/, /e/, /i/, /o/ and /u/), in 3 different intended intensities (weak, normal, loud) for 4 different singers. All the notes were sung in a comfortable pitch register for all singers. In Figure 3.12 we plot the values obtained by this manual annotation.

Note that the spectral envelope is separately annotated for the harmonic and the residual components. These components are extracted using the algorithm presented in Section 2.7.4, specifically the implementation proposed in [Serra and Smith, 2014] (freely available in https://github.com/MTG/sms-tools).

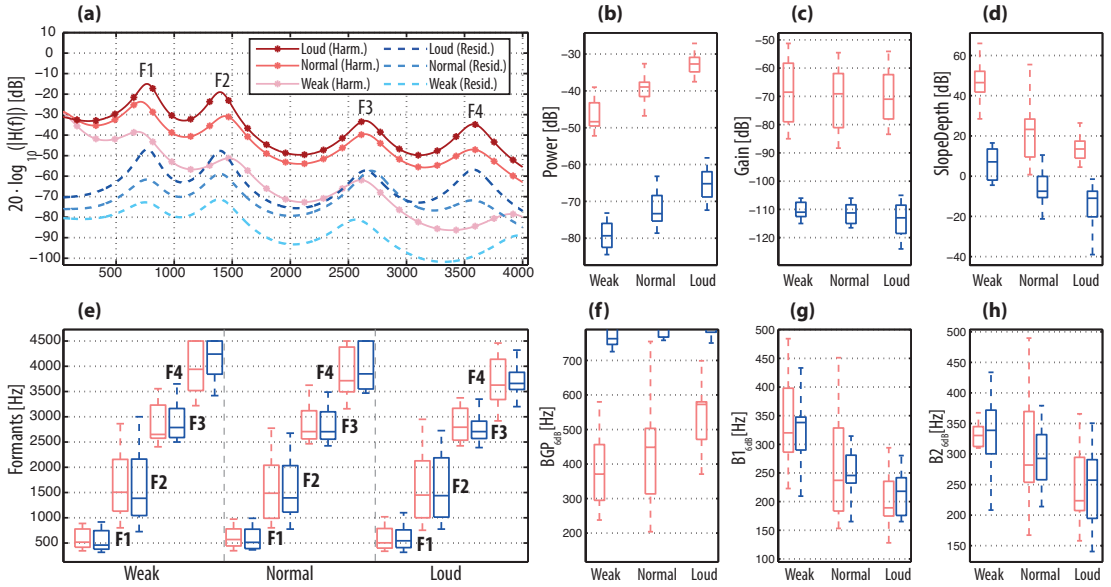

Figure 3.12: Information about the harmonic component (light red color) and the residual component (dark blue color) at different degrees of intensity **(a)** Spectral envelope of an */a/* vowel sung by a male singer **(b)** Average power **(c)** Average Gain **(d)** Average SlopeDepth **(e)** Average frequency values of the first four formants **(f)** Average bandwidths of the glottal resonator R_{GP} **(g)** Average bandwidths of the first formant R_1 **(h)** Average bandwidths of the second formant R_2

3. Modeling of parameters variation along intensity: Considering ΔI as the intensity variation introduced by the singer, each parameter has been modeled as

$$\Delta p_x = \Delta I \cdot w_x \tag{3.7}$$

where w_x is a weight obtained through linear regression on the analysis dataset described in step 2. As a consequence, in order to produce a perceived change of intensity ΔI, each parameter p_x must be assigned the value $p'_x \leftarrow p_x + \Delta p_x$. More details can be found in [Molina et al., 2014c].

3.4.2 Evaluation of the Approach

In this section, we describe the dataset (Section 3.4.2.1) and the methodology used for evaluation (Section 3.4.2.2).

3.4.2.1 Evaluation Dataset

We have collected 12 pairs of weak-loud sung vowels in mono audio with a sample rate of 11025 Hz: 4 weak-loud pairs sung by two singers (male M1 and female F1) taken from the analysis dataset, 4 sung by two singers (male M2 and female F2) not analysed before, and 4 pairs synthesized with "Bruno" (VM) and "Clara" (VF) singers in Vocaloid 3.0. Each singer (either real or synthetic) has sung a weak-loud pair using both an open vowel ($/a/$) and a closed vowel ($/i/$)) in a comfortable register.

3.4.2.2 Evaluation Methodology

In the case of natural vowels, we have compared our approach (using a intensity variation of $\Delta I = \pm 10$) against Melodyne Editor[12] (state-of-the-art commercial software). In the case of synthetic vowels, we have compared our approach with Vocaloid 3.0[13] by setting the parameter *Dynamics* to 127 (loud vowels) and 32 (weak vowels). It makes a total of 48 pairs of weak-loud or loud-weak changes[14]. The evaluation has been performed by four amateur musicians, who listened (with high-quality headphones) the different systems in random order, and they were asked to evaluate how close to a real change of intensity was the applied processing.

3.4.3 Results & Discussion

In Figure 3.13 we show the perceived closeness to a real change of intensity for each of the 48 pairs described in Section 3.4.2.2. In general, our approach achieves better results for loud-to-weak transformations, whereas in the case of weak-to-loud transformations, the results are less realistic. Indeed, we have observed that formants are less defined in weak sounds (see example in Figure 3.12.a), and therefore they are harder to analyse and manipulate. Regarding the results with synthetic vowels, our approach achieves more realism than Vocaloid at modifying the intensity for all cases.

As a conclusion, our experiments show that the manipulation of the spectral envelope significantly improve the realism of intensity changes in singing voice. Due to it, our approach provide relevant insights towards realistic intensity transformation in singing voice in real-world use cases.

[12] www.celemony.com

[13] www.vocaloid.com/en

[14] Available at: http://www.atic.uma.es/icassp2014singing

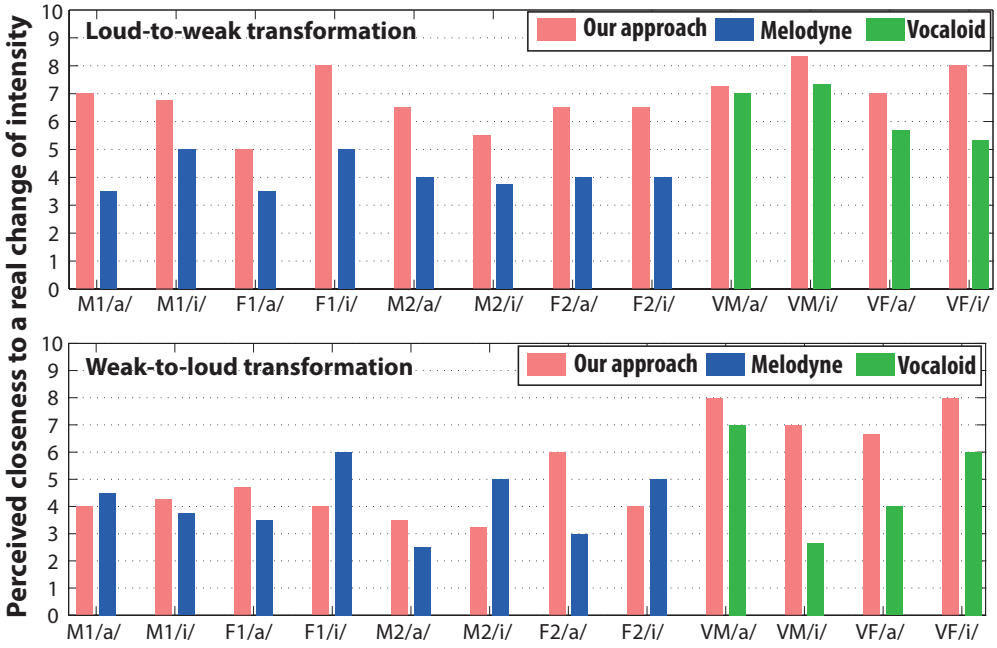

Figure 3.13: Mean perceived closeness to a real change of intensity. Each specific combination of singer/vowel (see Section 3.4.2.2) has been evaluated with various approaches, represented with different colours. The meaning of axis values is defined in Section 3.4.2.1.

3.5 Dissonance Reduction in Polyphonic Audio

In this thesis, we also propose a method for automatic reduction of dissonance in recorded isolated chords, which is described in this section.
Previous approaches address this problem using source separation and note-level processing. In our approach, we manipulate the harmonic structure as a whole in order to avoid beating partials which, according to prior research on dissonance perception, typically produce an unpleasant sound.
The proposed system firstly performs a sinusoidal plus residual modelling of the input and analyses the various fundamental frequencies existing in the chord. This information is used to create a symbolic representation of the in-tune version of the input according to some musical rules. Then, the partials of the signals are shifted in order to fit the in-tune harmonic structure of the input chord. The input is assumed to contain one isolated chord, with relatively stable fundamental frequencies belonging to the Western chromatic scale.
The evaluation has been performed by 31 expert musicians, who have quantified

the perceived consonance of six varied, out-of-tune chords in three variants: unprocessed, processed with our system and processed by a state-of-the-art commercial tool (Melodyne Editor). The proposed approach attains an important reduction of the perceived dissonance, showing better performance than Melodyne Editor for most of the cases evaluated.
The detailed description of the proposed method is in [Molina et al., 2014a]:

Molina, E., Barbancho, A. M., Tardón, L. J., and Barbancho, I. (2014). Dissonance reduction in polyphonic music using harmonic reorganization. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 22(2):325-334.

In this section, we summarize the content of this paper by focusing on the most relevant aspects of the approach. Specifically, we skip the description of related work, e.g. studies about perceptual dissonance, or review of *sinusoidal plus residual* model that is already described in Section 2.7. We also skip the description of the scales taken into account for frequency rounding in the *overtones grid generation* stage, and the formulas associated to the *beating reduction* stage. Finally, in this thesis we include an extra evaluation using vocal chords, which is not included in the published paper.

3.5.1 Description of the Approach

The proposed approach is based on a analysis-resynthesis scheme, and it can be divided into three main blocks: Analysis, Harmonic reorganization and Synthesis.

3.5.1.1 Analysis Stage

In this stage, two analysis algorithms are applied: sinusoidal plus residual modeling [Serra, 1989], and multiple-f_0 analysis.

Sinusoidal plus residual modeling

The sinusoidal plus residual modeling is performed using Serra's approach [Serra, 1989] (Section 2.7) with the following parameters: sample rate 44100Hz, window size $M = 8001$, window type Blackman-Harris 92dB, FFT size $N = 8192$ (zero-padded), hopsize $H = 2048$. Then, the sinusoids are temporally tracked by connecting spectral peaks if they are close in time (<70ms), frequency (<0.2 semitones) and amplitude (<20dB). A sequence of connected spectral peaks is called *partial*. Partials shorter than 200ms are directly removed.

Multiple-f_0 analysis

The multiple-f_0 analysis is performed using the approach proposed by [Klapuri, 2005]. This method consists of a computational model of the human auditory periphery, followed by a periodicity analysis mechanism. Estimation of multiple fundamental frequencies is achieved by cancelling each detected sound from the mixture and by repeating the estimation process with the residual. The vector of estimated f_0s in the input chord is $\widehat{\mathbf{f_0}} = [\widehat{f}_{0\,1}, \widehat{f}_{0\,2} \dots \widehat{f}_{0\,n}]$.

3.5.1.2 Harmonic Reorganization Stage

In this stage, three different processing algorithms are applied: overtones grid generation, beating reduction and harmonic reorganization.

Overtones grid generation

Given the vector of estimated f_0s corresponding to the out-of-tune input chord $\widehat{\mathbf{f_0}}$, an in-tune version of the input chord $\widehat{\mathbf{f_0^*}}$ is found by rounding each note to the closest slot within a given scale (typically major or minor). Then, the R first harmonics for each note of the in-tune version of the chords are added to the vector $\widehat{\mathbf{f^*_{H}}}_{\text{whole}}$ (called *overtones grid*):

$$\widehat{\mathbf{f^*_{H}}}_{\text{whole}} = [\widehat{f}^*_{0\,1}, \widehat{f}^*_{0\,2}, \dots, \widehat{f}^*_{0\,n}, 2\widehat{f}^*_{0\,1}, 2\widehat{f}^*_{0\,2}, \dots, 2\widehat{f}^*_{0\,n}, \dots, R\widehat{f}^*_{0\,1}, R\widehat{f}^*_{0\,2}, \dots, R\widehat{f}^*_{0\,n}] \quad (3.8)$$

Note that this procedure just handles symbolic information, and it does not apply any processing to the input signal.

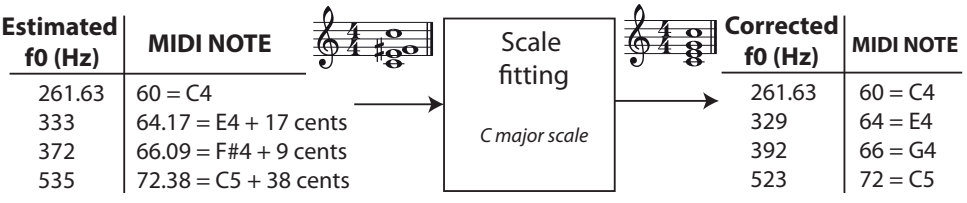

Figure 3.14: Adjustment with musical restrictions of a largely out-of-tune C major chord.

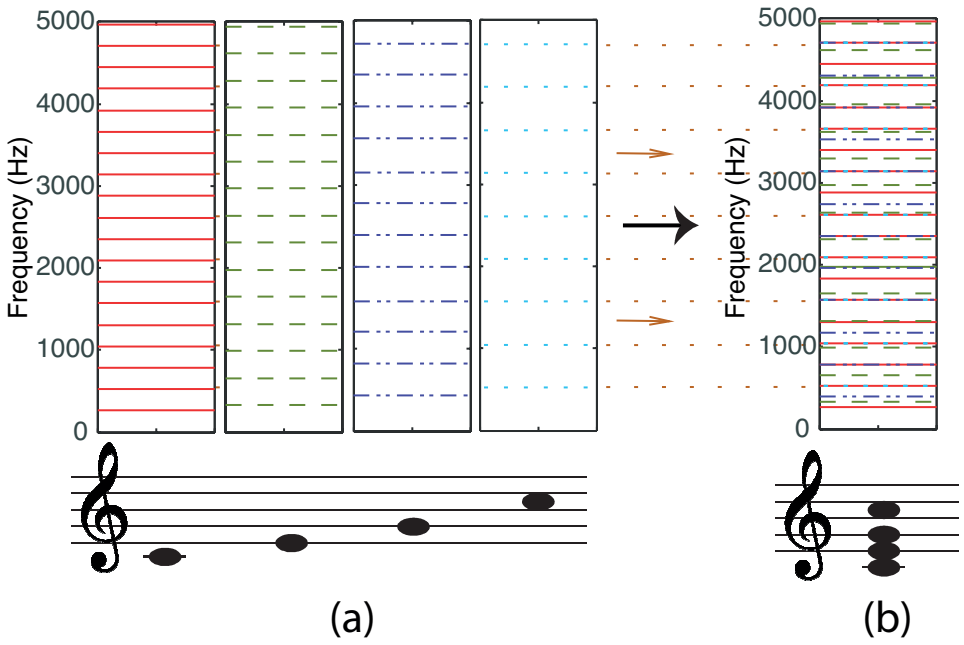

Figure 3.15: Generation of overtones grid. The overtones of every note (a) are combined into a single grid for the complete chord (b).

Beating reduction

Typically, out-of-tune chords contain undesired tremolo and vibrato in partials due to the sum of tones with similar, but not exact, frequencies. In the proposed approach, tremolo and vibrato are reduced by using envelope reconstruction and frequency stabilization (see [Molina et al., 2014a] for more details). This processing is especially noticeable for clean and pure sounds (e.g. synthetic sounds), and its contribution to the in-tuneness of the processed sound is quite limited for real-world sounds.

Harmonic reorganization

The core of the proposed approach is harmonic reorganization. In this sub-stage, the partials are shifted to the closest frequency from the overtones grid $\widehat{\mathbf{f}^{*}_{\mathbf{H}_{\text{whole}}}}$. For it, each partial is characterized with its average frequency value, and then they are pitch shifted accordingly to fit the overtones grid.
In Figure 3.16 all steps of the harmonic reorganization are shown.

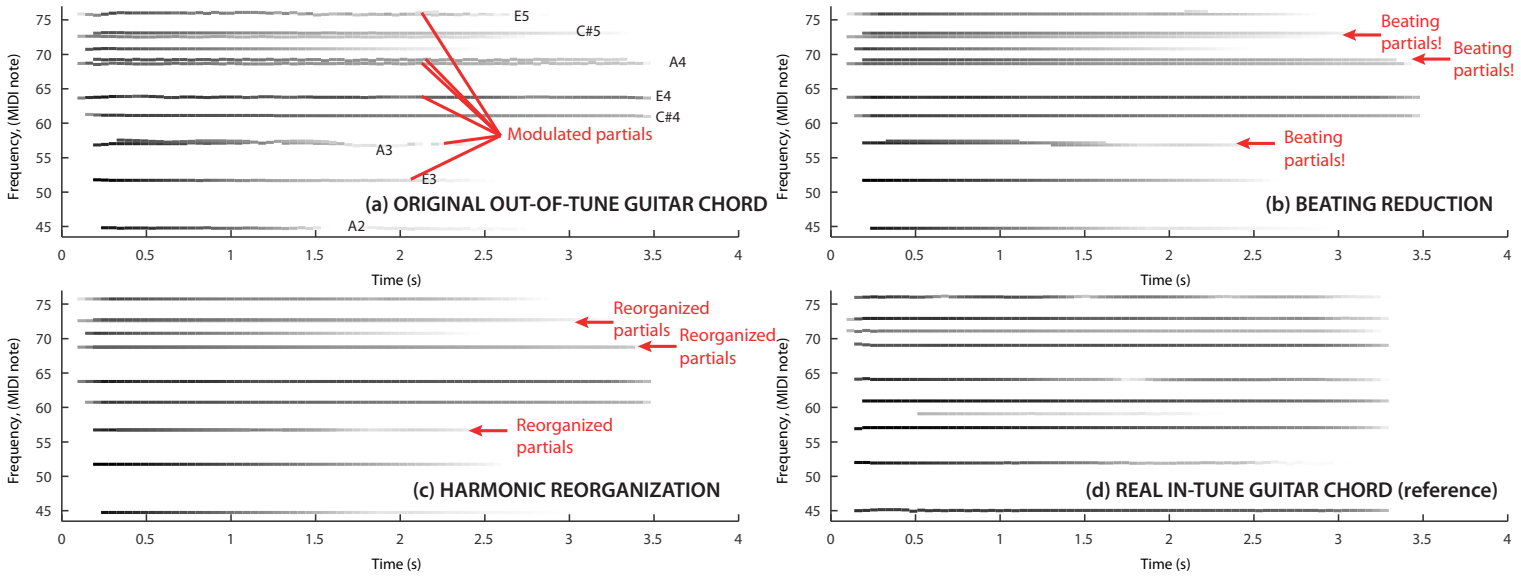

Figure 3.16: Detail of the peak frequency spectrograms of several versions of a A major chord played with acoustic guitar. **(a)** Original out-of-tune chord, whose notes are (name and MIDI number): $A2 - 33$ cents $= 44.77$, $E2 - 33$ cents $= 51.77$, $A3 + 27$ cents $= 57.27$, $C\#4 + 16$ cents $= 61.16$, $E4 - 20$ cents $= 63.80$. **(b)** Original chord after the *beating reduction* stage **(c)** Original chord after the *beating reduction* and the *harmonic reorganization* stages. **(d)** A major chord played with a real in-tune guitar.

3.5.2 Evaluation Methodology

The evaluation of the system has been performed in two parts.
First, a group of 31 experts musicians rated the perceived consonance of 18 instrumental chords through a questionnaire. This is the evaluation published in [Molina et al., 2014a].
In addition, in this thesis, we include a second evaluation using 9 chords sung by a barbershop quartet. In this case, the perceived consonance of the chords was rated by 12 experts musicians (different from the previous 31 musicians) using the same questionnaire as in the first evaluation. This extra evaluation is not included in [Molina et al., 2014a], because it has been carried out after its publication.

3.5.2.1 Dataset

The dataset contains 18 instrumental chords and 9 barbershop quartet chords. More specifically, there are 3 versions of 9 different types of out-of-tune chords (6 instrumental and 3 sung chords). The instrumental sounds are increasingly complex (from synthetic stable sounds to real chamber ensembles). The barbershop quartet have been recorded by professional singers in a recording studio. Most of the chosen sounds are major chords, because they are very common in Western music and the difference between in-tune and out-of-tune chords is quite noticeable:

- Type of out-of-tune chords

    1. C Major played with 6 harmonic complex tones with ADSR envelope. Notes: $C4$, $E4$ + 11 cents, $G4$ − 21 cents, $C5$ + 30 cents. The single notes were artificially synthesized and then combined.
    2. C minor played with 6 harmonic complex tones with ADSR envelope. Notes: $C4$, $E\flat4$ + 13 cents, $G4$ + 17 cents, $C5$ − 32 cents. As the previous case, the notes were artificially synthesized and then combined.
    3. A Major played with a real acoustic guitar. Notes: $A2$ − 33 cents, $E2$ − 33 cents, $A3$ + 27 cents, $C\#4$ + 16 cents, $E4$ − 20 cents. The guitar was deliberately left out-of-tune to sound strongly dissonant, and all the strings were played together. Then, each note was separately analysed to find out its accurate frequency value.
    4. D Major played with a real acoustic guitar. Notes: $D3$ − 30 cents, $A3$ + 28 cents, $D4$ + 15 cents, $F\#4$ + 3 cents. The recording procedure was the same as in the previous case.
    5. B♭ Major played with a real woodwind quartet: $B\flat2$, $F3$ − 44 cents, $B\flat3$ − 50 cents, $D5$ + 31 cents. The notes of the chord were extracted

from RWC database [Goto et al., 2003], carefully pitch-shifted and then combined.

6. C Major played with a real string quartet: $C3 - 6$ cents, $E3 - 7$ cents, $C4 + 30$ cents, $G4 - 73$ cents. This chord was generated in the same way as the previous one.
7. D♭ major chord sung by a real barbershop quartet: $D♭2$ - 18 cents, $A♭2 -$ 44 cents, $D♭3 + 31$ cents, $F3 + 35$ cents. This chord has been generated by applying pitch-shifting to each vocal track of a barbershop quartet multitrack recording [15].
8. E♭ major chord sung by a real barbershop quartet: $E♭2$ - 58 cents, $G2$, $B♭2 + 45$, $E♭3 + 52$ cents. This chord was generated in the same way as the previous one.
9. D♭ major chord (high register) sung by a real barbershop quartet: $D♭3$ $- 58$, $F3 + 45$, $A♭3 + 52$, $D♭4$. This chord was generated in the same way as the previous one.

- Versions
  (A) Unprocessed chord.
  (B) Processed (developed approach).
  (C) Processed (Melodyne Editor).

In version B, we have used the following parameters for all the sounds: sampling rate = 44100 Hz, window size $M = 8001$ samples, FFT size $N = 8192$ samples, number of partials per note $R = 30$ and degree of polyphony $n = 5$. In version C, the degree of polyphony and the notes of the chord have been manually adjusted for each case in order to achieve the best results. In next sections, sounds will be identified by combining the number of the chord and the type of version, i.e. *1.A* would be the first chord in the unprocessed version.
In chords 1, 2, 3, 4, 5 and 7 the chosen f_S is the tempered chromatic scale (no musical assumptions are made about the input). In 6, 8 and 9, some notes could be incorrectly rounded due to deviations higher than 50 cents, so in these cases the major scale has been chosen instead of the chromatic one.

3.5.2.2 Evaluation

Subjects

For the evaluation, 31 musicians were interviewed about the instrumental chords,

[15] *Rounders*' recording at http://www.cambridge-mt.com/ms-mtk.htm

and 12 different musicians were interviewed about the barbershop quartet chords. All of them have passed a minimum of 7 years of formal music education, and they play very different instruments (woodwind, piano, percussion...), so there is no predominant instrument. In the first group of musicians there are 16 male and 15 female individuals, and most of the subjects' age is below 25. In the second group, there are 8 male and 4 female individuals, and most of the subjects' age is below 35.

Questionnaires

The subjects were asked to rate from 1 to 10 the perceived consonance of 18 sounds. For every group of three versions (A, B and C), they were also asked to choose, globally, the best version if they had to use such chord in a musical context.

Statistics

Different measures have been taken from the questionnaires for each sound in the dataset.

- Mean perceived consonance μ_c.
- Standard deviation of the perceived consonance σ_c.
- Percentage of times that a version has been chosen as the best option among the three versions.

3.5.3 Results & Discussion

The results obtained for instrumental chords are shown in Table 3.5. In the case of synthetic sounds (chords 1 and 2), the results show a clear improvement in the consonance of the processed sounds (either with Melodyne or either with our approach). Unprocessed sounds were strongly perceived as dissonant, whereas the processed ones improved the consonance rating around 3 points. Moreover, the developed approach provides better results than Melodyne Editor for the case of synthetic sounds, since in Melodyne case noticeable beating partials are still present in the processed chords.
The case of the acoustic guitar (chords 3 and 4) is especially interesting, since it is a very common instrument and the results are quite satisfactory. More than 70% of the subjects considered the selected approach to be better than Melodyne Editor. We conclude that plucked string instruments are very appropriate to be processed with the selected approach, since the assumed partial stability holds true for most of the cases.

Chord version	***Perceived consonance [1-10]***	***Chosen as best result***
1.A Original	$\mu_c = 3.48$ $\sigma_c = 1.48$	3.2%
1.B Our approach	$\mu_{\mathbf{c}} = \mathbf{6.64}$ $\sigma_{\mathbf{c}} = \mathbf{2.05}$	**77.4%**
1.C Melodyne	$\mu_c = 5.48$ $\sigma_c = 1.80$	19.35%
2.A Original	$\mu_c = 2.67$ $\sigma_c = 1.30$	6.45%
2.B Our approach	$\mu_{\mathbf{c}} = \mathbf{5.35}$ $\sigma_{\mathbf{c}} = \mathbf{2.25}$	**74.2%**
2.C Melodyne	$\mu_c = 3.96$ $\sigma_c = 1.87$	19.3%
3.A Original	$\mu_c = 4.61$ $\sigma_c = 1.89$	3.2%
3.B Our approach	$\mu_{\mathbf{c}} = \mathbf{7.19}$ $\sigma_{\mathbf{c}} = \mathbf{1.86}$	**83.9%**
3.C Melodyne	$\mu_c = 5.83$ $\sigma_c = 2.35$	9.7%
4.A Original	$\mu_c = 4.32$ $\sigma_c = 1.81$	3.2%
4.B Our approach	$\mu_{\mathbf{c}} = \mathbf{7.09}$ $\sigma_{\mathbf{c}} = \mathbf{1.68}$	**71%**
4.C Melodyne	$\mu_c = 6.19$ $\sigma_c = 1.99$	25.8%
5.A Original	$\mu_c = 2.19$ $\sigma_c = 1.27$	0%
5.B Our approach	$\mu_{\mathbf{c}} = \mathbf{4.03}$ $\sigma_{\mathbf{c}} = \mathbf{2.33}$	**32%**
5.C Melodyne	$\mu_c = 4.64$ $\sigma_c = 2.38$	68%
6.A Original	$\mu_c = 1.54$ $\sigma_c = 0.80$	0%
6.B Our approach	$\mu_{\mathbf{c}} = \mathbf{5.54}$ $\sigma_{\mathbf{c}} = \mathbf{2.15}$	**77.4%**
6.C Melodyne	$\mu_c = 4.77$ $\sigma_c = 1.96$	22.6%

Table 3.5: Questionnaires results for instrumental chords. **x.A:** Unprocessed sound; **x.B:** Developed approach; **x.C:** Melodyne Editor.

Chord version	***Perceived consonance [1-10]***	***Chosen as best result***
7.A Original	$\mu_c = 6.09$ $\sigma_c = 1.7$	0%
7.B Our approach	$\mu_\mathbf{c} = \mathbf{8.36}$ $\sigma_\mathbf{c} = \mathbf{1.12}$	**100%**
7.C Melodyne	$\mu_c = 4.54$ $\sigma_c = 1.81$	0%
8.A Original	$\mu_c = 3.90$ $\sigma_c = 1.22$	0%
8.B Our approach	$\mu_\mathbf{c} = \mathbf{7.09}$ $\sigma_\mathbf{c} = \mathbf{1.04}$	**36%**
8.C Melodyne	$\mu_c = 6.63$ $\sigma_c = 1.02$	64 %
9.A Original	$\mu_c = 3.36$ $\sigma_c = 1.62$	0%
9.B Our approach	$\mu_\mathbf{c} = \mathbf{6.0}$ $\sigma_\mathbf{c} = \mathbf{2.04}$	**73%**
9.C Melodyne	$\mu_c = 3.63$ $\sigma_c = 2.37$	27%

Table 3.6: Questionnaires results for vocal chords. **x.A:** Unprocessed sound; **x.B:** Developed approach; **x.C:** Melodyne Editor.

In the case of a woodwind quartet (5), Melodyne performs better than our approach, with a perceived consonance of 4.63 and 4.02 respectively. If both versions are carefully compared, it can be noticed that the difference between them in terms of dissonance is mild, but Melodyne produces a more natural result. In the case of the strings quartet (6) Melodyne does not properly separate the various notes of the chord, so 6.C is still dissonant and unnatural compared to 6.B. In all comparisons, a *t-Student* test (with $p < 5\%$) revealed statistical validity [Zabell, 2008]. Regarding the results obtained with the barbershop quartet chords, they are similar to previous cases (shown in Table 3.6). In general, unprocessed sounds are perceived as strongly dissonant, whereas processed chords have a clear improvement of perceived consonance. In the case of 7 and 9, our approach performs definitely better than Melodyne, since Melodyne is not totally able to distinguish the various frequencies comprising the original chord. In the case of 8, Melodyne provides a more natural sound because it detects all frequencies in the original chord.
Therefore, our observations lead us to conclude Melodyne has a bottleneck in multiple-F0 estimation when input chords are out-of-tune chords. In that sense, our approach has the advantage of being truly robust to missing F0s in the multiple-F0 estimation.

CHAPTER 4

Conclusions and Future Research

In this chapter, we draft some conclusions about the content presented (Section 4.1), we list the contributions of this thesis (Section 4.2), and we give some suggestions for future research (Section 4.3).

4.1 Conclusions and Research Contributions

In this thesis, we have proposed a varied set of techniques and applications in the field of *singing information processing*. Specifically, the goals presented at the outset of this dissertation (Section 1.1) address the following three topics: singing transcription (both pitch and note tracking), singing skill assessment, and sound transformation (concretely: voice timbre processing, and pitch shifting in polyphonic audio). Of course, the achievement of such goals also require a deep review of the state-of-the-art on related fields. Now, in view of the data and results presented in this thesis, we can say these goals have been successfully achieved.

Review of the state-of-the-art

First, a review covering all relevant literature about the topics addressed in this thesis has been presented in Section 2. It reflects the knowledge acquired during our investigation, and it is useful to contextualize the achieved results. The topics covered by this review are: singing voice production, pitch estimation (monophonic F0 estimation, melody extraction and multi-F0 estimation), singing transcription, dynamic time warping, automatic singing assessment, timbre processing and spectral modeling synthesis.

Comparative analysis of F0-trackers for query-by-singing-humming

A comparative study of several state-of-the-art F0-trackers in the context of query-by-singing-humming has been presented (Section 3.1). Specifically, eight different F0-trackers have been tested with two state-of-the-art melody matchers for query-by-singing-humming, plus a publicly available baseline method. Three main conclusions can be drawn from this study. The first conclusion is that the three melody matchers obtain the best results with the same F0-trackers in all cases. This suggests that a simple baseline melody-matcher can be used to compare the performance of different F0-trackers in query-by-singing-humming. The second conclusion is that the recently published pYIN method for F0-tracking [Mauch, 2014] has a surprisingly great performance in noisy environments. The third conclusion is that the way F0-tracking is usually evaluated in the literature is not totally representative of its suitability for query-by-singing-humming, since it does not considers the kind of errors committed by the F0-tracker in unvoiced frames.

Singing transcription

In this thesis, a singing transcription method based on a hysteresis process on the pitch-time curve (called SiPTH, as described in Section 3.2.1) has been proposed. This method applies a hysteresis-based transformation to the Yin algorithm in order to transform its ouputs: F0, aperiodicity and energy, into a sequence of notes. The results show that this approach, which is simple to understand and to implement, achieves a performance comparable to other more complex state-of-the-art approaches for singing transcription.
In addition, a comprehensive evaluation framework for singing transcription has been presented in this thesis (Section 3.2.2). This framework includes an annotated dataset and a software tool to compute evaluation metrics and visualize the transcription. The evaluation metrics included in this framework report detailed information about the type of errors committed by the target transcriber, so they are useful to highlight its weaknesses. This framework has been used by some recent articles on singing transcription (e.g. [Mauch et al., 2015a]), and it is intended to encourage reproducible research in the area of singing transcription.

Automatic singing assessment

In Section 3.3, two different approaches for automatic singing assessment have been proposed and compared: (1) frame-level similarity against a target reference using f_0 curve alignment through Dynamic Time Warping, (2) note-level similarity using singing transcription. Both approaches require a target reference, which is considered the ideal performance. This ideal performance can be the MIDI file of

the original song, or the performance of a target user (e.g. a teacher). The system has been evaluated by analyzing the correlation between the scores provided by it, and the scores provided by a set of experts musicians. The results of our comparison show that frame-level similarity is a simple but effective technique for intonation and rhythm assessment, and that using singing transcription introduces more complexity to the system without a clear advantage.

Timbre analysis and processing

A method to model the variations of spectral envelope along intensity in singing voice has been proposed in Section 3.4. This method is based on a parametric spectral-envelope model, whose parameters are shfited accordingly to emulate the intensity variations in singing voice. Three contributions are related to this investigation: (1) a parametric model of spectral envelope based on 4-pole filter for formants modeling, (2) a software tool to annotate sung vowels using such parametric model, and (3) a method to shift the parameters of such model in order to produce realistic intensity variations. We observed that two parameters are mainly responsible for the perception of vocal intensity, since they decrease when vocal intensity increases: spectral tilt and formants bandwidth. The proposed system has been compared against Melodyne Editor and Vocaloid 3.0, through a listening questionnaire answered by four amateur musicians. The results show that the suggested approach significantly increases the realism of the transformations in comparison with the two other approaches, specially for the case of loud-to-weak transformations.

Dissonance reduction in polyphonic audio

Finally, a method for automatic reduction of dissonance in recorded isolated chords has been proposed in Section 3.5. This method performs a multiple-F0 estimation to identify the chord to be tuned, and a sinusoidal plus residual modeling to shift its partials. These partials are shifted to fit the harmonic structure of the in-tune version of the same chord. The evaluation methodology has been based on listening tests where a set of expert musicians have assessed the perceived consonance of several recorded chords before and after the processing. Our results show that the proposed system performs generally better than Melodyne Editor to improve the consonance of out-of-tune chords, both in instrumental and vocal chords.

4.2 Summary of Contributions

In this Section, we enumerate the scientific contributions of this thesis, together with the research resources published during our investigation.

Scientific contributions

- **Review of previous research:** A comprehensive review of current state-of-the-art methods and techniques related to *Singing Information Retrieval* field is provided in Chapter 2. This review covers the following topics: singing voice production (Section 2.1), pitch estimation (Section 2.2), singing transcription (Section 2.3), dynamic time warping (Section 2.4), automatic singing assessment (Section 2.5), timbre processing (Section 2.6) and spectral modeling synthesis (Section 2.7).

- **Comparative analysis of monophonic F0 trackers:** A comparative analysis of several state-of-the-art F0 trackers in the context of query-by-singing-humming has been carried out. It has been published in [Molina et al., 2014d], and summarized in Section 3.1.

- **Novel method for singing transcription:** A method for singing transcription (named as *SiPTH*) using interval-based segmentation with a hysteresis cycle on the pitch-time curve has been proposed. This method is simple to implement, and its performance is similar to other state-of-the-art methods. It has been published in [Molina et al., 2015], and summarized in Section 3.2.1.

- **Evaluation framework for singing transcription:** An analysis of previous evaluation strategies in singing transcription (used datasets and evaluation metrics), and an evaluation framework (annotated dataset, implemented metrics and GUI) has been presented. It has been published in [Molina et al., 2014b] and summarized in Section 3.2.2.

- **Method for singing assessment:** A novel approach for automatic singing assessment based on pitch contour alignment using dynamic time warping has been proposed. It has been published in [Molina et al., 2013] and summarized in Section 3.3.

- **Method for timbre processing:** A parametric model of spectral envelope based on 4-pole filters for formants modeling, together with an study about the variations of spectral envelope along singing intensity, and a method to perform realistic intensity variations in singing voice have been presented. It has been published in [Molina et al., 2014c] and summarized in Section 3.4.

- **Method for dissonance reduction in polyphonic audio:** A method for dissonance reduction of out-of-tune chords using harmonic reorganization has been proposed. It has been published in [Molina et al., 2015] and summarized in Section 3.5.

Research resources

- **Baseline algorithm for audio-to-MIDI melody matching:** We provide a Matlab implementation of a DTW-based baseline algorithm for audio-to-MIDI melody matching. It can be used as a starting point to work in query-by-singing-humming, or to measure the suitability of a given F0 tracker for query-by-singing-humming (as stated in Section 3.1). It can be found in the following link:

  `www.atic.uma.es/ismir2014qbsh`

- **Spectral envelope annotation tool:** Matlab tool (with GUI) to annotate parameters of spectral envelope in sustained vowels. More details about this tool can be found in Section 3.4, and it can be downloaded at the following link:

  `www.atic.uma.es/icassp2014singing`

- **Singing transcription evaluation tool:** Matlab tool (with GUI) to visualize and evaluate monophonic melody transcriptions and annotated dataset for singing transcription. This dataset consists of 38 melodies (1154 seconds) sung untrained singers (men, women and children), annotated by expert musicians at note-level. More details can be found in Section 3.2.2, and it can be downloaded in:

  `www.atic.uma.es/ismir2014singing`

- **Database of Piano Chords**: In addition to all provided material related to singing voice, a database of piano chords for multiple F0 estimation has been also published [Barbancho et al., 2013].

4.3 Suggestions for Future Research

Apart from the published material, many other relevant observations and ideas have appeared during our investigation. Some of these considerations are worth to be mentioned because they may be solutions for specific weaknesses of the proposed approaches, or may even be a promising alternative to deal with the addressed problems. In this section, we discuss these ideas and propose specific suggestions for future research.

Singing transcription

- **Better evaluation framework:** The usefulness of the evaluation framework described in Section 3.2.2 may be improved by adding more annotated data. The manual annotation might be efficiently performed using the recent Tony software tool [Mauch et al., 2015a]. Eventually, if this dataset become large enough, it might be used to define a *singing transcription task* in MIREX[1]. Additionally, the evaluation metrics used for it may be integrated in `mir_eval` [Raffel et al., 2014] Python package in order to make them available in a standarized format. Finally, this evaluation framework could include not only context-independent metrics (e.g. note accuracy), but also context-specific metrics (e.g. answering *how well does your transcriber work for query-by-singing-humming?*)

- **Singing transcription based on Hidden Markov Models (HMM) using timbre features:** According to our observations, an HMM-based method for singing transcription including timbre features (e.g. MFCCs) could be an interesting path forward. This idea is based on three main facts: (1) many successful system for speech recognition are based on HMMs using MFCCs as main feature (see Section 2.6.4.1), (2) speech recognition and singing transcription seem to share a similar nature (especially when lyrics are present), and (3) some successful approaches for singing transcription are already based on HMMs, but, to the best of our knowledge, none of them use timbre features (see Section 2.3).

Singing skill assessment

- **Robust pitch contour alignment:** Pitch contour alignment is the basis of our approach for singing assessment (see Section 3.3), but it can be challenging when the singer makes considerable intonation or rhythm errors. In order to deal with it, we propose to perform audio-to-audio alignment using not only pitch information, but also other features such as energy, aperiodicity, and even MFCCs. In this case, audio-to-MIDI alignment does not longer apply, so we propose the use of several reference audio recordings for each song (for higher robustness), corresponding to accurate, real singing performances.

- **Song-independent approach for automatic singing assessment:** The use of reference melodies has a clear disadvantage: a lot of material must be prepared in an eventual singing game to create a large set of singing exercises.

[1] www.music-ir.org/mirex

However, as stated by [Nakano et al., 2009], the accuracy of singing performances can be often assessed by a human listener even if the melody being sung is unknown. Due to this fact, in some contexts, a song-independent approach might be more suitable to achieve a full working system for singing assessment, so we recommend to explore this way in further research.

Timbre processing

- **Alternative approach for realistic intensity variation using LPC poles warping:** The proposed method for realistic intensity variation in singing voice (see Section 3.4) is based on a parametric model of spectral envelope. We observed that two parameters are mainly varied along intensity: spectral slope and formants bandwidth. In view of this result, we suggest to explore LPC poles warping to process singing voice, since it is computationally lighter and might lead to other relevant real-time applications.

- **Removal of partial tracking stage in polyphonic audio processing:** Our approach for dissonance reduction in polyphonic audio (see Section 3.5) uses sinusoidal plus residual modeling, and tracks each sinusoid along time in order to identify the partials of the sound, as proposed by [Serra, 1989]. However, partial tracking is computationally costly, so we suggest to experiment without any kind of partial tracking to achieve a lighter, and more compact scheme for dissonance audio reduction in polyphonic audio.

APPENDIX A

Relevant online research resources

In this appendix, we include a set of links with relevant research resources that have been referenced along this thesis.

A.1 Software

Sonic visualizer and sonic annotator

Sonic visualizer is a software tool with a GUI to visualize waveforms, spectrograms, descriptors, etc. It supports VAMP plugins for audio analysis, which include a long list of state-of-the-art descriptors. It has been used in many processes during our investigation). On the other hand, *sonic annotator* allows to use VAMP pluging in command line without a GUI:

- `http://www.sonicvisualiser.org/`
- `http://www.vamp-plugins.org/sonic-annotator/`

Essentia

Essentia is an open-source C++ library for audio analysis and audio-based music information retrieval [Bogdanov et al., 2013]. It contains an extensive collection of reusable algorithms which implement audio input/output functionality, standard digital signal processing blocks, statistical characterization of data, and a large set of spectral, temporal, tonal and high-level music descriptors. The library is also wrapped in Python and includes a number of predefined executable extractors for the available music descriptors.

- `http://essentia.upf.edu/`

Librosa

LibROSA is a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems [McFee et al., 2015].

- `https://github.com/librosa/librosa`

Madmom

Madmom is an audio signal processing library written in Python with a strong focus on music information retrieval (MIR) tasks.

- `https://github.com/CPJKU/madmom`

mir_eval

Python library for computing common heuristic accuracy scores for various music/audio information retrieval/signal processing tasks [Raffel et al., 2014].

- `https://github.com/craffel/mir_eval`

MIRtoolbox

MIRtoolbox offers an integrated set of functions written in Matlab, dedicated to the extraction from audio files of musical features such as tonality, rhythm, structures, etc. The objective is to offer an overview of computational approaches in the area of Music Information Retrieval.

- `https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox`

VOICEBOX: Speech Processing Toolbox for MATLAB

VOICEBOX is a speech processing toolbox consists of MATLAB routines.

- `http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html`

Yin algorithm implementations

Some implementations of Yin algorithm for monophonic pitch tracking [De Cheveigné and Kawahara, 2002], which is described in Section 2.2.1, can be found in the following links:

- `http://audition.ens.fr/adc/` (in Matlab, implemented by the author)
- `https://github.com/JorenSix/TarsosDSP` (in Java)
- `https://code.soundsoftware.ac.uk/projects/pyin` (in C++)
- `https://github.com/ashokfernandez/Yin-Pitch-Tracking` (pure C)

Praat software for voice analysis

Praat is a classical software tool that implements several voice analysis algorithms. It includes the classical (and well performing, as showed in Section 3.1) method for pitch tracking by [Boersma, 1993], as well as several methods for formant tracking.

- `http://www.fon.hum.uva.nl/praat/`

Melody extraction: MELODIA

MELODIA is an algorithm for melody extraction developed by [Salamon, 2013], and it is available at:

- `http://mtg.upf.edu/technologies/melodia`

Melotranscript

SampleSumo's Melody Transcription (MeloTranscript) library, is a technology package for offline monophonic melody transcription. It is the latest evolution of the method proposed by [De Mulder et al., 2004].

- `https://www.samplesumo.com/melody-transcription`

Songs2See

Songs2See is a representative example of the state-of-the-art in automatic singing skill assessment, and it works as an online game [Dittmar et al., 2010].

- `http://www.songs2see.com/en/`

LabROSA: Matlab Audio Processing Examples

Managed by Dan Ellis, it contains several little pieces of Matlab code related to MIR that might be fun or useful to play with.

- `https://www.ee.columbia.edu/~dpwe/resources/matlab/`

sms-tools

Sound analysis/synthesis tools for music applications written in python (with a bit of C) plus complementary lecture materials. It implements the spectral models described in Section 2.7.

- `https://github.com/MTG/sms-tools`

Baseline method for QBSH

It is described in Section 3.1.1.2, implemented in Matlab and based on DTW. It is useful to getting started in QBSH and to evaluate new F0 trackers in the context of QBSH.

- `http://www.atic.uma.es/ismir2014qbsh/`

Evaluation framework for singing transcription

Presented in Section 3.2.2, it is a Matlab tool (with GUI) to visualize and evaluate monophonic melody transcriptions. It implements a set of relevant metrics to analyze the behavior of the target transcriber.

- `http://www.atic.uma.es/ismir2014singing/`

Tool to annotate spectral envelope of singing

Matlab tool (with GUI) to annotate parameters of spectral envelope in sustained vowels. More details about the tool can be found in Section 3.4.

- `http://www.atic.uma.es/icassp2014singing/`

A.2 Datasets

MIR corpora by Roger Jang

Datasets for query-by-singing-humming, singing voice separation, query-by-tapping, etc.

- `http://mirlab.org/dataSet/public/`

Singing transcription dataset

Dataset for singing transcription used for evaluation in Section 3.2.2. It consists of 38 melodies sung by adult and child untrained singers, recorded in mono with a sample rate of 44100Hz and a resolution of 16 bits. The duration of the excerpts ranges from 15 to 86 seconds, and the total duration of the whole dataset is 1154 seconds.

- `http://www.atic.uma.es/ismir2014singing/`

APPENDIX B

Publications

In this appendix, the publications associated to this PhD thesis are included, which are:

- Molina, E., Barbancho, I., Gómez, E., Barbancho, A. M., and Tardón, L. J. (2014). Fundamental frequency alignment vs. note-based melodic similarity for singing voice assessment. In *Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, pages 744-748, Vancouver (Canada).

- Molina, E., Barbancho, A. M., Tardón, L. J., and Barbancho, I. (2014). Dissonance reduction in polyphonic music using harmonic reorganization. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 22(2):325-334.

- Molina, E., Barbancho, A. M., Tardón, L. J., and Barbancho, I. (2014). Evaluation framework for automatic singing transcription. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 567-572, Taipei (Taiwan).

- Molina, E., Barbancho, I., Barbancho, A. M., and Tardón, L. J. (2014). Parametric model of spectral envelope to synthesize realistic intensity variations in singing voice. In *Proceedings of the 39th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, pages 634-638, Florence (Italy).

- Molina, E., Tardón, L. J., Barbancho, I., and Barbancho, A. M. (2014). The importance of F0 tracking in query-by-singing-humming. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 277-282, Taipei (Taiwan).

- Molina, E., Tardón, L. J., Barbancho, A. M., and Barbancho, I. (2015). SiPTH: Singing transcription based on hysteresis defined on the pitch-time curve. *IEEE/ACM Transactions on Acoustics, Speech and Language Processing*, 23(2):252-263.

B.1

Molina, E., Barbancho, I., Gómez, E., Barbancho, A. M., and Tardón, L. J. (2014). Fundamental frequency alignment vs. note-based melodic similarity for singing voice assessment. In *Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, pages 744-748, Vancouver (Canada).

FUNDAMENTAL FREQUENCY ALIGNMENT VS. NOTE-BASED MELODIC SIMILARITY FOR SINGING VOICE ASSESSMENT

Emilio Molina[1], *Isabel Barbancho*[1], *Emilia Gómez*[2], *Ana María Barbancho*[1], *Lorenzo J. Tardón*[1]

[1]Dept. Ingeniería de Comunicaciones, ETSI Telecomunicación, Universidad de Málaga, Spain
[2]Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
emm@ic.uma.es, ibp@ic.uma.es, emilia.gomez@upf.edu, abp@ic.uma.es, lorenzo@ic.uma.es

ABSTRACT

This paper presents a generic approach for automatic singing assessment for basic singing levels. The system provides the user with a set of intonation, rhythm and overall ratings obtained by measuring the similarity of the sung melody and a target performance. Two different similarity approaches are discussed: f_0 curve alignment through Dynamic Time Warping (DTW), and singing transcription plus note-level similarity. From these two approaches, we extract different intonation and rhythm similarity measures which are combined through quadratic polynomial regression analysis in order to fit the judgement of 4 trained musicians on 27 performances. The results show that the proposed system is suitable for automatic singing voice rating and that DTW based measures are specially simple and effective for intonation and rhythm assessment.

Index Terms— singing assessment, automatic transcription, score alignment, melodic similarity, singing voice

1. INTRODUCTION

The assessment of a given musical performance is commonly affected by many subjective factors, even in the case of expert musicians [1]. Therefore, the development of an automatic performance evaluation system is a challenging problem. Under controlled conditions, some objective aspects can be considered and computationally modelled. Some studies have analyzed the reliability of judgements in music performance evaluation [1, 2, 3]. In such studies, different musicians were asked to rate a certain number of performers according to different aspects, with the aim of studying how objective the different judgements were. Some aspects such as intonation accuracy, vibrato or rhythm seem to be quite reliably judged by musicians, unlike more subjective aspects such as diction.

This work has been funded by the Ministerio de Economia y Competitividad of the Spanish Government under Project No. TIN2010-21089-C03-02 and Project No. IPT-2011-0885-430000 and by the Ministerio de Industria, Turismo y Comercio under Project No. TSI-090100-2011-25. This work has been partially done under Campus de Excelencia Internacional CEI Andalucía TECH in the context of Program Campus de Excelencia Internacional of the Spanish Ministerio de Educación.

Prior work has lead to various solutions for automatic singing rating [4, 5, 6, 7, 8, 9, 10]. In general, all these systems focus on intonation assessment with visually attractive real-time feedback. Songs2See [10] is a recent and representative example of the state of the art. Nevertheless, current approaches do not generally handle rhythmic misalignments, and the feedback provided is not directly based on trained musicians' judgements. This study deals with automatic intonation and rhythm assessment of singing performances, being our main goal to provide the user with meaningful feedback based on modeling teachers' criteria. We focus on basic singing levels, i.e. children and beginners. Two different approaches for singing assessment are evaluated: dynamic time warping (DTW) and note-level similarity with respect to a target melody.

This paper is organized as follows: Section 2 provides an overall description of the selected approach. The evaluation methodology is presented in Section 3, including ground truth gathering (Section 3.1) and evaluation measures (Section 3.2). Section 4 presents our main results and Section 5 draws some conclusions about this study.

2. SELECTED APPROACH

We propose a generic schema for singing assessment based on melodic similarity with respect to a target melody. The overall block diagram is illustrated in Figure 1. The audio input is first analyzed to extract a set of low-level descriptors (Section 2.1). They are then used to measure melodic similarity with respect to a target melody, whose definition is discussed in Section 2.2. Two different similarity measures are computed simultaneously: fundamental frequency (f_0) alignment (Section 2.3), and automatic singing transcription (Section 2.4) combined with note-level similarity (Section 2.5). The final step of the singing assessment system is the Performance rating stage (Section 2.6), which assigns an overall rating to the user performance.

2.1. Low-level feature extraction

We use the well-known Yin algorithm [11] to compute two related features: f_0 and aperiodicity (or voicing). These descriptors, combined with the instantaneous power of the audio

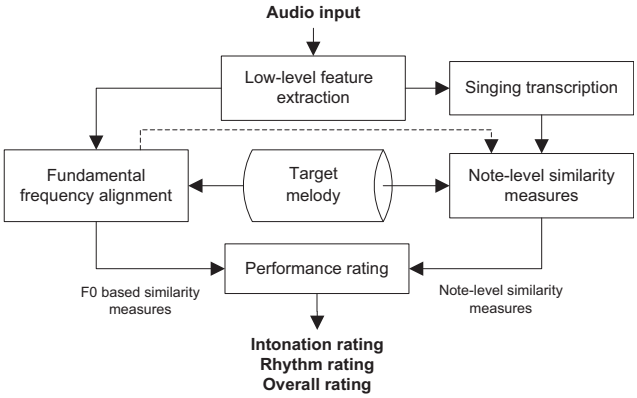

Fig. 1. Overall block diagram

signal, are given to other system blocks for singing assessment.

2.2. Target melody

The target melody is the performance that should be imitated by the student to achieve a good score. In our approach, the target melody is sung by is a *target singer*, i.e. a trained singer who is asked to sing with a rather pure voice, without vibrato, trying to be a good reference for beginners and children. Some post-processing is then applied to correct minor pitch and rhythm mistakes. Although we initially considered the symbolic score as a target melody, the fact of having a target singing voice allows a better alignment between f0 sequences and the measurement of detailed expressive resources.

2.3. Fundamental frequency alignment

Dynamic Time Warping (DTW) [12, 13, 14] is employed in order to find an optimal match between two given sequences under certain restrictions. However, it must be noted that the definition of optimal match strongly affects the robustness of the alignment. We have substituted the f_0 value of unvoiced regions by a constant value $f_{\text{unvoiced}} = 0$ Hz (see more details on voiced/unvoiced frame classification in Section 2.4). By removing the unvoiced sections, spurious f_0 values are avoided and only actual sung regions are compared. Therefore, the cost matrix M of the DTW can be defined as follow:

$$M_{ij} = min\{(f_{0T}(i) - f_{0U}(j))^2, \alpha\} \quad (1)$$

where $f_{0T}(i)$ is the f_0 value of the target melody in the frame i, $f_{0U}(j)$ represents the f_0 value of the user's performance in the frame j, M_{ij} is the cost value and α is a constant. When the squared f_0 difference becomes larger than α, it is assumed that an spurious case has been found and its contribution to the cost matrix is limited.

The DTW algorithm takes as input the cost matrix, and it provides an optimal path $[i_k, j_k]$ for $k \in 1 \dots K$, where K is the length of the path. We limit the slope of the path to the range $[10°, 80°]$ (deviations between transcription and reference are considered to be moderate).

2.3.1. DTW as an intonation similarity measure

The cost matrix provides information about the instantaneous deviation of the sung note with respect to the reference, as well as information about the total f_0 deviation of the sung melody. We consider the total cost of the optimal path to be a similarity measure for intonation assessment. The total intonation error (TIE) is computed as follow:

$$TIE = \sum_{k=1}^{K} M_{i_k j_k} \quad (2)$$

where M is the cost matrix, $[i_k, j_k]$ for $k \in 1 \dots K$ is the optimal path, and K is the length of the path.

2.3.2. DTW as a rhythmic similarity measure

In this paper, we propose DTW as a powerful procedure for automatic rhythm assessment. The idea is to analyze the shape of the optimal path, since it is a rich source of information about the rhythmic performance. In the cost matrix of the DTW, a $45°$ straight line represents a perfect rhythmic performance (no deviation with respect to the target melody). A poor rhythmic performance would yield deviations with respect to such straight line. The precise deviation location can be extracted from this curve, as well as the total amount of rhythmic error. On the other hand, a straight line with an angle $\alpha \neq 45°$ represents a good rhythmic performance in a different tempo. The straightness can be quantified through a linear regression analysis: Let $\mathbf{y} = \beta_0 + \beta_1 \mathbf{x} + \boldsymbol{\varepsilon}$ be the linear model that best fits the optimal path within the cost matrix, with $\boldsymbol{\varepsilon}$ the error of fit. The error measure proposed is the root mean square (RMS): $\varepsilon_{\text{RMS}} = \sqrt{\frac{1}{N}\sum_{k=1}^{K} {\varepsilon_k}^2}$ in seconds. In Figure 2, two different situations are illustrated: a bad rhythmic performance that leads to a high linear regression error ($\varepsilon_{\text{RMS}} = 0.36s$, solid line), and the result of a perfect rhythmic performance played in a different tempo (dotted line). In the latter case, the linear regression error is very low ($\varepsilon_{\text{RMS}} = 0.047s$). Note that ε_{RMS} is a tempo-independent measure.

2.4. Singing transcription

We consider a f_0-based note segmentation approach with a hysteresis cycle for singing transcription [15, 16], and performed in the following steps: (1) locate the segments where the user is singing, (2) split the voiced segments into different notes and (3) label each note in terms of pitch.

We classify voiced and unvoiced frames by detecting stable frequency regions. If the f_0 curve is stable during a certain time (100ms in the implemented system), we create a new voiced segment. When there is a gap in the f_0 curve, such segment ends. Gaps of one exact octave are not considered,

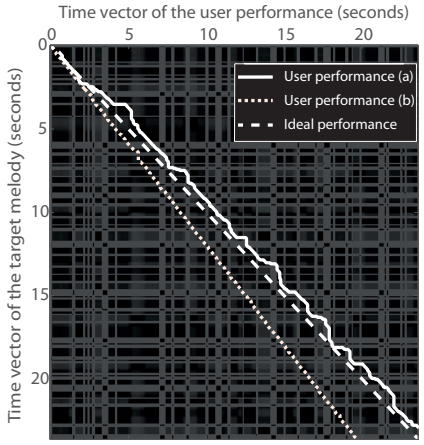

Fig. 2. Cost matrix of the DTW, together with the path for an ideal performance (dashed line) and two different user performances. Rhythmically unstable: $\varepsilon_{\text{RMS}} = 0.36s$ (solid line) and rhythmically stable (different tempo): $\varepsilon_{\text{RMS}} = 0.047s$ (dotted line).

since they are usually due to octave jumps during the same note. This process is carried out for the whole signal. In addition, voiced segments with a mean power below a threshold t_{pwr}, or mean aperiodicity above a threshold t_{ap} are directly tagged as unvoiced. This later classification avoids harmonic noises to be estimated as false voiced regions.

Once voiced segments are located, we segment voiced portions through f_0-based note segmentation. We use an hysteresis cycle in time and frequency in order to ignore minor deviations with respect to a pitch center. We dynamically estimate such pitch center by averaging f_0 values within a note. The estimated pitch average then becomes more precise as the note length increases. When the instantaneous f_0 of a note greatly deviates respect to its pitch average, a note split happens and the process starts again.

Once the sung notes are estimated, we assign a single pitch to each note. According to [17], the best pitch estimation for a note is a weighted mean of the most representative range of f_0 values. This type of mean is called alpha-trimmed mean [18], and it removes the extreme f_0 values (usually corresponding to the boundaries) before computing the mean. We have chosen this approach in this paper.

2.5. Note-level similarity measures

Note-level similarity measures are used to compare the symbolic representations of the sung melody and the target one. f_0 alignment, combined with melodic transcription, provides a note-to-note comparison even when rhythmic misalignment is present. The considered measures consist on the *average* ($\bar{x}$) and the *rhythmically weighted average* ($\bar{x}_W$) of three different magnitudes: onset time deviation (ΔO), note frequency deviation (Δf) and interval deviation (ΔI). While the *average* does not take into account note durations, the *rhythmically weighted average* does:

$$\overline{\Delta x} = \frac{\sum_{i=1}^{n} |\Delta x_i|}{n} \qquad \overline{\Delta x}_W = \frac{\sum_{i=1}^{n} l_i \cdot |\Delta x_i|}{\sum_{i=1}^{n} l_i} \quad (3)$$

where Δx_i is the deviation between the user transcription and target melody of the magnitude x (onset time, note frequency, or interval) for the note i, $\overline{\Delta x}$ is the average deviation of the generic magnitude x, $\overline{\Delta x}_W$ is the rhythmically weighted average deviation, l_i is the length of the note i, and n is the total number of notes. We now present the considered magnitudes.

2.5.1. *Onset time*

Let O_i be the onset time of the note i of the target melody, and $\hat{O}_i$ the onset time of the related note of the user performance. Then, the onset deviation is defined as $\Delta O_i = O_i - \hat{O}_i$.

2.5.2. *Note frequency*

We define f_i as the frequency of the note i of the target melody, and $\hat{f}_i$ as the frequency of the same note of the user performance. The note frequency deviation is then defined as $\Delta f_i = f_i - \hat{f}_i$ (where f_i is measured in cents in all cases) .

2.5.3. *Interval*

The interval is defined as the difference between the frequency of two consecutive notes $I_i = f_{i+1} - f_i$ in the target melody. The same interval in the user performance is defined as $\hat{I}_i = \hat{f}_{i+1} - \hat{f}_i$. The interval deviation is defined as $ID_i = I_i - \hat{I}_i$. This measure is key independent, so it is appropriated for a-cappella singing with no tuning reference.

2.6. Performance rating

In the performance rating stage, we combine the 8 similarity measures (2 DTW based and 6 at note-level) in order to provide three different ratings: rhythm rating, intonation rating, and overall rating. The optimal combination of the similarity measures has been considered to be the one that best fits the judgement of 4 trained musicians about 27 different singing performances. We have obtained such optimal combination through a quadratic polynomial regression analysis performed in Weka [19].

3. EVALUATION

3.1. Ground truth

We combine the use of real recordings and artificially generated melodies in order to systematically control the level of intonation and rhythm deviations. The evaluation dataset is then built by introducing random pitch/rhythm variations to three different target melodies, using an harmonic plus stochastic modelling of the input signal [20]. Three levels of random

variations have been applied for both pitch and rhythm. In total, nine combinations with different degree of error are generated from each reference melody. Therefore, 27 melodies (around 22 minutes of audio) comprise the whole evaluation dataset[1].

Human judgements were collected from four trained musicians, who were asked to score from 1 to 10 the evaluation dataset in three different aspects: intonation, rhythm and overall impression. Melodies were presented in random order using headphones.

3.2. Evaluation measures

Three different measures have been computed to evaluate the singing voice assessment system: interjudgement reliability, correlation between similarity measures and human judgements and polynomial regression error. Interjudgement reliability, proposed in [1], measures the correlation between human ratings. This measure aims to quantify the objectivity of the ratings. We have computed the correlation between the ratings for each pair of musicians (in total $n(n-1)/2 = 6$ pairs), and then averaged all the correlations. We have also computed the correlation coefficient for each similarity measure with respect to the different mean score given by musicians. This is a good reference about how meaningful each similarity measure is for performance assessment. A total of 27 (9 similarity measures $\times$ 3 ratings) correlation coefficients have been computed. Finally, the human criteria has been modelled in Weka through quadratic polynomial regression. The regression error quantifies the accuracy of the data fitting procedure. In this case, the evaluation dataset is the same as the training dataset. We consider the following measures from regression analysis: the correlation coefficient and the root mean squared error.

4. RESULTS & DISCUSSION

The mean correlation values corresponding to the interjudgement reliability measure are shown in Table 1. The results show that the agreement on rhythmic evaluation is lower. Nevertheless, the correlation in all cases is acceptable, and the case of intonation is specially good.

Type of score	Mean correlation coefficient
Intonation	0.93
Rhythm	0.82
Overall	0.90

Table 1. Results of interjudgement reliability

Table 2 shows the correlation between the different similarity measures and the human ratings. We observe a high correlation of human ratings and DTW based measures (TIE and ε_{RMS}), specially for rhythm assessment. DTW based measures do not require singing transcription, since it directly uses the low-level feature. Therefore, DTW is a simple but efficient technique for intonation and rhythm automatic assessment.

[1]Audio samples extracted from the ground truth can be found at http://www.atic.uma.es/singing

Similarity measure	Corr. with Intonation rating	Corr. with Rhythm rating	Corr. with Overall rating
TIE	0.92	0.21	0.81
ε_{RMS}	0.0012	0.81	0.52
$\overline{\Delta O}$	0.026	0.68	0.48
$\overline{\Delta O}_W$	0.037	0.68	0.48
$\overline{\Delta f}$	0.96	0.2	0.82
$\overline{\Delta f}_W$	0.89	0.23	0.82
$\overline{\Delta I}$	0.94	0.34	0.9
$\overline{\Delta I}_W$	0.87	0.35	0.87

Table 2. Correlation values of each similarity measure with the ratings given by trained musicians.

Type of error	Intonation	Rhythm	Overall
Correlation coefficient	0.988	0.969	0.976
Root mean squared error	0.4167	0.58	0.44

Table 3. Polynomial regression error.

Finally, Table 3 shows the obtained regression errors. The optimal polynomial combination of similarity measures provides high correlation with human judgements. For intonation, the results are specially good, because the chosen similarity measures are very representative and there is a high interjudgement reliability.

5. CONCLUSIONS

This paper presents a generic schema for automatic singing assessment, applied to the context of basic singing levels. The system provides the user with several ratings (intonation, rhythm and overall) by combining a set of melodic similarity measures with respect to a target melody. Target melodies are sung by a trained singer with neutral expression. The combination of f_0 alignment and symbolic similarity measures has been proven to be very appropriated for automatic rating. Furthermore, DTW based similarity measure is specially simple and effective for intonation and rhythm assessment, and such approach has not been considered in prior work. We have combined similarity measures through polynomial regression in order to fit the judgement of trained musicians. This approach then succeeds in modelling the musicians' criteria, as shown by our results. This study also contributes with a systematic evaluation methodology, applicable to other types of systems for automatic singing rating. Our approach is easily extensible to other expressive features such as vibrato or dynamics if new similarity measures are incorporated. In addition, the symbolic score of the melody could be used as target melody to avoid the need of a target singer. Finally, the proposed schema could be applied to realtime assessment if an on-line time warping algorithm [21] is integrated.

6. REFERENCES

[1] J. Wapnick and E. Ekholm, "Expert consensus in solo voice performance evaluation.," *Journal of voice official journal of the Voice Foundation*, vol. 11, no. 4, pp. 429–436, 1997.

[2] M. J. Bergee, "Faculty Interjudge Reliability of Music Performance Evaluation," *Journal of Research in Music Education*, vol. 51, no. 2, pp. 137, 2003.

[3] E. Ekholm, G. C. Papagiannis, and F. P. Chagnon, "Relating objective measurements to expert evaluation of voice quality in Western classical singing: critical perceptual parameters.," *Journal of voice official journal of the Voice Foundation*, vol. 12, no. 2, pp. 182–196, 1998.

[4] 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.

[5] Barcelona Music & Audio Technologies, "SKORE Performance Rating," *Internet, http://skore.bmat.me*, 2008.

[6] O. Mayor, J. Bonada, and A. Loscos, "The singing tutor: Expression categorization and segmentation of the singing voice," *Proceedings of the AES 121st Convention*, 2006.

[7] D. Rossiter and D. M. Howard, "ALBERT: a real-time visual feedback computer tool for professional vocal development.," *Journal of voice official journal of the Voice Foundation*, vol. 10, no. 4, pp. 321–336, 1996.

[8] Sony Computer Entertainment Europe, "Singstar," 2004.

[9] J. Callaghan and P. Wilson, *How to Sing and See: Singing Pedagogy in the Digital Era*, Cantare Systems, 2004.

[10] S. Grollmisch, E. Cano Cerón, and C. Dittmar, "Songs2see: Learn to play by playing," *Watermark*, vol. 1, 2012.

[11] A. De Cheveigne and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[12] Hiroaki Sakoe, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, pp. 43–49, 1978.

[13] C. A. Ratanamahatana and E. Keogh, "Everything you know about dynamic time warping is wrong," in *3rd Workshop on Mining Temporal and Sequential Data, in conjunction with the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004)*, 2004.

[14] D. Ellis, "Dynamic time warp (DTW) in Matlab," *Internet, http://labrosa.ee.columbia.edu/matlab/dtw. Last view: 29/11/2012*, 2003.

[15] I. Barbancho, C. de la Bandera, A.M. Barbancho, and L.J. Tardon, "Transcription and expressiveness detection system for violin music," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, april 2009, pp. 189 –192.

[16] E. Molina, E. Gómez, and Barbancho I., "Automatic scoring of singing voice based on melodic similarity measures," M.S. thesis, Universitat Pompeu Fabra, Music Technology Group, 2012.

[17] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson, and S. J. Cunningham, "Towards the digital music library: Tune retrieval from acoustic input," *Lloydia Cincinnati*, , no. 1978, pp. 11–18, 1996.

[18] J. Bednar and T. Watt, "Alpha-trimmed means and their relationship to median filters," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 32, no. 1, pp. 145–153, 1984.

[19] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[20] E. Gómez, G. Peterschmitt, X. Amatriain, and P. Herrera, "Content-based melodic transformations of audio material for a music processing application," *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx)*, 2003.

[21] S. Dixon, "Live tracking of musical performances using on-line time warping," *Proceedings of the 8th International Conference on Digital Audio Effects (DAFx)*, 2005.

B.2

Molina, E., Barbancho, A. M., Tardón, L. J., and Barbancho, I. (2014). Dissonance reduction in polyphonic music using harmonic reorganization. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 22(2):325-334.

Dissonance Reduction In Polyphonic Audio Using Harmonic Reorganization

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

***Abstract*—In this paper, a method for automatic reduction of dissonance in recorded isolated chords is proposed. Previous approaches address this problem using source separation and note-level processing. In our approach, we manipulate the harmonic structure as a whole in order to avoid beating partials which, according to prior research on dissonance perception, typically produce an unpleasant sound. The proposed system firstly performs a sinusoidal plus residual modeling of the input and analyses the various fundamental frequencies present in the chord. This information is used to create a symbolic representation of the in-tune version of the input according to some musical rules. Then, the partials of the signals are shifted in order to fit the in-tune harmonic structure of the input chord. The input is assumed to contain one isolated chord, with relatively stable fundamental frequencies belonging to the Western chromatic scale. The evaluation has been performed by 31 expert musicians, which have quantified the perceived consonance of six varied, out-of-tune chords in three variants: unprocessed, processed with our system and processed by a state-of-the-art commercial tool (Melodyne Editor). The proposed approach attains an important reduction of the perceived dissonance, showing better performance than Melodyne Editor for most of the cases evaluated.**

***Index Terms*—Audio analysis and synthesis, audio for multimedia, content-based music processing, music processing systems.**

I. Introduction

MUSICAL Tuning has been an important object of study along history. A *tuning system* defines which tones, or pitches, are used when playing music. The first written evidence related to the tuning of instruments belongs to the old Babylon (around 1500 BC), where a detailed description of the Babylonian harp tuning is described in cuneiform script [1]. Such tuning system was based on the frequency ratio 3:2 (perfect fifth). Later, Pythagoras (sixth century BC) developed such tuning system based on the perfect fifth in order to define the well known Pythagorean tuning [2]. Such strong relationship between mathematics and harmony has been studied by many later scientists and musicologists, among which Zarlino was especially important during the Renaissance.

During the process of music recording in modern studios, the tuning of instruments and vocals is an important aspect to take into account [3]. Depending on the style, the presence of out-of-tune sounds can be a reason to repeat a take. However this is not always possible due to technical limitations of the musicians or, simply, due to a lack of time or economic resources. In the case of monophonic instruments or voice, there are many software tools that can be used during a post-production process for tuning adjustment. For instance, *Melodyne Studio* (Celemony[1] 2003) or *Auto-Tune* (Antares Technology[2] 1997) have been widely used for the tuning of vocals in recording studios during the last years.

This task becomes much harder when dealing with polyphonic recordings. Indeed, despite the fact that polyphonic transcription and source separation are trending topics within the research community, current solutions are not fully practical for professional post-processing purposes. The best commercial solution for such problems is *Melodyne Editor* (Celemony 2009). This software addresses the polyphonic tuning problem through multiple-f_0 estimation, source separation and pitch-shifting [4]. However, in the case of out-of-tune chords Melodyne does not perform source separation accurately, and beating partials are still present in the apparently corrected chord [5]. The presence of beating partials is related to the perceived *dissonance* of a sound [6].

The concept of dissonance can be interpreted differently depending on the context. On the one hand, the musical dissonance is defined as the interval that, according to the classical harmony rules, is unpleasant to the ear [7]. Typically the intervals of minor second (1 semitone), major seventh (11 semitones) and tritone (6 semitones) are considered dissonant. On the other hand, sensory dissonance is defined in perceptual terms as the 'roughness' of a sound, and it can be applied either to musical or non musical sounds. This kind of dissonance has been addressed by many authors [8], but the most important study about dissonance perception was carried out by Plomp & Leveltz in 1965 [6]. The main contribution of this work was to relate the perception of dissonance to the concept of critical bands proposed by

Manuscript received April 16, 2013; revised July 23, 2013; accepted October 04, 2013. Date of publication October 23, 2013; date of current version December 31, 2013. This work was supported by the Ministerio de Economía y Competitividad of the Spanish Government under Project No. TIN2010-21089-C03-02 and Project No. IPT-2011-0885-430000, by the Junta de Andalucía under Project No. P11-TIC-7154 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 Prof. Søren Holdt Jensen.

The authors are with the Department of Ingeniería de Comunicaciones, E.T.S.I. Telecomunicación, University of Málaga, 29010 Malaga, Spain (e-mail: emm@ic.uma.es; abp@ic.uma.es; lorenzo@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.2013.2287056

[1]http://www.celemony.com

[2]http://www.antarestech.com

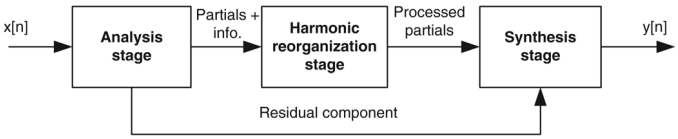

Fig. 1. General scheme of the selected approach for polyphonic dissonance reduction.

Harvey Fletcher in the 1940's [9]. According to Plomp & Leveltz, two tones are perceived as dissonant when they fall within the same critical band.

In this paper, we propose a novel approach for dissonance reduction in polyphonic audio, processing harmony as *a whole* instead of performing source separation. We address both the reduction of *musical dissonance* and *sensory dissonance*. The proposed method assumes that the input chord is stable (f_0s are relatively constant along time) and it is composed of harmonic sounds (i.e. the overtones are placed at integer multiples of each f_0). Some subjective factors related to harmony have been predefined to achieve a good compromise for a practical use in common recording studios. Specifically, we assume that Western music is analyzed.

The selected approach is based on an *analysis-resynthesis* scheme (Fig. 1). This approach is based on the sinusoidal plus residual modeling scheme proposed in [10]. The input to the system is a mono audio signal $x[n]$ containing the original dissonant sound, and the output is a processed mono audio signal $y[n]$. The developed system can be divided into three main blocks:

- **Analysis stage:** The parameters of the sinusoidal component of the signal are extracted and, also, separated. This block is mainly based on the techniques described in [10].
- **Harmonic reorganization stage:** This is the core of the system in which the most interesting techniques presented in this paper are implemented. In this stage, the sinusoidal parametrization previously obtained is manipulated in order to reduce the dissonance of the original sound.
- **Synthesis stage:** This block synthesizes the audio signal making use of the sinusoidal parametrization developed after the Harmonic reorganization stage. It is mainly based on the overlap-add technique [11].

This paper is organized according to the scheme described. The three main blocks are explained in Sections II, III and IV. The evaluation methodology is detailed in Section V, and the results obtained are discussed in Section VI. Finally, in Section VII, a summary of conclusions and contributions about the present work has been included.

II. Analysis Stage

In this section, the *Analysis stage* is described. Some main aspects of the sinusoidal plus residual modeling are presented, namely: sinusoidal estimation (Section II-A), partials tracking (Section II-B) and extraction of the residual component (Section II-C). Also, the method selected for multiple f_0 extraction is explained in Section II-D. A diagram block of the analysis stage is shown in Fig. 2.

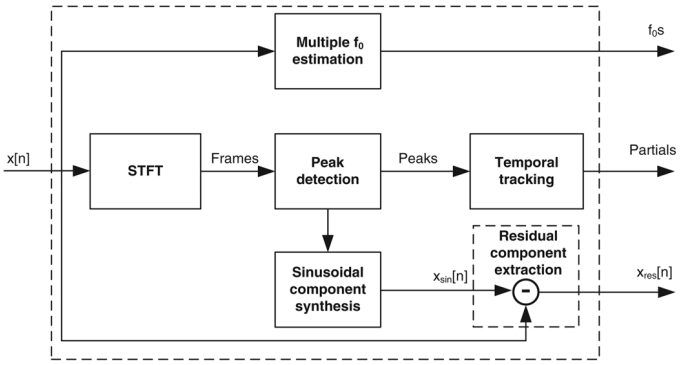

Fig. 2. Block diagram of the analysis stage.

A. Sinusoidal Plus Residual Modeling

The input chords are analyzed using a *sinusoidal plus residual* approach. This model separates the signal $x[n]$ into a sum of stable sinusoids $x_{sin}[n]$ and a residual component $x_{res}[n]$: $x[n] = x_{sin}[n] + x_{res}[n]$ [10]. This model is especially useful for the analysis of sustained musical sounds.

In our approach, the parameters of the model are estimated frame by frame using the STFT. Specifically, for each frame $l \in [1, L]$, three values are estimated: amplitude $\hat{A}_r^l$, frequency $\hat{\omega}_r^l$ and phase $\hat{\phi}_r^l$ corresponding to each partial $r \in [1, R]$. With these three parameters, the sinusoidal component can be synthesized according to:

$$x_{sin}^l[n] = \sum_{r=1}^{R^l} \hat{A}_r^l \cos(\hat{\omega}_r^l \cdot n + \hat{\phi}_r^l) \tag{1}$$

The parameters $(\hat{A}_r^l, \hat{\omega}_r^l, \hat{\phi}_r^l)$ are estimated from the spectrogram $\tilde{X}[l, k]$ of the windowed signal, where l is the frame index and k denotes the frequency value in bins. The STFT computes the N-point spectrum of consecutive, windowed excerpts of length M with a certain hop-size H. Depending on the parameters (H, N, M and window function $w[n]$), the computed spectrogram can vary in terms of resolution and presence of secondary lobes [12]. In the proposed method, low presence of secondary lobes and good frequency resolution, rather than temporal resolution, are desired. Recall that the input sounds are expected to be stable along time, therefore temporal resolution is not considered a critical point. The chosen window $w[n]$ is Blackman-Harris 92 dB with parameters $M = 8001$, $N = 8192$ and $H = 2048$ for a sample rate $f_s = 44100$ Hz. The attenuation of the secondary lobes with respect to the main lobe in this window is 92 dB, and the width of the main lobe is $W_m = 8$ bins. The frequency resolution achieved with this window is:

$$\Delta f = W_m \frac{f_s}{M} \approx 44 \text{ Hz}. \tag{2}$$

Note that in low registers, this frequency resolution could be insufficient for one-semitone distances. This situation has been considered in the *Beating reduction* block of the *Harmonic reorganization* stage (see details in Section III-C).

B. Estimation and Tracking of Sinusoids

The sinusoidal component is estimated frame by frame by analyzing the existing peaks within the spectrum, since a stable sharp peak in frequency corresponds to a stable sinusoid. According to the chosen approach [10], a local maximum above a certain threshold t is considered a peak. A peak is detected at $k = k_r$ if $|X_w[k_r]|$ is larger than its neighboring values in the magnitude spectrum and larger than t. Note that the precise frequency value k_r^* of the r sinusoid could correspond to a non-integer bin value. So, we estimate k_r^* by finding the maximum of the parabola that fits $|X_w[k_r - 1]|$, $|X_w[k_r]|$ and $|X_w[k_r + 1]|$, which is computed through parabolic interpolation as explained in [13]. Additionally, only the largest thirty peaks are selected to avoid noisy regions to be estimated as sinusoids.

Once the peaks have been detected in a frame of the signal, a temporal tracking is performed to group them into the same partial. In this way, we can process each partial independently to maintain the naturalness of the sound. The chosen method for time tracking, proposed in [10], is simple but effective. It connects sinusoids that are close in time, frequency and magnitude. Note that we have assumed that the input sounds are static chords with partials that remain relatively stable along time, so the time tracking approach chosen works well. With our set of parameters, the algorithm considers two spectral peaks to be grouped into the same partial if they fall within a three dimensional mask defined as follows:

(a) they are close in time (< 70 ms)
(b) they are close in frequency (< 0.2 semitones)
(c) they are close in magnitude (< 20 dB)

The mask is wide to allow the tracking of beating partials, which could oscillate in magnitude and frequency. If more than two consecutive peaks fall into the same mask, the Euclidean distance in the three dimensions, time-frequency-magnitude, is considered to decide the correct track of each partial.

The short partials are discarded because we consider that they do not contribute to the perceived dissonance. According to Moore [14], 200 ms is an acceptable duration to correctly perceive the pitch of a sound, therefore we discard all the partials shorter than this value. The discarded short sinusoids are not lost, but kept in the residual component of the signal.

C. Residual Component Extraction

The usual procedure to extract the residual part is to subtract a synthesized version of the sinusoidal component from the original signal: $x_{res}[n] = x[n] - x_{sin}[n]$. The quality of the residual component is directly dependent on the sinusoidal estimation. If the sinusoidal component is properly estimated, the residual component should contain just transients and noise. Observe that phase coherence in the subtraction $x[n] - x_{sin}[n]$ is very important, otherwise a clean residual part would not be obtained.

Since the residual component is not processed at all, the transients and noisy aspects of the signals remain unaltered in the output signal.

D. Multiple-f_0 Estimation

The *Multiple-f_0 estimation* stage analyses the input chord in order to compute a vector of estimated fundamental frequencies $\widehat{\mathbf{f_0}} = [\widehat{f}_{01}, \widehat{f}_{02} \dots \widehat{f}_{0n}]$. Ideally, $\widehat{\mathbf{f_0}}$ would equal the vector $\mathbf{f_0} = [f_{01}, f_{02} \dots f_{0n}]$, which contains the actual fundamental frequencies of the input chord. We assume that these frequencies do not change along time, i.e. we are dealing with an isolated chord. If the input signal consists of a sequence of chords, it must be segmented by the user in order to process each chord separately.

Many f_0 estimation methods have been proposed in the literature [15]–[18]. In our approach, we have used the multiple-f_0 estimation algorithm proposed by Klapuri in 2005 [16] because it is relatively straightforward to implement and it outperforms other reference methods (such as [17] and [18]). This method consists of a computational model of the human auditory periphery, followed by a periodicity analysis mechanism. Estimation of multiple fundamental frequencies is achieved by cancelling each detected sound from the mixture and by repeating the estimation process with the residual. Therefore, three steps can be distinguished:

1) *Auditory filter bank and neural transduction:* The acoustic signal $x(n)$ is filtered by a set of auditory filters uniformly distributed in the critical-band scale [19]. The auditory nerve signal for each channel c is then modeled by a cascade of (i) compression, (ii), half-wave rectification and (iii) low pass filtering.
2) *Periodicity analysis:* In this stage, each channel is analyzed through several operations based on the Fourier Transform. The periodicity information of all the channels is combined to generate a summary magnitude spectrum (SMS). This information leads to the computation of the salience function $\lambda(\tau)$, which represents the strength of each period candidate τ. Finally, the function $\lambda(\tau)$ is normalized in order not to favour either high or low f_0s in order to generate the final salience function $\tilde{\lambda}(\tau)$.
3) *Iterative estimation and cancellation:* The global maximum of $\tilde{\lambda}(\tau)$ is a robust indicator of one of the correct f_0s in polyphonic signals. However, the next-highest weight was often assigned to half or twice of the firstly detected f_0. So, an iterative procedure has been developed in which the cumulative spectrum of the detected f_0s is synthesized and, then, subtracted from the original signal in order to iteratively remove the detected f_0s from the mixture.

We apply this algorithm frame by frame with the following parameters: sampling rate $= 44100$ Hz, window size $= 4096$ samples, hop-size $= 2048$ samples, minimum $f_0 = 50$ Hz, maximum $f_0 = 3$ kHz, 60 auditory filters from 60 Hz to 5 kHz, and compression factor $\nu = 0.33$ for the neural transduction modeling. The degree of polyphony is set to $n = 5$ by default, but this parameter that can be modified by the user.

III. Harmonic Reorganization Stage

The *Harmonic reorganization stage* (see Fig. 1 and 3) is the main contribution of this paper, since the methods employed have been specially designed for the goal described. This section is organized according to the block diagram shown in Fig. 3. There are two parallel paths in this block: the scale fitting of $\widehat{\mathbf{f_0}}$ and the processing of the sinusoidal component. The upper path computes the target fundamental frequencies vector $\widehat{\mathbf{f_0^*}}$ (explained in Section III-A) and then generates a grid of overtones

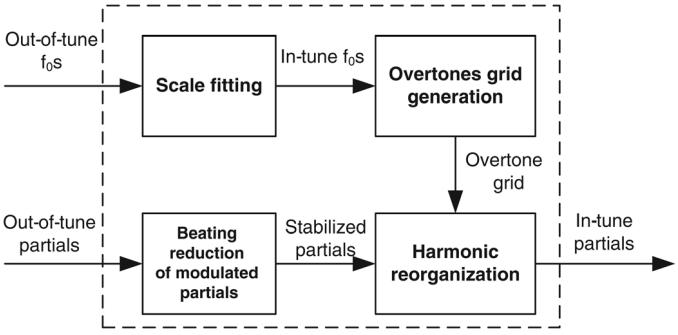

Fig. 3. Block diagram of the Harmonic reorganization stage.

corresponding to the corrected chord (Section III-B). The lower path processes the partials in two steps: first, a beating reduction stage removes the modulation of partials that are merged (due to low frequency resolution, as explained in Section III-C), and then the processed partials are shifted to conform to the in-tune output chord (Section III-D).

A. Scale Fitting

Let $\mathbf{f_S} = [f_{S1}, f_{S2} \dots f_{SN_S}]$ be a vector of frequencies corresponding to N_S notes of a given scale along several octaves. The *Scale fitting* block substitutes each value of the vector $\widehat{\mathbf{f_0}}$ by its closest value in $\mathbf{f_S}$ in order to generate $\widehat{\mathbf{f_0^*}} = [\widehat{f_0^*}_1, \widehat{f_0^*}_2 \dots \widehat{f_0^*}_n]$ (corrected f_0s). The distance between notes is measured in cents, because all the frequencies have been converted to MIDI numbers using:

$$MIDI = 69 + 12\ \log_2\left(\frac{f}{440}\right) \tag{3}$$

Therefore, the vector $\widehat{\mathbf{f_0^*}}$ contains the target frequency of each note in the 'in tune' version of the input chord. Note that this stage just handles symbolic information, and it does not apply any processing to the input chord.

We assume that the vector $\mathbf{f_S}$ is made of notes from of the Western tempered chromatic scale. The following three cases have been considered to cover a wide range of practical situations:

1) $\mathbf{f_S}$ *is the whole chromatic scale:* The first approach makes use of the tempered chromatic scale. We consider that the notes of the input chord are out-of-tune if they deviate from the MIDI scale [20] (i.e. the tempered chromatic scale). During the tuning adjustment, every element in $\widehat{\mathbf{f_0}}$ is simply rounded to the closest integer. In this case, no musical assumptions about the input data have been made and deviations larger than one semitone would imply rounding to an incorrect note. This approach is useful for cases in which there is not additional information about the input material.
2) $\mathbf{f_S}$ *depends on* $\widehat{f}_{01}$*:* The *Scale fitting* process can be improved if the user provides the system with some musical knowledge about the input chord. Different presets can be selected by the user: *Pentatonic scale*, *Major scale*, *Minor scale*, *Major chord (root position)*, *Major chord (any inversion)*, etc. The scale $\mathbf{f_S}$ is built upon the lowest note of the chord, $\widehat{f}_{01}$, which is taken as the root note. The resulting scale contains musically meaningful notes related to the input chord. In Fig. 4, an example in which the use of the major scale has been assumed is shown. The lowest note of the input chord is C, and the used scale is C major. The note $F\#4 + 9$ cents has been moved to $G4$ because $F\#4$ is not admitted by the C major scale. The system implemented allows the user to define customized musical constraints. This is useful if the user knows which type of chords must be obtained.
3) $\mathbf{f_S}$ *is customized by the user:* Sometimes, the user knows exactly all the notes of the input chord. In such case, the user provides the system with all the notes that comprise the input chord. The user imposes $\mathbf{f_S} = \mathbf{f_0}$, and so each element in $\widehat{\mathbf{f_0}}$ is rounded to the desired frequency. The customization of $\mathbf{f_S}$ by the user allows the system to achieve good results with uncommon chords.

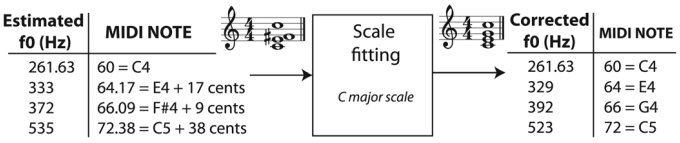

Fig. 4. Adjustment with musical restrictions of a largely out-of-tune C major chord.

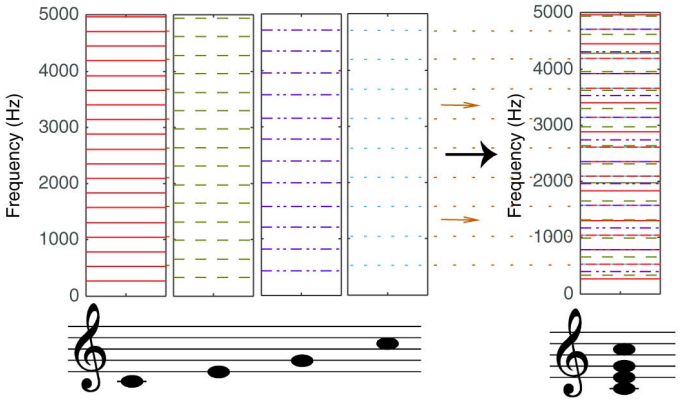

Fig. 5. Generation of overtone grid. The overtones of every note (left) are combined into a single grid for the complete chord (right).

B. Overtone Grid Generation

At this stage, the target frequencies of the notes comprising the output chord, $\widehat{\mathbf{f_0^*}} = [\widehat{f_0^*}_1, \widehat{f_0^*}_2 \dots \widehat{f_0^*}_n]$, are known. Then, we compute the partials structure $\widehat{\mathbf{f^*_{H_{whole}}}}$ of the whole chord to define the so called *overtone grid*. The overtone grid is made of a combination of the harmonics of every f_0^* within the corrected chord. We denote the harmonic structure of each note as follows:

$$\widehat{\mathbf{f^*_{H1}}} = [\widehat{f_0^*}_1, 2\widehat{f_0^*}_1 \dots R\widehat{f_0^*}_1] \tag{4}$$
$$\widehat{\mathbf{f^*_{H2}}} = [\widehat{f_0^*}_2, 2\widehat{f_0^*}_2 \dots R\widehat{f_0^*}_2] \tag{5}$$
$$\vdots \tag{6}$$
$$\widehat{\mathbf{f^*_{Hn}}} = [\widehat{f_0^*}_n, 2\widehat{f_0^*}_n \dots R\widehat{f_0^*}_n] \tag{7}$$

We have defined the maximum number of harmonics to be $R = 20$, since the energy of the harmonic content over this value can be assumed to be very low. Consequently, the minimum frequency of the overtone grid is the first harmonic of the lowest note, and the maximum frequency is the twentieth harmonic of

the highest note. The frequencies of all the partials of the chord are sorted in a single array in order to define $\widehat{\mathbf{f}}^*_{\mathbf{H}\,\mathrm{whole}}$.

In Fig. 5, this process is illustrated for a C major chord in root inversion ($\widehat{\mathbf{f}}^*_{\mathbf{0}} \rightarrow [C4, E4, G4, C5]$). These notes correspond to the following frequency values in Hz:

$$\widehat{\mathbf{f}}^*_{\mathbf{0}} = [261.63, 329.63, 392.00, 523.25]\ \mathrm{Hz}$$

According to the explained procedure, the overtone grid of the whole chord would be:

$$\widehat{\mathbf{f}}^*_{\mathbf{H}\,\mathrm{whole}} = [261.63, 329.62, 392.00, 523.25, 659.24, 784.00, 1046.5, \ldots, 10465]\ \mathrm{Hz} \quad (8)$$

Note that all the input notes comprising the chord are supposed to be harmonic, i.e. the overtones are placed in multiples of the fundamental frequency. Typically, major and minor chords contain harmonically related notes, so there is a large number of overlapped overtones. Indeed, this overlapping reduces the perceived dissonance, because beating partials are avoided [21].

The overtone grid generation is robust to certain types of errors in the multiple-f_0 estimation. For instance, an error in the degree of polyphony is not critical if the overlap between harmonics within the chord is relatively high. This is especially noticeable when one note and its octave are present in the chord, since the upper octave does not contribute to a more complete overtone grid. In the same way, octave errors in the multiple-f_0 estimation process (specially when $\widehat{f}_{01} = f_{01}/2$) or fifth errors do not introduce significant changes to the harmonic structure of the whole chord.

C. Beating Reduction of Modulated Partials

In this section, the behavior of merged beating partials (usually found in out-of-tune sounds at low frequencies) is analyzed and a method to avoid them is proposed.

Recall that, as shown in eq. (2), the frequency resolution achieved in the Analysis stage is around 44 Hz. This is less than one semitone for notes below $F\#5$ (due to the logarithmic scaling of the frequency axis). This resolution is definitely too low to resolve beating sinusoids in low frequency, out-of-tune sounds. When two partials are not independently estimated, a single peak with periodic oscillations of amplitude and frequency is detected instead of two stable peaks (i.e. the partials are *merged* during the analysis). These oscillations are usually found in out-of-tune sounds and they increase the perceived dissonance [6].

1) Mathematical Analysis of Beating Sinusoids: We present a mathematical analysis of two common situations in order to justify the proposed beating reduction method.

a) Beating sinusoids with the same amplitude: Let $x(t)$ be the sum of two sinusoids with similar frequencies and equal amplitude. As shown in eq. (9), this is equivalent to a product of a carrier and a modulating sinusoid (amplitude modulated tone):

$$\begin{aligned} x(t) &= A\cos((\omega_0 + \Delta\omega)t) + A\cos((\omega_0 - \Delta\omega)t) \\ &= 2A \cdot \cos(\Delta\omega_0 t) \cdot \cos(\omega_0 t) \end{aligned} \quad (9)$$

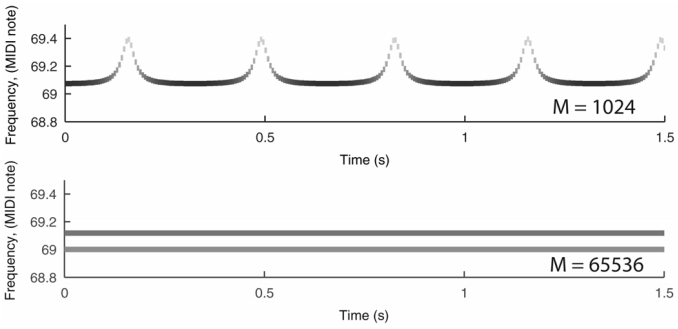

Fig. 6. Peak frequency spectrogram of two beating sinusoids (440 Hz and 443 Hz) with different amplitude (-12 dB and -9 dB respectively) for two different window sizes: $M = 1024$ and $M = 65536$.

If the analysis window used with the STFT is too narrow, the representation of the result becomes close to a amplitude modulated tone. On the other hand, if the window is large enough, two different peaks can be distinguished.

The instantaneous frequency of $x(t)$ in this case is constant and equals ω_0. Due to this, small window sizes provide a constant frequency value but a modulated amplitude.

b) Beating sinusoids with different amplitude: Let $x(t)$ be a sum of two sinusoids with similar frequencies and different amplitudes: $x(t) = A_1\cos((\omega_0+\Delta\omega)t)+A_2\cos((\omega_0-\Delta\omega)t)$. In this case, according to [22], this signal can be expressed as:

$$\begin{aligned} x(t) = &\sqrt{A_1^2 + A_2^2 + 2A_1A_2\cos(2\Delta\omega\, t)} \\ &\cdot \cos\left(\omega_0\, t + \arctan\left(\Delta\omega\, t \cdot \frac{A_1 - A_2}{A_1 + A_2}\right)\right) \end{aligned} \quad (10)$$

The instantaneous frequency, $\hat{\omega}(t)$, can be extracted by taking the derivative of the instantaneous phase in eq. (10):

$$\hat{\omega}(t) = \omega_0 + \frac{(A_1^2 - A_2^2)\Delta\omega}{A_1^2 + A_2^2 + 2A_1A_2\cos(2\Delta\omega t)} \quad (11)$$

The instantaneous frequency is constant when $A_1 = A_2$. However, if $A_1 \neq A_2$, a type of periodic modulation in frequency appears, which explains the presence of strange periodic patterns in the STFT of polyphonic sounds out-of-tune. Fig. 6 shows the peak frequency spectrogram of two beating sinusoids as an example of this situation. The peak frequency spectrogram only shows the local maxima over a given threshold (-80 dB in our case) of the common spectrogram.

2) Proposed Method for Beating Reduction: The proposed beating reduction method removes amplitude and/or frequency modulations. Several assumptions about the input chord are made:

- It is a stable out-of-tune chord whose f_0s do not vary along time. Vibratos or glissandos are not addressed.
- Attack-Decay-Sustain-Release envelope (ADSR) [23] is assumed in the amplitude of the partials. Tremolo or atypical envelope patterns are not addressed.

The contribution of the beating reduction stage to the perceived consonance strongly depends on the type of signal. The improvement attained is especially noticeable when the chord is simple and stable (e.g. perfect major/minor synthetic chords in our evaluation dataset). In contrast, if the input is a significantly time-varying signal in terms of timbre and frequency, the

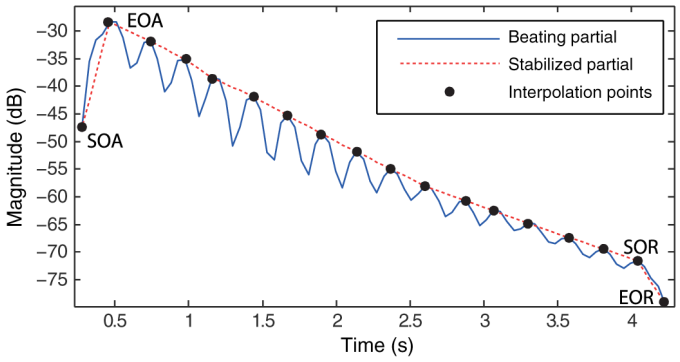

Fig. 7. Magnitude envelope stabilization of a beating partial. The attack and the release of the note are fixed, and only beatings are removed.

beating reduction scheme can produce a lack of naturalness or introduce artefacts, in such case, better results are achieved if it is disabled. Three steps are performed in our beating reduction scheme: Envelope reconstruction, Frequency stabilization and Phase reconstruction.

a) Envelope reconstruction: First, the amplitude envelope is processed to fit the ADSR model by using a variation of the envelope reconstruction procedure presented in [23]. The ADSR model considers several split-points in the envelope of the signal: start of attack (*SOA*), end of attack (*EOA*), start of release (*SOR*), and end of release (*EOR*) (see Fig. 7). Our approach defines each split-point as follows:

- SOA: First amplitude value above -80 dB (noise threshold).
- EOA: First local maximum of the envelope.
- SOR: Last local maximum of the envelope.
- EOR: Last amplitude value above -80 dB.

The envelope reconstruction proposed in [23] only considers these four split-points. In our approach, we also take every local maximum between the EOA and the SOR (interval called sustain or decay) as a set of tracking points in order to faithfully fit the original envelope. The interval between each split-point is modeled by an exponential curve, as proposed in [23]. Note that, if the amplitude is represented in dBs (logarithmic scale), exponential curves become straight lines. Our approach for envelope reconstruction performs linear interpolation of the split and tracking points in a logarithmic scale.

b) Frequency stabilization: Frequency modulations might appear in beating sinusoids, as shown in eq. (11). The proposed technique removes such modulations by time-averaging the measured frequencies of the partial. From an analytical point of view, this can be understood as averaging the instantaneous frequency $\hat{\omega}(t)$ calculated in (11). Note that the average of $\hat{\omega}(t)$ can be bounded in the following interval:

$$\left[\omega_0 + \frac{A_1 - A_2}{A_1 + A_2}\Delta\omega,\ \omega_0 + \frac{A_1 + A_2}{A_1 - A_2}\Delta\omega\right] \tag{12}$$

simply substituting $\cos(2\Delta\omega t)$ in (12) by its maximum and minimum values, 1 and -1, respectively:

This result is useful to know the approximate range of the computed average. Later, this average will be moved to a fixed grid to finally generate the output, as it will be explained in later sections.

c) Phase reconstruction: If the frequency of a partial is changed, the phase evolution has to be adapted to guarantee the continuity of the sinusoid. The phase state for every frame l is estimated as follows:

$$\phi_r^l = \phi_r^{l-1} + \frac{H \cdot 2\pi f_r}{f_s} \tag{13}$$

Where ϕ_r^l is the new phase value for partial r in frame l, H is the hop-size, f_r is the new frequency value and f_s is the sampling rate.

D. Harmonic Reorganization

The final step in the processing is *harmonic reorganization*, which shifts each partial of the input chord in order to fit the overtone grid $\widehat{\mathbf{f}^*_{\mathbf{H}\,\text{whole}}}$ defined in Section III-B, thus, the perceived consonance in sustained chords is improved. We observed that harmonic reorganization is responsible for the highest dissonance reduction in most of the cases evaluated. This process is performed in several steps:

1) **Parametrization of individual partials:** As explained in Section II-A, each partial r is defined by three vectors: $(\hat{A}_r^l, \hat{f}_r^l, \hat{\phi}_r^l)$. We take the average $\overline{\hat{f}_r} = \sum_{l=1}^{l=L} \hat{f}_r^l / L$ as a single representative value of the frequency of the partial. The complete array of average frequencies corresponding to all the partials is called $\widehat{\mathbf{f}_{\mathbf{H}\,\text{whole}}}$, according to the notation used in Section III-B.
2) **Search for the target frequency of each partial:** The vector $\widehat{\mathbf{f}_{\mathbf{H}\,\text{whole}}}$ and the overtone grid $\widehat{\mathbf{f}^*_{\mathbf{H}\,\text{whole}}}$ (defined in Section III-B) are expressed in MIDI numbers, using eq. (3). Then, for each element in $\widehat{\mathbf{f}_{\mathbf{H}\,\text{whole}}}$, the nearest element in $\widehat{\mathbf{f}^*_{\mathbf{H}\,\text{whole}}}$ in semitones is taken as its target frequency.
3) **Shifting the partials:** Let $f_{diff} = \widehat{f^{*r}_{H\ \text{whole}}} - \overline{\hat{f}_r}$ be the distance, in semitones, to the target frequency of the partial r, where $\overline{\hat{f}_r}$ is the average frequency value of the partial r, and $\widehat{f^{*r}_{H\ \text{whole}}}$ is the closest frequency value in the overtone grid (target frequency). Then, the frequency shift applied to each partial is:

$$\Delta f = \left\{ \begin{array}{cc} f_{\text{diff}} & \text{si} \quad f_{\text{diff}} < 3 \text{ semitone} \\ 0 & \text{si} \quad f_{\text{diff}} > 3 \text{ semitone} \end{array} \right\} \tag{14}$$

This condition is applied to avoid shifts of partials to excessively distant positions. In addition, the partials out of the minimum and the maximum frequencies of the overtone grid (respectively, the first harmonic of the lowest note and the 20th harmonic of the highest note of the chord, respectively; see Section III-B) remain unchanged.

Finally, the frequency vector of the corrected partial is defined as $\hat{f}_r^{*^l} = \hat{f}_r^l + \Delta f_r$, and the complete set of parameters for each partial r is: $(\hat{A}_r^l, \hat{f}_r^{*^l}, \hat{\phi}_r^l)$.

In Fig. 8, the peak frequency spectrogram of an out-of-tune guitar chord at different stages of the system developed is shown. A reference spectrum obtained analyzing the same chord played with an in-tune guitar is also presented (Fig. 8(d)). Frequency and amplitude oscillations along time are noticeable

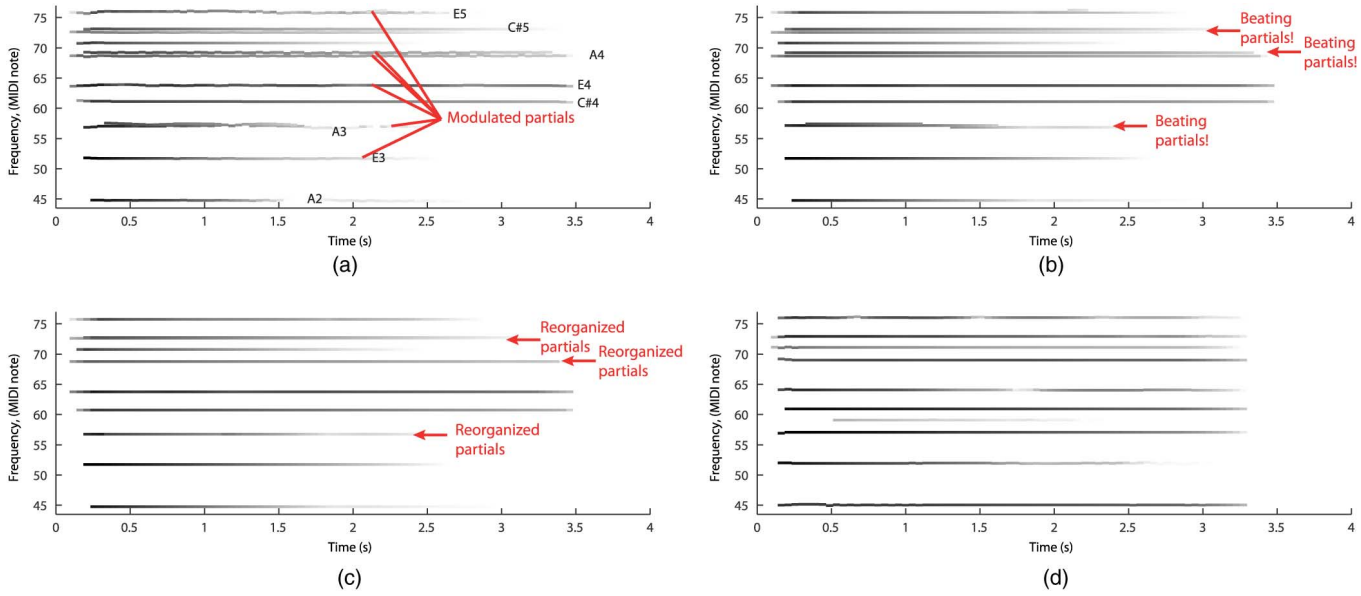

Fig. 8. Detail of the peak frequency spectrograms of several versions of a A major chord played with acoustic guitar. **(a)** Original out-of-tune chord, whose notes are (name and MIDI number): $A2-33$ cents $=44.77$, $E2-33$ cents $=51.77$, $A3+27$ cents $=57.27$, $C\#4+16$ cents $=61.16$, $E4-20$ cents $=63.80$. **(b)** Original chord after the *beating reduction* stage **(c)** Original chord after the *beating reduction* and the *harmonic reorganization* stages. **(d)** A major chord played with a real in-tune guitar.

in the partials of the original chord (the reason for this was discussed in Section III-C). The beating reduction stage removes these oscillations, and then the harmonic reorganization shifts the partials in order to produce a spectrogram more similar to a real in-tune chord. As long as the most important f_0s are correctly estimated, the harmonic reorganization stage is almost free of artifacts for sustained and stable chords.

In the case of time-varying signals, the analysis stage produces less representative parameters, and beating reduction together with harmonic reorganization stages can produce artefacts. In Section V, we provide some comments about specific examples.

IV. Synthesis Stage

With the parameters of the partials modified to be in-tune, a resynthesis stage of the sinusoidal component is performed. This process is carried out frame by frame through an *overlap-add* process. Every frame is synthesized in the frequency domain and converted into a short windowed waveform making use of the inverse FFT. Finally, all the frames are overlapped and added to give rise to the final output waveform. Some previous methods are directly based on the inverse FFT operation, such as [10] or [24].

In the chosen approach [10], the synthesis of one single sinusoid in the frequency domain is straightforward due to the use of Blackman-Harris 92 dB window. In this type of window, secondary lobes are negligible and only the main lobe is needed to accurately represent a sinusoid in the spectrum. For each sinusoid, a lobe containing frequency, magnitude and phase information is placed in the spectrum.

The synthesis of the complete waveform is performed overlapping all the frames. For the case of the Blackman-Harris window, the constant overlap factor is 75%. The spectrum of each frame is used to create a short windowed waveform through the IFFT, and this process is repeated for all the frames. Then all the short waveforms are overlapped and added to create the final sinusoidal output [12]. The last step of the synthesis is adding the residual component (which was extracted during the analysis stage) to give rise to the final output waveform.

V. Evaluation Methodology

The evaluation methodology is based on questionnaires that have been answered by 31 expert musicians. Such musicians have been asked to rate the perceived consonance of a set of chords in a dataset.

A. Dataset

The dataset contains 18 different chords. More specifically, there are 3 versions of 6 different types of out-of-tune chords[3]. The sounds are increasingly complex (from synthetic stable sounds to real chamber ensembles). Most of the chosen sounds are major chords, because they are very common in Western music and the difference between in-tune and out-of-tune chords is quite noticeable:

- Type of out-of-tune chords
  1) C Major played with 6 harmonic complex tones with ADSR envelope. Notes: $C4$, $E4+11$ cents, $G4-21$ cents, $C5+30$ cents. The single notes were artificially synthesized and then combined.
  2) C minor played with 6 harmonic complex tones with ADSR envelope. Notes: $C4$, $E\flat4+13$ cents, $G4+17$ cents, $C5-32$ cents. As the previous case, the notes were artificially synthesized and then combined.
  3) A Major played with a real acoustic guitar. Notes: $A2-33$ cents, $E2-33$ cents, $A3+27$ cents, $C\#4+16$ cents, $E4-20$ cents. The guitar was deliberately left out-of-tune to sound strongly dissonant, and all the strings

[3]These sounds are available at http://www.atic.uma.es/polytuning

were played together. Then, each note was separately analyzed to find out its accurate frequency value.

4) D Major played with a real acoustic guitar. Notes: $D3-30$ cents, $A3+28$ cents, $D4+15$ cents, $F\#4+3$ cents. The recording procedure was the same as in the previous case.
5) $B\flat$ Major played with a real woodwind quartet: $B\flat 2$, $F3-44$ cents, $B\flat 3-50$ cents, $D5+31$ cents. The notes of the chord were extracted from RWC database [25], carefully pitch-shifted and then combined.
6) C Major played with a real string quartet: $C3-6$ cents, $E3-7$ cents, $C4+30$ cents, $G4-73$ cents. This chord was generated in the same way as the previous one.

- Versions
  (A) Unprocessed chord.
  (B) Processed (developed approach).
  (C) Processed (Melodyne Editor).

In version B, we have used the following parameters for all the sounds: sampling rate $= 44100$ Hz, window size $M = 8001$ samples, FFT size $N = 8192$ samples, number of partials per note $R = 30$ and degree of polyphony $n = 5$. In version C, the degree of polyphony and the notes of the chord have been manually adjusted for each case in order to achieve the best results. In next sections, sounds will be identified by combining the number of the chord and the type of version, i.e. *1.A* would be the first chord in the unprocessed version.

B. Evaluation

1) Subjects: For the evaluation, 31 musicians were interviewed. All of them have passed a minimum of 7 years of formal music education. There were 16 male and 15 female individuals, and most of the subjects' age is below 25. The interviewed musicians play very different instruments (woodwind, piano, percussion…), so there is no predominant instrument.

2) Questionnaires: The subjects were asked to rate from 1 to 10 the perceived consonance of 18 sounds. For every group of three versions (A,B and C), they were also asked to choose, globally, the best version if they had to use such chord in a musical context.

3) Statistics: Different measures have been taken from the questionnaires for each sound in the dataset.

- Mean perceived consonance μ_c.
- Standard deviation of the perceived consonance σ_c.
- Percentage of times that each version has been chosen as the best option among the three versions.

VI. Results & Discussion

The results obtained are shown in Table I. In all cases, the multiple-f_0 estimation stage (with a degree of polyphony set to $n = 5$) perfectly identified the most important f_0s of the chord, and so the target overtone grid was correct. In the chords 1.x, 2.x, 3.x, 4.x and 5.x, the chosen f_S is the tempered chromatic scale (no musical assumptions are made about the input). In 6.x, the note $G4 - 73$ cents could be incorrectly rounded to $F\#4$ because of the large deviation of the partials, so the user must correct f_S with the major scale built upon $\widehat{f_0}_1 = C3$, i.e. C major scale.

TABLE I
Questionnaires Results. **x.A:** Unprocessed Sound; **x.B:** Developed Approach; **x.C:** Melodyne Editor

Chord version	***Perceived consonance [1-10]***	***Chosen as best result***
1.A Original	$\mu_c = 3.48$ $\sigma_c = 1.48$	3.2%
1.B Our approach	$\boldsymbol{\mu_c = 6.64}$ $\boldsymbol{\sigma_c = 2.05}$	**77.4%**
1.C Melodyne	$\mu_c = 5.48$ $\sigma_c = 1.80$	19.35%
2.A Original	$\mu_c = 2.67$ $\sigma_c = 1.30$	6.45%
2.B Our approach	$\boldsymbol{\mu_c = 5.35}$ $\boldsymbol{\sigma_c = 2.25}$	**74.2%**
2.C Melodyne	$\mu_c = 3.96$ $\sigma_c = 1.87$	19.3%
3.A Original	$\mu_c = 4.61$ $\sigma_c = 1.89$	3.2%
3.B Our approach	$\boldsymbol{\mu_c = 7.19}$ $\boldsymbol{\sigma_c = 1.86}$	**83.9%**
3.C Melodyne	$\mu_c = 5.83$ $\sigma_c = 2.35$	9.7%
4.A Original	$\mu_c = 4.32$ $\sigma_c = 1.81$	3.2%
4.B Our approach	$\boldsymbol{\mu_c = 7.09}$ $\boldsymbol{\sigma_c = 1.68}$	**71%**
4.C Melodyne	$\mu_c = 6.19$ $\sigma_c = 1.99$	25.8%
5.A Original	$\mu_c = 2.19$ $\sigma_c = 1.27$	0%
5.B Our approach	$\boldsymbol{\mu_c = 4.03}$ $\boldsymbol{\sigma_c = 2.33}$	**32%**
5.C Melodyne	$\mu_c = 4.64$ $\sigma_c = 2.38$	68%
6.A Original	$\mu_c = 1.54$ $\sigma_c = 0.80$	0%
6.B Our approach	$\boldsymbol{\mu_c = 5.54}$ $\boldsymbol{\sigma_c = 2.15}$	**77.4%**
6.C Melodyne	$\mu_c = 4.77$ $\sigma_c = 1.96$	22.6%

In the case of synthetic sounds (chords 1.x and 2.x) the results show a clear improvement in the consonance of the processed sounds. Unprocessed sounds were strongly perceived as dissonant, whereas the processed ones improved the consonance rating around 3 points. Moreover, the developed approach provides better results than Melodyne Editor for the case of synthetic sounds, since in this case noticeable beating partials are still present in the processed chords. In all comparisons, a *t-Student* test (with $p < 5\%$) revealed statistical validity [26].

The case of the acoustic guitar (chords 3.x and 4.x) is especially interesting, since it is a very common instrument and the results are quite satisfactory. More than 70% of the subjects considered the selected approach to be better than Melodyne Editor. We conclude that plucked string instruments are very appropriate to be processed with the selected approach, since the assumed partial stability holds true for most of the cases.

In the case of a woodwind quartet (5.x), Melodyne performs better than our approach, with a perceived consonance of 4.63 and 4.02 respectively. If both versions are carefully compared, it can be noticed that the difference between them in terms of dissonance is mild, but Melodyne produces a more natural result.

In the case of the strings quartet (6.x) Melodyne does not properly separate the various notes of the chord, so 6.C is still dissonant and unnatural compared to 6.B.

VII. Conclusions

In this paper, a novel method for automatic reduction of dissonance in recorded out-of-tune chords has been proposed. The method shown manipulates the harmonic structure of the input chord as a whole in order to make it fit onto a previously estimated overtone grid. This scheme has applications for professional post-processing tasks of recorded audio. The most prominent commercial tool for polyphonic sound processing is *Melo-*

dyne Editor, which is based on source separation and note-level processing. However, Melodyne is not really suitable for out-of-tune sounds, since it is not able to effectively separate two notes very close in frequency [5].

Additionally, the method proposed reduces amplitude and/or frequency modulations due to unresolved close partials by means of a novel beating reduction algorithm. This algorithm produces a noticeable performance improvement for simple and sustained chords.

The selected approach is based on an *analysis-resynthesis* schema with a *sinusoidal plus residual* model. The system is composed of three stages: *Analysis*, *Harmonic reorganization* and *Synthesis*. In the *Analysis stage*, the sinusoidal and residual components are separated, the partials of the signal are tracked, and the various f_0 comprising the input chord are estimated. In the *Harmonic reorganization stage*, the partials are stabilized and shifted to generate the parameters of the in-tune output chord. Finally, in the *Synthesis stage*, the final waveform is generated using an overlap-add process.

Our approach assumes that the f_0s of the input do not change along time (i.e. it is an isolated chord), the envelope of the signal corresponds to the ADSR model [27] and the notes of the chord are relatively harmonic (i.e. the overtones are placed in multiples of the f_0). Therefore, input chords with vibrato and/or tremolo are not addressed. However, plucked strings instruments, such as the guitar, are very appropriated to be processed with our approach, as demonstrated in Section VI. The achieved results with other type of instruments are varied, but we observed they are quite acceptable as long as the signal is stable in terms of timbre and frequency.

The performance of the system has been evaluated by 31 expert musicians and it has been compared against the performance of the professional reference tools for this task (Melodyne Editor). In the results, the proposed approach shows an important reduction of the inner dissonance of the chords. For most of the cases evaluated, our method provides better results than Melodyne Editor. The most interesting results are found with acoustic guitar recordings, which are almost free of artefacts after processing.

In our future work, we intend to overcome the current limitations of our approach. For instance, the developed system can benefit from one of the existing chord-segmentation methods [28] to deal with sequences of chords. Additionally, the analysis of time-varying sounds (e.g. vibrato or tremolo) can be addressed with predictive time-tracking algorithms (e.g. HMM-based approaches [29]). Moreover, further research is needed to really address the importance of beating reduction in time-varying chords. Furthermore, the system can be easily adapted to process inharmonic sounds if the overtone grid is adapted to the specific inharmonicity of the input notes [30], [31]. Finally, our system can be also adapted to other temperaments, such as Pythagorean or Zarlino [2], if the scale vector $\mathbf{f_s}$ is redefined.

REFERENCES

[1] O. Gurney, "An old Babylonian treatise on the tuning of the harp," *Iraq*, vol. 30, no. 2, pp. 229–233, 1968.

[2] J. Barbour, *Tuning and temperament: a historical survey*. New York, NY, USa: Dover, 2004 [Online]. Available: http://books.google.com/books?id=G-pG77pmlp4C

[3] R. Clark, "Mixing, recording and producing techniques of the pros," Thomson Course Technology PTR, 2006 [Online]. Available: http://bks9.books.google.co.ke/books?id=l4oJAQAAMAAJ

[4] P. Neubäcker, "Sound-object oriented analysis and note-object oriented processing of polyphonic sound recordings," U.S. patent 8,022,286, Sep. 20, 2011.

[5] J. Akin, "Celemony melodyne editor review," *Mix Mag. Profess. Audio and Music Product.*, Feb. 2010 [Online]. Available: http://mixonline.com/gear/reviews/celemony-melodyne-editor-0210

[6] R. Plomp and W. Levelt, "Tonal consonance and critical bandwidth," *J. Acoust. Soc. Amer.*, vol. 38, no. 4, pp. 518–560, 1965.

[7] W. Piston and M. DeVoto, *Harmony*. New York, NY, USA: Norton, 1987 [Online]. Available: http://books.google.com/books?id=MozDQgAACAAJ

[8] N. Cazden, "Sensory theories of musical consonance," *J. Aesthetics Art Criticism*, vol. 20, no. 3, p. 301, 1962.

[9] J. A. Swets, D. M. Green, and W. P. Tanner, "On the width of critical bands," *J. Acoust. Soc. Amer.*, vol. 34, no. 1, p. 108, 1962 [Online]. Available: http://link.aip.org/link/JASMAN/v34/i1/p108/s1&Agg=doi

[10] X. Serra, "A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition," Ph.D. dissertation, Univ. of Stanford, Stanford, CA, USA, 1989.

[11] A. Oppenheim, R. Schafer, and J. Buck, *Discrete-Time Signal Processing*, ser. ser. signal processing series, A. V. Oppenheim, Ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1999, vol. 23 [Online]. Available: http://link.aip.org/link/ELPWAQ/v23/i2/p157/s1&Agg=doi, no. 2

[12] J. Smith, *Spectral Audio Signal Processing, October 2008 Draft*. Stanford, CA, USA: Stanford Univ., Oct., 2013 [Online]. Available: http://ccrma.stanford.edu/~jos/sasp/

[13] M. Abe and J. O. Smith, "Design criteria for the quadratically interpolated FFT method (i): Bias due to interpolation," no. STAN-M-114, 2004 [Online]. Available: https://ccrma.stanford.edu/files/papers/stanm114.pdf

[14] B. Moore, "Frequency difference limens for short-duration tones," *J. Acoust. Soc. Amer.*, vol. 54, no. 3, pp. 610–619, 1973 [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/4754385

[15] C. Yeh, A. Roebel, and X. Rodet, "Multiple fundamental frequency estimation and polyphony inference of polyphonic music signals," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 6, pp. 1116–1126, Aug. 2010.

[16] A. P. Klapuri, "A perceptually motivated multiple-f0 estimation method," in *Proc. IEEE Workshop Applicat. Signal Process. Audio Acoust.*, 2005, pp. 291–294.

[17] A. P. Klapuri, "Multiple fundamental frequency estimation based on harmonicity and spectral smoothness," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 6, pp. 804–816, Nov. 2003.

[18] 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.

[19] M. Slaney, "An efficient implementation of the Patterson-Holdsworth auditory filter bank," Apple Computer, Perception Group, Tech. Rep, 1993.

[20] "Complete MIDI 1.0 Detailed Specification," M. M. Association, 1999/2008 [Online]. Available: http://www.midi.org/techspecs/gm.php

[21] H. Helmholtz, *Die Lehre den Tonempfindungen als physiologische Grundlage fur die Theorie der Musik / von H. Helmholtz*. Braunschweig, Germany: F. Vieweg und Sohn, 1877.

[22] R. Maher, "An approach for the separation of voices in composite musical signals," Ph.D. dissertation, Univ. of Illinois, Urbana, IL, USA, 1989.

[23] K. Jensen, "Envelope model for isolated musical sounds," in *Proc. 2nd COST-G6 Workshop Digital Audio Effects (DAFx99)*, Trondheim, Norway, Dec. 1999, pp. 35–39.

[24] P. Rodet and X. Depalle, "Spectral envelopes and inverse FFT synthesis," *Audio Eng. Soc. Conv. 93*, vol. 10, 1992 [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=6740

[25] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Music genre database and musical instrument sound database," in *Proc. ISMIR*, 2003, vol. 3, pp. 229–230.

[26] S. L. Zabell, "On student's 1908 article the probable error of a mean," *J. Amer. Statist. Assoc.*, vol. 103, no. 481, pp. 1–7, 2008 [Online]. Available: http://amstat.tandfonline.com/doi/abs/10.1198/016214508000000030

[27] G. Torelli and G. Caironi, "New polyphonic sound generator chip with integrated microprocessor-programmable ADSR envelope shaper," *IEEE Trans. Consumer Electron.*, vol. CE-29, no. 3, pp. 203–212, Aug. 1983.
[28] W. De Haas, "Music information retrieval based on tonal harmony," Ph.D. dissertation, Utrecht Univ., Utrecht, The Netherlands, 2012.
[29] P. Depalle, G. Garcia, and X. Rodet, "Tracking of partials for additive sound synthesis using hidden Markov models," in *Proc. ICASSP*, 1993, vol. 1, pp. 225–228 [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=319096
[30] I. Barbancho, L. Tardon, S. Sammartino, and A. Barbancho, "Inharmonicity-based method for the automatic generation of guitar tablature," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 6, pp. 1857–1868, Aug. 2012.
[31] S. Dixon, M. Mauch, and D. Tidhar, "Estimation of harpsichord inharmonicity and temperament from musical recordings," *J. Acoust. Soc. Amer.*, vol. 131, pp. 878–887, 2012.

Emilio Molina received his degrees in Technical Telecommunications Engineering and Telecommunications Engineering from University of Málaga, Spain in 2007 and 2011, respectively. In 2013, he received his MSc in 'Sound and Music Computing' from Universitat Pompeu Fabra, Barcelona, Spain. He obtained the Professional Degree of Classic Piano at Conservatori del Liceu, Barcelona, Spain, in 2012. During his studies, he obtained the Best Final Year Project award from University of Málaga in 2007. He was nominated as finalist for the Best Final Year Project by the Official National Telecommunications Engineering Board in 2013. Currently, he is a PhD candidate at the Application of Information and Communication Technologies Research Group (ATIC) under the supervision of Lorenzo J. Tardón. His main research topic is the automatic analysis and processing of audio signals with special focus on singing voice and its applications.

Ana M. Barbancho received her degree in telecommunications engineering and her Ph.D. degree from University of Málaga, Málaga, Spain, in 2000 and 2006, respectively. In 2001, she also 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, new educational methods, 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.

Lorenzo J. Tardón received his degree in Telecommunications Engineering from University of Valladolid, Valladolid, Spain, in 1995 and his Ph.D. degree from Polytechnic University of Madrid, Madrid, Spain, in 1999. In 1999 he worked for ISDEFE on air traffic control systems at 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, Málaga, Spain. Lorenzo J. Tardón is currently the head of the Application of Information and Communications Technologies (ATIC) Research Group. He has worked as 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 has been 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.

Isabel Barbancho (SM'10]) received her degree in telecommunications engineering and her Ph.D. degree from the University of Málaga (UMA), 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, UMA, as an Assistant and then Associate Professor. During 2013, she has been a Visiting Scholar at University of Victoria, Victoria, BC, Canada. She has been the main researcher in 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.'

B.3

Molina, E., Barbancho, A. M., Tardón, L. J., and Barbancho, I. (2014). Evaluation framework for automatic singing transcription. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 567-572, Taipei (Taiwan).

EVALUATION FRAMEWORK FOR AUTOMATIC SINGING TRANSCRIPTION

Emilio Molina, Ana M. Barbancho, Lorenzo J. Tardón, Isabel Barbancho
Universidad de Málaga, ATIC Research Group, Andalucía Tech,
ETSI Telecomunicación, Campus de Teatinos s/n, 29071 Málaga, SPAIN
emm@ic.uma.es, abp@ic.uma.es, lorenzo@ic.uma.es, ibp@ic.uma.es

ABSTRACT

In this paper, we analyse the evaluation strategies used in previous works on automatic singing transcription, and we present a novel, comprehensive and freely available evaluation framework for automatic singing transcription. This framework consists of a cross-annotated dataset and a set of extended evaluation measures, which are integrated in a Matlab toolbox. The presented evaluation measures are based on standard MIREX note-tracking measures, but they provide extra information about the type of errors made by the singing transcriber. Finally, a practical case of use is presented, in which the evaluation framework has been used to perform a comparison in detail of several state-of-the-art singing transcribers.

1. INTRODUCTION

Singing transcription refers to the automatic conversion of a recorded singing signal into a symbolic representation (e.g. a MIDI file) by applying signal-processing methods [1]. One of its renowned applications is query-by-humming [5], but other types of applications also are related to this task, like singing tutors [2], computer games (e.g. Singstar [1]), etc. In general, singing transcription is considered a specific case of melody transcription (also called note tracking), which is more general problem. However, singing transcription not only relates to melody transcription but also to speech recognition, and still nowadays it is a challenging problem even in the case of monophonic signals without accompaniment [3].

In the literature, various approaches for singing transcription can be found. A simple but commonly referenced approach was proposed by McNab in 1996 [4], and it relied on several handcrafted pitch-based and energy-based segmentation methods. Later, in 2001 Haus et al. used a similar approach with some rules to deal with intonation issues [5], and in 2002, Clarisse et al. [6] contributed with an auditory model, leading to later improved systems such as [7] (later included in MAMI project [2] and today in SampleSumo products [3]). Additionally, other more recent approaches use hidden Markov models (HMM) to detect note-events in singing voice [8, 9, 11]. One of the most representative HMM-based singing transcribers was published by Ryynänen in 2004 [9]. More recently, in 2013, another probabilistic approach for singing transcription has been proposed in [3], also leading to relevant results. Regarding the evaluation methodologies used in these works (see Sections 2.1 and 3.1 for a review), there is not a standard methodology.

In this paper, we present a comprehensive evaluation framework for singing transcription. This framework consists of a cross-annotated dataset (Section 2) and a novel, compact set of evaluation measures (Section 3), which report information about the type of errors made by the singing transcriber. These measures have been integrated in a freely available Matlab toolbox (see Section 3.3). Then, we present a practical case in which the evaluation framework has been used to perform a comparison in detail of several state-of-the-art singing transcribers (Section 4). Finally, some relevant conclusions are presented in Section 5

2. DATASETS

In this section, we review the evaluation datasets used in prior works on singing transcription , and we describe the proposed evaluation dataset and our strategy for ground-truth annotation.

2.1 Datasets used in prior works

In Table 1, we present the datasets used in some relevant works on singing transcription. Note that none of the datasets fully represents the possible contexts in which singing transcription might be applied, since they are either too small (e.g. [5,6]), either very specific in style (e.g. [11] for opera and [3] for flamenco), or either they use an annotation strategy that may be subjective (e.g. [5, 6]), or only valid for very good performances in rhythm and intonation (e.g. [8, 9]). In addition, only the flamenco dataset used in [3] is freely available.

2.2 Proposed dataset

In this section we describe the music collection, as well as the annotation strategy used to build the ground-truth.

[1] http://www.singstar.com

[2] http://www.ipem.ugent.be/MAMI

[3] http://www.samplesumo.com

Author	Year	Dataset size	Audio quality	Music style	Singing style	Ground-truth (GT) annotation strategy	Tunning devs. annotated in GT	Freely available
McNab [4]	1996	NONE						
Haus & Pollastri [5]	2001	20 short melodies	Low & moderated noise	Popular and scales	Syllables: 'na-na'...	Annotated by one musician	No	No
Clarisse et al. [6]	2002	22 short melodies	Low & moderated noise	Popular	Singing with & without lyrics	Annotation by one musician	No	No
Viitaniemi et al. [8]	2003	66 melodies (120 minutes)	High quality (studio conditions)	Folk songs & scales	Singing, humming & whistling	Original score used as ground-truth	No	No
Ryynänen et al. [9]	2004							
Mulder et al. [7]	2004	52 melo. (1354 notes)	Good & moderated noise	Popular songs	Syllables, singing & whistling	Team of musicologists	No	No
Kumar et al. [10]	2007	47 songs (2513 notes)	Good	Indian music	Syllables: /la/ /da/ /na/	Manual annot. of vowel onsets [REf]	No	No
Krige et al. [11]	2008	13842 notes	High quality but strong reverberation	Opera lessons & scales	Syllables	Time alignment using Viterbi	No	No
Gómez & Bonada [3]	2013	72 excerpts (2803 notes)	Good & slightly noisy	Flamenco songs	Lyrics & ornaments	Musicians team (cross-annotation)	Yes	Yes

Table 1. Review of the evaluation datasets used in prior works on singing transcription. Some details about the dataset are not provided in some cases, so certain fields can not be expressed in the same units (e.g. dataset size).

2.2.1 Music collection

The proposed dataset consists of 38 melodies sung by adult and child untrained singers, recorded in mono with a sample rate of 44100Hz 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 [4]): 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 dataset MTG-QBH [5] [12].
- Adult female: 11 pop melodies (281 seconds) sung by 5 different adult female untrained singers, also taken from MTG-QBH dataset.

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

2.2.2 Ground-truth: annotation strategy

The described music collection has been manually annotated to build the ground truth [4] . First, we have transcribed the audio recordings with a baseline algorithm (Section 4.2), and then all the transcription errors have been corrected by an expert musician with more than 10 years of music training. Then, a second expert musician (with 7 years of music training) checked all the annotations until both musicians agreed in their correctness. The transcription errors were corrected by listening, at the same time, to the synthesized transcription and the original audio. The musicians were given a set of instructions about the specific criteria to annotate the singing melody:

- Ornaments such as pitch bending at the beginning of the notes or vibratos are not considered independent notes. This criterion is based on Vocaloid's [6] approach, where ornaments are not modelled with extra notes.
- Portamento between two notes does not produce an extra third note (again, this is the criteria used in Vocaloid).
- 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 pitch of each note is annotated with cents resolution as perceived by the team of experts. Note that we annotate the tuning deviation for each independent note.

3. EVALUATION MEASURES

In this section, we describe the evaluation measures used in prior works on automatic singing transcription, and we present the proposed ones.

3.1 Evaluation measures used in prior works

In Table 2, we review the evaluation measures used in some relevant works on singing transcription. In some cases, only the note and/or frame error is provided as a compact, representative measure [5, 9], whereas other approaches provide extra information about the type of errors made by the system using dynamic time warping (DTW) [6] or Viterbi-based alignment [11]. In our case, we have taken the most relevant aspects of these approaches and we added some novel ideas in order to define a novel, compact and comprehensive set of evaluations.

[4] Available at http://www.atic.uma.es/ismir2014singing

[5] http://mtg.upf.edu/download/datasets/mtg-qbh

[6] http://www.vocaloid.com

Author	Year	Evaluation measures
McNab	1996	NONE
Haus & Pollastri [5]	2001	Rate of note pitch errors (segmentation errors are not considered)
Clarisse et al. [6]	2002	DTW-based measurement of various note errors, e.g. insertions deletions and substitutions.
Viitaniemi et al. [8]	2003	Frame-based errors. Do not report information about type of errors made.
Ryynänen et al. [9]	2004	Note-based and frame-based errors. Do not report information about type of errors made.
Mulder et al. [7]	2004	DTW-based measurement of various note errors, e.g. insertions deletions and substitutions.
Kumar et al. [10]	2007	Onset detection errors (pitch and durations are ignored).
Krige et al. [11]	2008	Viterbi-based measurement of deletions, insertions and substitutions (typical evaluation in speech recognition).
Gómez & Bonada [3]	2013	MIREX measures for audio melody extraction and note-tracking. Do not report information about type of errors made.

Table 2. Evaluation measures used in prior works on singing transcription.

3.2 Proposed measures

In this section, we firstly present the notation and some needed definitions that are used in the rest of sections, and then we describe the evaluation measures used to quantify the proportion of correctly transcribed notes. Finally, we present a set of novel evaluation measures that independently report the importance of each type of error. In Figure 1 we show an example of the types of errors considered.

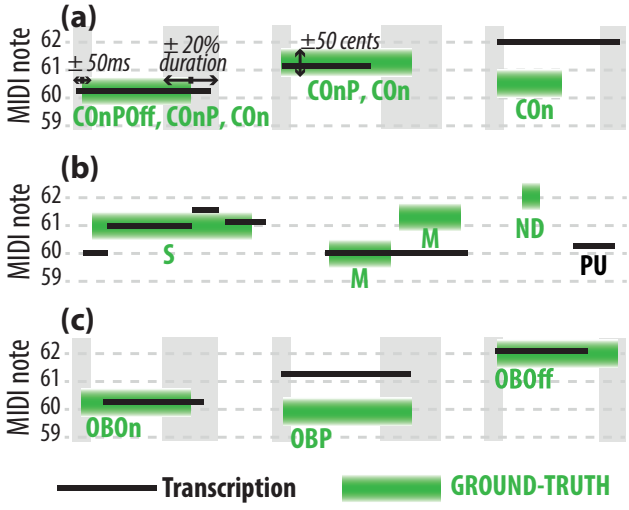

Figure 1. Examples of the different proposed measures.

3.2.1 Notation

The i:th note of the ground-truth is noted as n_i^{GT}, and the j:th note of the transcription is noted as n_j^{TR}. The total number of notes in the ground-truth and the transcription are N^{GT} and N^{TR}, respectively. Regarding the expressions used in the for correct notes, we have used Precision, Recall and F-measure, which are defined as follow:

$$\mathbf{CX}_{\text{Precision}} = \frac{N_{\mathbf{CX}}^{\text{GT}}}{N^{\text{GT}}} \quad (1)$$

$$\mathbf{CX}_{\text{Recall}} = \frac{N_{\mathbf{CX}}^{\text{TR}}}{N^{\text{TR}}} \quad (2)$$

$$\mathbf{CX}_{\text{F-measure}} = 2 \cdot \frac{\mathbf{CX}_{\text{Precision}} \cdot \mathbf{CX}_{\text{Recall}}}{\mathbf{CX}_{\text{Precision}} + \mathbf{CX}_{\text{Recall}}} \quad (3)$$

where $C\mathbf{X}$ makes reference to the specific category of correct note: Correct Onset & Pitch & Offset ($\mathbf{X}$ = COnPOff), Correct Onset & Pitch ($\mathbf{X}$ = COnP) or Correct Onset ($\mathbf{X}$ = COn). Finally, $N_{\mathbf{CX}}^{\text{GT}}$ and $N_{\mathbf{CX}}^{\text{TR}}$ are the total number of matching $C\mathbf{X}$ conditions in the ground-truth and the transcription, respectively.

Regarding the measures used for errors, we have computed the Error Rate with respect to N^{GT}, or with respect to N^{TR}, as follow:

$$\mathbf{X}_{\text{RateGT}} = \frac{N_{\mathbf{X}}^{\text{GT}}}{N^{\text{GT}}} \quad (4)$$

$$\mathbf{X}_{\text{RateTR}} = \frac{N_{\mathbf{X}}^{\text{TR}}}{N^{\text{TR}}} \quad (5)$$

Finally, in the case of segmentation errors (Section 3.2.5), we also compute the mean number of notes tagged as $\mathbf{X}$ in the transcription for each note tagged as $\mathbf{X}$ in the ground-truth. This magnitude has been expressed as a ratio:

$$\mathbf{X}_{\text{Ratio}} = \frac{N_{\mathbf{X}}^{\text{TR}}}{N_{\mathbf{X}}^{\text{GT}}} \quad (6)$$

3.2.2 Definition of correct onset/pitch/offset

The definitions of correctly transcribed notes (given in Section 3.2.3) consists of combinations of three independent conditions: correct onset, correct pitch and correct offset. We have defined these conditions according to MIREX (*Multiple F0 estimation and tracking* and *Audio Onset Detection* tasks), and so they are defined as follow:

• Correct Onset: If the note's onset of a transcribed note n_j^{TR} is within a $\pm$50ms range of the onset of a ground-truth note n_i^{GT}, i.e.:

$$\text{onset}(n_j^{TR}) \in [\text{onset}(n_i^{GT}) - 50\text{ms}, \text{onset}(n_i^{GT}) + 50\text{ms}] \quad (7)$$

then we consider that n_i^{GT} has a correct onset with respect to n_j^{TR}.

• Correct Pitch: If the note's pitch of a transcribed note n_j^{TR} is within a $\pm$0.5 semitones range of the pitch of a ground-truth note n_i^{GT}, i.e.:

$$\text{pitch}(n_j^{TR}) \in [\text{pitch}(n_i^{GT}) - 0.5\text{ st}, \text{pitch}(n_i^{GT}) + 0.5\text{ st}] \quad (8)$$

then we consider that n_i^{GT} has a correct pitch with respect to n_j^{TR}.

• Correct Offset: If the offsets of the ground-truth note n_i^{GT} and the transcribed note n_j^{TR} are within a range of $\pm 20\%$ of the duration of n_i^{GT} or ± 50 ms, whichever is larger, i.e.:

$$\text{offset}(n_j^{TR}) \in [\text{offset}(n_i^{GT}) - \text{OffRan}, \text{offset}(n_i^{GT}) + \text{OffRan}] \quad (9)$$

where $\text{OffRan} = max(50\text{ms}, \text{duration}(n_i^{GT}))$, then we consider that n_i^{GT} has a correct offset with respect to n_j^{TR}.

3.2.3 Correctly transcribed notes

The definition of "correct note" should be useful to measure the suitability of a given singing transcriber for a specific application. However, different applications may require a different definition of correct note. Therefore, we have chosen three different definitions of correct note as defined in MIREX:

• Correct onset, pitch and offset (COnPOff): This is a standard correctness criteria, since it is used in MIREX (*Multiple F0 estimation and tracking* task), and it is the most restrictive one. The note n_i^{GT} is assumed to be correctly transcribed into the note n_j^{TR} if it has correct onset, correct pitch and correct offset (as defined in Section 3.2.2). In addition, one ground truth note n_i^{GT} can only be associated with one transcribed note n_j^{TR}. In our evaluation framework, we report Precision, Recall and F-measure as defined in Section 3.2.1:

$$\text{COnPOff}_{\text{Precision}}, \text{COnPOff}_{\text{Recall}} \text{ and } \text{COnPOff}_{\text{F-measure}}.$$

• Correct Onset, Pitch (COnP): This criteria is also used in MIREX, but it is less restrictive since it just considers onset and pitch, and ignores the offset value. Therefore, in COnP criteria, a note n_i^{GT} is assumed to be correctly transcribed into the note n_j^{TR} if it has correct onset and correct pitch. In addition, one ground truth note n_i^{GT} can only be associated with one transcribed note n_j^{TR}. In our evaluation framework, we report Precision, Recall and F-measure:

$$\text{COnP}_{\text{Precision}}, \text{COnP}_{\text{Recall}} \text{ and } \text{COnP}_{\text{F-measure}}.$$

• Correct Onset (COn): Additionally, we have included the evaluation criteria used in MIREX *Audio Onset Detection* task. In this case, a note n_i^{GT} is assumed to be correctly transcribed into the note n_j^{TR} if it has correct onset. In addition, one ground truth note n_i^{GT} can only be associated with one transcribed note n_j^{TR}. In our evaluation framework, we report Precision, Recall and F-measure:

$$\text{COnPOff}_{\text{Precision}}, \text{COnPOff}_{\text{Recall}} \text{ and } \text{COnPOff}_{\text{F-measure}}.$$

3.2.4 Incorrect notes with one single error

In addition, we have included some novel evaluation measures to identify the notes that are close to be correctly transcribed, but they fail in one single aspect. These measures are useful to identify specific weaknesses of a given singing transcriber. The proposed categories are:

• Only-Bad-Onset (OBOn): A ground-truth note n_i^{GT} is labelled as OBOn if it has been transcribed into a note n_j^{TR} with correct pitch and offset, but wrong onset. In order to detect them, firstly we find all ground-truth notes with correct pitch and offset, taking into account that one ground-truth note can only be associated with one transcribed note. Then, we remove all notes previously tagged as COnPOff (Section 3.2.3). The reported measure is the rate of OBOn notes in the ground-truth:

$$\text{OBOn}_{\text{RateGT}}$$

• Only-Bad-Pitch (OBP): A ground-truth note n_i^{GT} is labelled as OBP if it has been transcribed into a note n_j^{TR} with correct onset and offset, but wrong pitch. In order to detect them, firstly we find all ground-truth notes with correct onset and offset, taking into account that one ground-truth note can only be associated with one transcribed note. Then, we remove all notes previously tagged as COnPOff (Section 3.2.3). The reported measure is the rate of OBP notes in the ground-truth:

$$\text{OBP}_{\text{RateGT}}$$

• Only-Bad-Offset (OBOff): A ground-truth note n_i^{GT} is labelled as OBOn if it has been transcribed into a note n_j^{TR} with correct pitch and onset, but wrong offset. In order to detect them, firstly we find all ground-truth notes with correct pitch and onset, taking into account that one ground-truth note can only be associated with one transcribed note. Then, we remove all notes previously tagged as COnPOff (Section 3.2.3). The reported measure is the rate of OBOff notes in the ground-truth:

$$\text{OBOff}_{\text{RateGT}}$$

3.2.5 Incorrect notes with segmentation errors

Segmentation errors refer to the case in which sung notes are incorrectly split or merged during the transcription. Depending on the final application, certain types of segmentation errors may not be important (e.g. frame-based systems for query-by-humming are not affected by splits), but they can lead to problems in many other situations. Therefore, we have defined two evaluation measures which are informative about the segmentation errors made by the singing transcriber.

• Split (S): A split note is a ground truth note n_i^{GT} that is incorrectly segmented into different consecutive notes $n_{j_1}^{TR}$, $n_{j_2}^{TR} \cdots n_{j_n}^{TR}$. Two requirements are needed in a split: (1) the set of transcribed notes $n_{j_1}^{TR}, n_{j_2}^{TR}, \ldots n_{j_n}^{TR}$ must overlap at least the 40% of n_i^{GT} in time (pitch is ignored), and (2) n_i^{GT} must overlap at least the 40% of every note $n_{j_1}^{TR}, n_{j_2}^{TR}, \ldots n_{j_n}^{TR}$ in time (again, pitch is ignored). These requirements are needed to ensure a consistent relationship between ground truth and transcribed notes. The specific reported measures are:

$$\text{S}_{\text{RateGT}} \text{ and } \text{S}_{\text{Ratio}}$$

Note that in this case $\text{S}_{\text{Ratio}} > 1$.

• Merged (M): A set of consecutive ground-truth notes $n_{i_1}^{GT}$, $n_{i_2}^{GT}$, $\cdots$ $n_{i_n}^{GT}$ are considered to be merged if they all are transcribed into the same note n_j^{TR}. This is the complementary case of split. Again, two requirements must be true to consider a group of merged notes: (1) the set of ground truth notes $n_{i_1}^{GT}$,$n_{i_2}^{GT}$, $\ldots$ $n_{i_n}^{GT}$ must overlap the 40% of n_j^{TR} in time (pitch is ignored), and (2) n_j^{TR} must overlap the 40% of every note $n_{i_1}^{GT}$,$n_{i_2}^{GT}$, $\ldots$ $n_{i_n}^{GT}$ in time (again, pitch is ignored). The specific reported measures are:

$$\text{M}_{\text{RateGT}} \text{ and } \text{M}_{\text{Ratio}}$$

Note that in this case $\text{M}_{\text{Ratio}} < 1$.

3.2.6 Incorrect notes with voicing errors

Voicing errors happen when an unvoiced sound produces a false transcribed note (spurious note), or when a sung note is not transcribed at all (non-detected note). This situation is commonly associated to a bad performance of the voicing stage within the singing transcriber. We have defined two categories:

• Spurious notes (PU): A spurious note is a transcribed note n_j^{TR} that does not overlap at all (neither in time nor in pitch) any note in the ground truth. The associated reported measure is:

$$\mathrm{PU}_{\mathrm{RateTR}}$$

• Non-detected notes (ND): A ground-truth note n_i^{GT} is non-detected if it does not overlap at all (neither in time nor in pitch) any transcribed note. The associated reported measure is:

$$\mathrm{ND}_{\mathrm{RateGT}}$$

3.3 Proposed Matlab toolbox

The presented evaluation measures have been implemented in a freely available Matlab toolbox [4], which consists of a set of functions and structures, as well as a graphical user interface to visually analyse the performance of the evaluated singing transcriber.

The main function of our toolbox is `evaluation.m`, which receives the ground-truth and the transcription of an audio clip as inputs, and it outputs the results of all the evaluation measures. In addition, we have included a function called `listnotes.m`, which receives as inputs the ground-truth, the transcription and the category **X** to be listed, and it outputs a list (in a two-columns format: onset time-offset time) of all the notes in the ground-truth tagged as **X** category. This information is useful to isolate the problematic audio excerpts for further analysis.

Finally, we have implemented a graphical user interface, where the ground-truth and the transcription of a given audio clip can be compared using a piano-roll representation. This interface also allows the user to highlight notes tagged as **X** (e.g. COnPOff, S, etc.).

4. PRACTICAL USE OF THE PROPOSED TOOLBOX

In this section, we describe a practical case of use in which the presented evaluation framework has been used to perform an improved comparative study of several state-of-the-art singing transcribers (presented in Section 4.1). In addition, a simple, easily reproducible baseline approach has been included in this comparative study. Finally, we show and discuss the obtained results.

4.1 Compared algorithms

We have compared three state-of-the-art algorithms for singing transcription:

Method (a): Gómez & Bonada (2013) [3]. It consists of three main steps: tuning-frequency estimation, transcription into short notes, and an iterative process involving note consolidation and refinement of the tuning frequency. For the experiment, we have used a binary provided by the authors of the algorithm.

Method (b): Ryynänen (2008) [13]. We have used the method for automatic transcription of melody, bass line and chords in polyphonic music published by Ryynänen in 2008 [13], although we only focus on melody transcription. It is the last evolution of the original HMM-based monophonic singing transcriber [9]. For the experiment, we have used a binary provided by the authors of the algorithm.

Method (c): Melotranscript [4] (based on Mulder 2004 [7]). It is the commercial version derived from the research carried out by Mulder et al. [7]. It is based on an auditory model. For the experiment, we have used the demo version available in SampleSumo website [3].

4.2 Baseline algorithm

According to [8], 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. The proposed baseline method is based on such idea, and it uses Yin [14] to extract the F0 and aperiodicity at frame-level. A frame is classified as unvoiced if its aperiodicity is under < 0.4. Finally, all notes shorter than 100ms are discarded.

4.3 Results & discussion

In Figure 2 we show the results of our comparative analysis. Regarding the F-measure of correct notes (COnPOff, COnP and COn), methods (a) and (c) attains similar values, whereas method (b) performs slightly worse. In addition, it seems that method (a) is slightly superior to method (c) for onset detection, but method (c) is superior when pitch and offset values must be also estimated. In all cases, the baseline is clearly worse than the rest of methods.

In addition, we observed that the rate of notes with incorrect onset (OBOn) is equally high (20%) in all methods. After analysing the specific recordings, we concluded that onset detection within a range of ± 50ms is very restrictive in the case of singing voice with lyrics, since many onsets are not clear even for an expert musician (as proved during the ground-truth building). Moreover, we also observed that all methods, and especially method (a), have problems with pitch bendings at the beginning of the notes, since they tend to split them.

Regarding the segmentation and voicing errors, we realised that method (a) tends to split notes, whereas method (b) tends to merge notes. This information, easily provided by our evaluation framework, may be useful to improve specific weaknesses of the algorithms during the development stage. Finally, we also realised that method (b) is worse than method (a) and (c) in terms of voicing.

To sum up, method (c) seems to be the best one in most measures, mainly due to a better performance in segmentation and voicing. However, method (a) is very appropriate for onset detection. Finally, although method (b) works clearly better than the baseline, has a poor performance due to errors in segmentation (mainly merged notes) and voicing (mainly spurious).

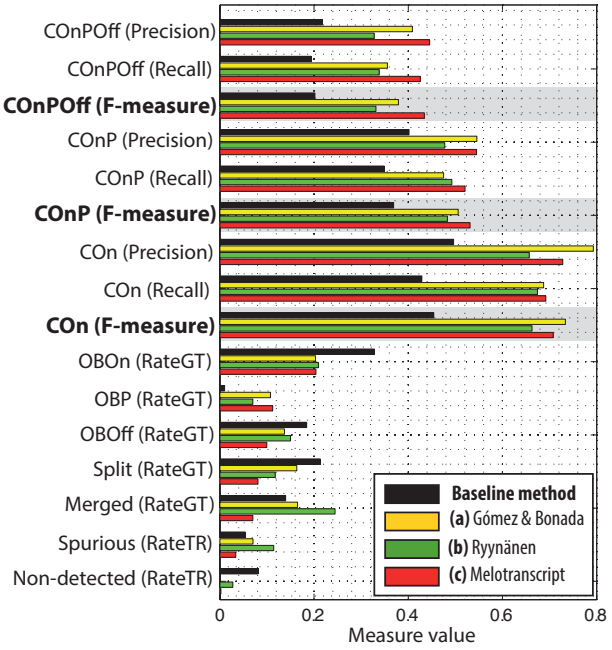

Figure 2. Comparison in detail of several state-of-the-art singing transcription systems using the presented evaluation framework.

5. CONCLUSIONS

In this paper, we have presented an evaluation framework for singing transcription. It consists of a cross-annotated dataset of 1154 seconds and a novel set of evaluation measures, able to report the type of errors made by the system. Both the dataset, and a Matlab toolbox including the presented evaluation measures, are freely available [4]. In order to show the utility of the work presented in this paper, we have performed an detailed comparative study of three state-of-the-art singing transcribers plus a baseline method, leading to relevant information about the performance of each method. In the future, we plan to expand our evaluation dataset in order to make it comparable to other datasets [7] used in MIREX (e.g. MIR-1K or MIR-QBSH).

6. ACKNOWLEDGEMENTS

This work has been funded by the Ministerio de Economía y Competitividad of the Spanish Government under Project No. TIN2013-47276-C6-2-R and by the Junta de Andalucía under Project No. P11-TIC-7154. The work has been done at Universidad de Málaga. Campus de Excelencia Internacional Andalucía Tech.

7. REFERENCES

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

[2] E. Molina, I. Barbancho, E. Gómez, A. M. Barbancho, and L. J. Tardón, "Fundamental frequency alignment vs. note-based melodic similarity for singing voice assessment," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP*, pp. 744–748, 2013.

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

[4] R. J. McNab, L. A. Smith, and I. H. Witten, "Signal Processing for Melody Transcription," *Proceedings of the 19th Australasian Computer Science Conference*, vol. 18, no. 4, pp. 301–307, 1996.

[5] G. Haus and E. Pollastri, "An audio front end for query-by-humming systems," in *Proceedings of the 2nd International Symposium on Music Information Retrieval ISMIR*, pp. 65–72, sn, 2001.

[6] 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 *Proceedings of the 3rd International Conference on Music Information Retrieval ISMIR*, pp. 116–123, 2002.

[7] T. De Mulder, J.P. Martens, M. Lesaffre, M. Leman, B. De Baets, H. De Meyer, "Recent improvements of an auditory model based front-end for the transcription of vocal queries", , *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP 2004)*, Montreal, Quebec, Canada, May 17–21, Vol. IV, pp. 257–260, 2004.

[8] T. Viitaniemi, A. Klapuri, and A. Eronen, "A probabilistic model for the transcription of single-voice melodies," in *Proceedings of the 2003 Finnish Signal Processing Symposium FINSIG03*, pp. 59–63, 2003.

[9] M. Ryynänen and A. Klapuri, "Modelling of note events for singing transcription," in *Proceedings of ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing SAPA*, (Jeju, Korea), Oct. 2004.

[10] P. Kumar, M. Joshi, S. Hariharan, and P. Rao, "Sung Note Segmentation for a Query-by-Humming System". In *Intl Joint Conferences on Artificial Intelligence (IJCAI)*, 2007.

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

[12] J. Salamon, J. Serrá and E. Gómez, "Tonal Representations for Music Retrieval: From Version Identification to Query-by-Humming", *International Journal of Multimedia Information Retrieval, special issue on Hybrid Music Information Retrieval*, vol. 2, no. 1, pp. 45–58, 2013.

[13] M. P. Ryynänen and A. P. Klapuri, "Automatic Transcription of Melody, Bass Line, and Chords in Polyphonic Music," in *Computer Music Journal*, vol.32, no. 3, 2008.

[14] A. De Cheveigné and H. Kawahara: "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustic Society of America*, Vol. 111, No. 4, pp. 1917-1930, 2002.

[7] http://mirlab.org/dataSet/public/

B.4

Molina, E., Barbancho, I., Barbancho, A. M., and Tardón, L. J. (2014). Parametric model of spectral envelope to synthesize realistic intensity variations in singing voice. In *Proceedings of the 39th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, pages 634-638, Florence (Italy).

PARAMETRIC MODEL OF SPECTRAL ENVELOPE TO SYNTHESIZE REALISTIC INTENSITY VARIATIONS IN SINGING VOICE

Emilio Molina, Isabel Barbancho, Ana M. Barbancho, Lorenzo J. Tardón

Universidad de Málaga, Andalucía Tech, ATIC Research Group,
ETSI Telecomunicación, Campus de Teatinos s/n, 29071 Málaga, SPAIN
emm@ic.uma.es, ibp@ic.uma.es, abp@ic.uma.es, lorenzo@ic.uma.es

ABSTRACT

In this paper, we propose a method to synthesize the natural variations of spectral envelope as intensity varies in singing voice. To this end, we propose a parametric model of spectral envelope based on novel 4-pole resonators as formant filters. This model has been used to analyse 60 vowels sung at different intensities in order to define a set of functions describing the global variations of parameters along intensity. These functions have been used to modify the intensity of 16 recorded vowels and 8 synthetic vowels generated with Vocaloid. The realism of the transformations performed with our approach has been evaluated by four amateur musicians in comparison to Melodyne for real sounds and to Vocaloid for synthetic sounds. The proposed approach has been proved to achieve more realistic sounds than Melodyne and Vocaloid, especially for loud-to-weak transformations.

Index Terms— Human voice, Acoustic signal processing, Speech Synthesis

1. INTRODUCTION

In recent years, the development of software to process and/or synthesize singing voice with creative purposes has become trendy [1, 2, 3]. In this context, a commonly addressed problem is related to the definition of singing processing algorithms (e.g. pitch shifting) able to preserve the spectral envelope of the original sound in order to avoid uncontrolled changes of timbre [4]. However, the spectral envelope has been proved to vary in different contexts of intensity and pitch [5, 6]. This fact motivated the development of the work presented in this paper. Indeed, timbre variations along F0 or intensity are commonly annotated by phoniatricians with different colors in the phonetogram [7] (a graph that displays the dynamic range of a given singer in terms of fundamental frequency and intensity).

This work has been funded by the Ministerio de Economía y Competitividad of the Spanish Government under Project No. TIN2010-21089-C03-02 and Project No. IPT-2011-0885-430000, by the Junta de Andalucía under Project No. P11-TIC-7154 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 work has been done in the context of Campus de Excelencia Internacional Andalucía Tech, Universidad de Málaga.

Prior research on singing processing/synthesis generally focus on artifact reduction [8], formant preservation [4][9], realistic f_0 evolution [10][11], etc. since these aspects are important to achieve a natural result. However, to the best of our knowledge, these references do not address the natural changes of the spectral envelope along intensity or pitch, and we consider that this gap limits the naturalness of the processed sounds.

In this paper, we address the modelling and synthesizing schemes of the natural variations of the spectral envelope as the intensity of the singing voice varies. We propose a parametric model of the spectral envelope, which has been implemented in a freely available software tool able to analyse and process recorded vowels (Section 2). Making use of this tool, we have analysed a set of 60 sung vowels at different intensities and we have defined a set of functions that describe the variations of parameters along intensity (Section 3). We have used these functions to modify the intensity of 24 sung notes and we have evaluated the naturalness of the processed sounds though a subjective analysis (Section 4). The results of this evaluation are presented in Section 5. Some conclusions about our study are given in Section 6.

2. PROPOSED PARAMETRIC MODEL OF SPECTRAL ENVELOPE

In this section, we describe a novel parametric model of the spectral envelope (Section 2.1), which is used to parametrize voiced sounds. Additionally, in Section 2.2 we describe the used procedure to estimate the parameters from real sounds.

2.1. Proposed parametric model of spectral envelope

Regarding the existing approaches to parameterize the spectral envelope of speech, Linear Prediction Coding (LPC) is one the most used techniques [12]. LPC efficiently fits the input signal using a N-order all-pole filter, which is appropriate to model the vocal tract acoustic response [13]. However, the optimal order of the filter is hard to obtain, and LPC technique contains systematic errors that are specially manifested in high-pitched signals [14]. Moreover, LPC coefficients are hard to manipulate to perform timbre transformations. Later related approaches, like Line Spectral Frequencies (LSF) [15]

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

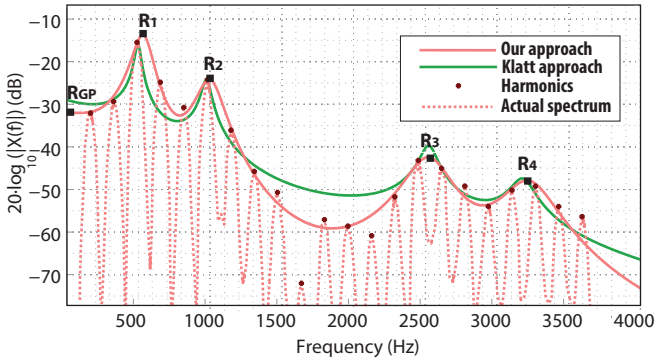

Fig. 1. Comparison between our approach (4-poles resonators) and Klatt's approach (2-poles resonators) to fit the harmonic spectrum of a */o/* vowel sung by a male singer.

or cepstrum-based techniques [14], partially improve some of these drawbacks. However, the parameters of these models are hard to manipulate in order to obtain natural timbre variations, since they are not directly related to physical aspects of the human vocal production. Therefore, we have discarded LPC, LSF or cepstrum-based techniques and we have focused on formant-based models, whose parameters are harder to obtain, but easier to manipulate to synthesize natural timbre variations.

The finally chosen approach is a formant-based parametric model of the spectral envelope, inspired by previous systems for speech/singing synthesis like [16] or [17]. This type of model consider that the spectral envelope of speech signals can be synthesized with several resonator filters in parallel (equivalent to the acoustic formants) with a certain overall slope (determined by the glottal source).

In our approach, we model the spectrum of sung vowels in the frequency band [0, 5000 Hz] with a source filter (as mentioned in [17], determining the overall slope of the spectrum) in cascade with a set of five parallel resonators (the glottal resonator R_{GP} + the first four formants of the vocal tract: R_1, R_2, R_3 and R_4). Therefore, the final envelope $|E(f)|_{dB}$ is defined by the following expression:

$$|E(f)|_{\text{dB}} = \text{Source}(f)_{\text{dB}} + \text{Resonances}(f)_{\text{dB}} \quad (1)$$

where:

$$\text{Source}(f)_{\text{dB}} = \text{Gain}_{\text{dB}} + \text{SlopeDepth}_{\text{dB}} \left(e^{\alpha \cdot f} - 1\right) \quad (2)$$

$$\text{Resonances}(f)_{\text{dB}} = 20 \log_{10} \left(|R_{GP}(f)| + \sum_{i=1}^{4} |R_i(f)| \right) \quad (3)$$

and where the constant $\alpha = -4 \cdot 10^{-4}$, and $R_i(f)$ is the frequency response of resonator i. In the literature, vocal formants are typically modelled with 2-poles resonators, which is the approach of well-known Klatt's speech synthesizer[16]. However we propose the use of 4-poles resonators, which proved to obtain better results at modelling the voiced sounds of our dataset. In Figure 1, we show an example in which our approach fits a natural spectral envelope more accurately than Klatt's approach. Specifically, the proposed 4-poles resonator consists of two identical 2nd-order filters in cascade, with poles $p_1 = p_2 = \rho \cdot e^{\theta}$ and $p_3 = p_4 = \rho \cdot e^{-\theta}$, defined by the following equation:

$$R_x(z) = \frac{K \cdot (1 - 2 \cdot \rho \cdot cos(\theta) + \rho^2)}{(1 - 2 \cdot \rho \cdot cos(\theta) z^{-1} + \rho^2 z^{-2})^2 \, (1 - \rho)^2} \quad (4)$$

$$\text{being: } K = 10^{-4} \quad \rho = e^{-\pi B_x / f_s} \quad \theta = \frac{2 \cdot \pi \cdot f_x}{f_s} \quad (5)$$

where f_x is the central frequency in Hz of resonator x, B_x is the 6 dB-bandwidth in Hz (it should not be confused with the typical 3 dB-bandwidth) and f_s is the sampling rate in Hz. In Table 1 we provide all the specific values of our model (based on [16]) to make our experiments reproducible.

Parameter	Range	Parameter	Range
Gain_{dB}	$[-200, 0]$ dB	f_2	$[500, 3000]$ Hz
$\text{SlopeDepth}_{\text{dB}}$	$[-50, 100]$ dB	B_2	$[40, 1000]$ Hz
f_{GP}	$[0, 600]$ Hz	f_3	$[500, 3000]$ Hz
B_{GP}	$[100, 2000]$ Hz	B_3	$[40, 1000]$ Hz
f_1	$[150, 1100]$ Hz	f_4	$[3000, 5000]$ Hz
B_1	$[40, 1000]$ Hz	B_4	$[100, 1000]$ Hz

Table 1. Range of values used to fit the spectrum of real sounds with the proposed model.

2.2. Estimation of parameters

The automatic estimation of formant frequencies and bandwidths from recorded audio is not straightforward, and such is one of the drawbacks of the proposed model. In the literature we can find many algorithms for formants estimation [18, 19], some of which are included in software tools like Praat [20]. However, we realized that they are not free of errors and sometimes may require human intervention for a reliable annotation of sounds [21]. Due to this, we have manually corrected the parameters obtained with Praat in order to accurately fit the target envelope. The harmonic and the residual component have been separated using the algorithms described in [22, 23] in order to analyse them independently .

3. ANALYSIS OF THE VARIATION OF PARAMETERS ALONG INTENSITY

We have analysed a dataset of sung notes (described in Section 3.1) in order to model the variation of parameters along intensity for both the harmonic (Section 3.2) and the residual component (Section 3.3). Then, we use this information to create a model that emulates the natural changes of spectral envelope for different degrees of intensity (Section 3.4).

3.1. Analysis dataset

We have annotated a total of 60 sustained notes sung by two male and two female singers with ages between 20 and 40

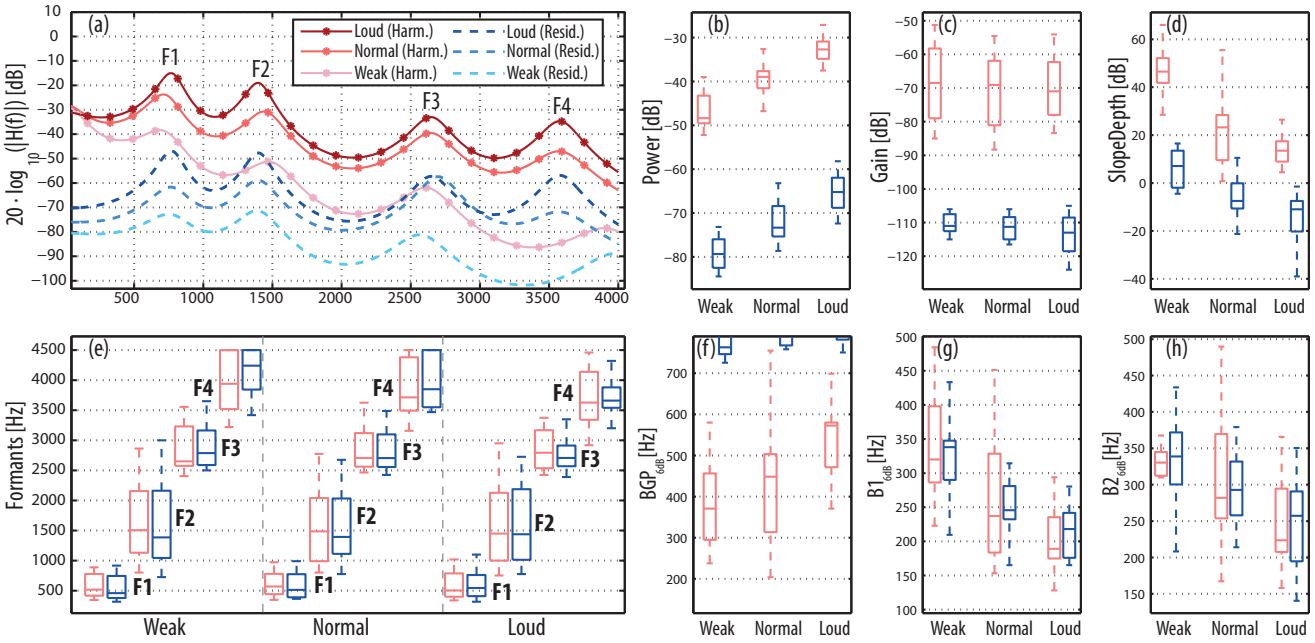

Fig. 2. Information about the harmonic component (light red color) and the residual component (dark blue color) at different degrees of intensity **(2.a)** Spectral envelope of an */a/* vowel sung by a male singer **(2.b)** Average power **(2.c)** Average Gain **(2.d)** Average SlopeDepth **(2.e)** Average frequency values of the first four formants **(2.f)** Average bandwidths of the glottal resonator R_{GP} **(2.g)** Average bandwidths of the first formant R_1 **(2.h)** Average bandwidths of the second formant R_2

years old. All of them have some experience as singers in amateur pop bands, but they do not have academic vocal training. The 60 sung notes correspond to 5 different sustained vowels (/a/, /e/, /i/, /o/ and /u/), in 3 different intended intensities (weak, normal, loud) for 4 different singers. All the notes were sung in a comfortable pitch register for all singers. The recordings were made in a semi-anechoic chamber with a microphone Neumann TLM103, a pop shield and a Onyx-Blackbird firewire interface with a sample rate of 11025 Hz (since we are modelling the frequency band [0,5000] Hz). Each singer was told to keep the same distance to the microphone for all the recordings.

3.2. Analysis for the harmonic envelope

In the analysis, the parameters were averaged over all the notes of the dataset in order to study their global tendency along intensity (see Fig. 2). According to the results, the parameter most noticeably affected by the changes of intensity is the slope depth (Figure 2.d). Indeed, vocal effort has been proved to affect the slope of the source spectrum in prior studies [24]. Moreover, we observed that the bandwidths of the various formants (Figures 2.f, 2.g and 2.h) are reduced as the intensity increases. This phenomenon, which is clearly manifested in our experiments, is coherent with the nonlinear energy damping model of vocal resonators proposed in [25], in which the Q of the filters depends on the input signal. Surprisingly, the gain is not strongly affected by the degree of intensity (2.c), since the measured variations of power (2.b) are mainly due to more prominent formants (decrease of bandwidths) and an increase of high frequencies (decrease of SlopeDepth). Additionally, the shifting of formants frequencies along intensity is slight, and there is not such a clear pattern.

3.3. Analysis for the residual component

The residual and the harmonic component behaves in a similar way as the intensity increases: the slope depth and the bandwidth of the formants decrease, the gain remains rather stable and the formants frequencies are not strongly modified. Additionally, the ratio between harmonic and residual power remains surprisingly stable for different intensity levels (when the whole bandwidth is considered). However, we have observed that such ratio is lower at high frequencies, since the slope depth is generally higher for the harmonic component (this effect is especially noticeable in weak notes). Additionally, we observed that the residual component of loud notes sounds rather "creaky", whereas weak notes has a more breathy texture.

3.4. Proposed model to modify the intensity of the notes

We have defined the *variation of intensity* ΔI as the desired change of intensity to produce. A positive variation produces an increase of intensity, and a negative variation a decrease. The parameter has been normalized so that a step $\Delta I = \pm 10$ produces a complete change from weak to loud, or viceversa.

Typically, $\Delta I = \pm 1$ is a reasonable unit to gradually increase or decrease the intensity of the signal.

Each parameter is modified according to the following expression:

$$p'_x = p_x + \Delta I \cdot w_x \quad (6)$$

where p'_x is the new value of parameter x, p_x is the old value of parameter x, and w_x is the specific weight of paramter x. Additionally, p'_x must always be limited to the range presented in Table 1. In Table 2, we show the specific weights w_x associated to all the parameters for both the residual and the harmonic component. These values have been obtained through linear regression on the analysis dataset described in Section 3.1.

$\mathbf{p_x}$	$\mathbf{w_x}$ (harm.)	$\mathbf{w_x}$ (res.)	$\mathbf{p_x}$	$\mathbf{w_x}$(harm.)	$\mathbf{w_x}$(res.)
Gain_{dB}	0.00 dB	−0.30 dB	F_{R2}	0 dB	0.95 Hz
$\text{SlopeDepth}_{\text{dB}}$	−3.00 dB	−2.04 Hz	B_{R2}	−5.20 dB	−8.26 Hz
F_{R0}	−8.15 Hz	2.33 Hz	F_{R3}	−1.70 Hz	−14.16 Hz
B_{R0}	15.50 Hz	−9.59 Hz	B_{R3}	−2.25 Hz	−8.53 Hz
F_{R1}	0 Hz	5.83 Hz	F_{R4}	−21.76 Hz	−42.16 Hz
B_{R1}	−8.00 Hz	−10.91 Hz	B_{R4}	−2.31 Hz	−9.28 Hz

Table 2. Proposed weights of parameters to modify the perceived intensity of sung notes

4. EVALUATION

In this section, we describe the dataset of sung vowels used for the evaluation (Section 4.1), as well as the used evaluation methodology (Section 4.2).

4.1. Evaluation dataset

We have collected 12 pairs of weak-loud sung vowels in mono audio with a sample rate of 11025 Hz: 4 weak-loud pairs sung by two singers (male M1 and female F1) taken from the analysis dataset (Section 3.1), 4 sung by two singers (male M2 and female F2) not analysed before, and 4 pairs synthesized with "Bruno" (VM) and "Clara" (VF) singers in Vocaloid 3.0. Each singer (either real or synthetic) has sung a weak-loud pair using both an open vowel (*/a/*) and a closed vowel (*/i/*)) in a comfortable register.

4.2. Evaluation methodology

In the case of natural vowels, we have compared our approach ($\Delta I = \pm 10$) with Melodyne Editor (state-of-the-art commercial software). In the case of synthetic vowels, we have compared our approach ($\Delta I = \pm 10$) with Vocaloid 3.0 by setting the parameter *Dynamics* to 127 (loud vowels) and 32 (weak vowels). It makes a total of 48 pairs of weal-loud or loud-weak changes[1]. The evaluation has been performed by four amateur musicians, who listened (with high quality headphones) the different systems in random order, and they were asked to evaluate how close to a real change of intensity was the applied processing.

5. RESULTS

In Figure 3 we show the perceived closeness to a real change of intensity for each of the 48 pairs described in Section 4.2.

[1] Available at: http://www.atic.uma.es/icassp2014singing

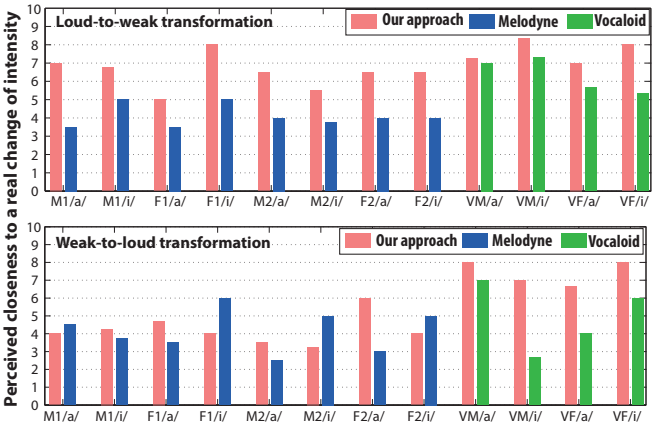

Fig. 3. Mean perceived closeness to a real change of intensity. Each specific combination of singer/vowel (see Section 4.2) has been evaluated with various approaches, represented with different colours.

In general, our approach achieves better results for loud-to-weak transformations, whereas in the case of weak-to-loud transformations, the results are less realistic. Indeed, we have observed that formants are less defined in weak sounds (see example in Figure 2.a), and therefore they are harder to analyse and manipulate. Regarding the results with synthetic vowels, our approach achieves more realism than Vocaloid at modifying the intensity for all cases.

6. CONCLUSIONS

In this paper, we have presented a novel parametric model of spectral envelope to produce realistic variations of intensity in recorded or synthetic vowels. Our model is inspired by previous approaches like [16] or [26], but we introduce some improvements to fit more accurately the spectral envelope of real sounds. Specifically, we propose the use of 4-poles resonators to synthesize the vocal formants, instead of 2-poles resonators. Using our parametric model, a set of 60 sung vowels (natural and synthesized with Vocaloid 3.0) at different intensities have been analysed with Praat (combined with manual annotation) in order to define a set of functions describing the variation of parameters along intensity in singing voice. These functions have been applied to 16 recorded and 8 synthetic vowels (generated with Vocaloid 3.0) to modify their intensity. We have also modified the intensity of the 16 natural vowels with Melodyne Editor, and of the 8 synthetic ones with Vocaloid. The realism of the transformations has been evaluated by four amateur musicians through a survey. The results showed that our approach is especially good when dealing with synthetic vowels, but it also performs well in loud-to-weak transformations with real sounds. In the future, we plan to apply our approach to model the natural changes of spectral envelope along pitch, since it could contribute to extend the idea of *envelope preservation* [4], which is recurrent in many state-of-the-art pitch shifting algorithms.

7. REFERENCES

[1] "Celemony Software: Melodyne Editor," Official website: //http://www.celemony.com.

[2] H. Kenmochi and H. Ohshita, "VOCALOID-commercial singing synthesizer based on sample concatenation.," in *INTERSPEECH*, 2007, pp. 4009–4010.

[3] C. Roig, I. Barbancho, E. Molina, L. J. Tardón, and A. M. Barbancho, "Rumbator: A flamenco rumba cover version generator based on audio processing at note-level," in *DAFx-13*, 2013.

[4] A. Röbel and X. Rodet, "Efficient spectral envelope estimation and its application to pitch shifting and envelope preservation," in *Proc. DAFx*, 2005.

[5] J. E. Huber, E. T. Stathopoulos, G. M. Curione, T. A. Ash, and K. Johnson, "Formants of children, women, and men: The effects of vocal intensity variation," *The Journal of the Acoustical Society of America*, vol. 106, pp. 1532, 1999.

[6] K. Chládková, P. Boersma, and V. J. Podlipskỳ, "On-line formant shifting as a function of F0.," in *INTERSPEECH*, 2009, pp. 464–467.

[7] P. Pabon, "Manual of Voice Profiler Version 4.0," 2010, http://kc.koncon.nl/staff/pabon.

[8] J. Laroche, "Frequency-domain techniques for high quality voice modification," *Proc. of DAFx-03*, pp. 328–322, 2003.

[9] D. Arfib, F. Keiler, U. Zölzer and V. Verfaille "DAFX: Digital Audio Effects, 2nd Edition", p.279-320, Willey, Chichester, UK, 2011.

[10] M. Saitou, T. Unoki and M. Akagi, "Development of an F0 control model based on F0 dynamic characteristics for singing-voice synthesis," *Speech communication*, vol. 46, no. 3, pp. 405–417, 2005.

[11] Y. Ohishi, H. Kameoka, D. Mochihashi, K. Kashino, "A Stochastic Model of Singing Voice F0 Contours for Characterizing Expressive Dynamic Components," *INTERSPEECH*, 2012.

[12] D. O'Shaughnessy, "Linear predictive coding," *Potentials, IEEE*, vol. 7, no. 1, pp. 29–32, 1988.

[13] B. Atal and J. Remde, "A new model of LPC excitation for producing natural-sounding speech at low bit rates," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'82*. IEEE, 1982, vol. 7, pp. 614–617.

[14] Axel Röbel, Fernando Villavicencio, and Xavier Rodet, "On cepstral and all-pole based spectral envelope modeling with unknown model order," *Pattern Recognition Letters*, vol. 28, no. 11, pp. 1343–1350, 2007.

[15] P. Kabal and R. P. Ramachandran, "The computation of line spectral frequencies using Chebyshev polynomials," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 34, no. 6, pp. 1419–1426, 1986.

[16] D. H. Klatt, "Software for a cascade/parallel formant synthesizer," *the Journal of the Acoustical Society of America*, vol. 67, pp. 971, 1980.

[17] J. Bonada, O. Celma, A. Loscos, J. Ortolà, X. Serra, Y. Yoshioka, H. Kayama, Y. Hisaminato, and H. Kenmochi, "Singing voice synthesis combining excitation plus resonance and sinusoidal plus residual models," in *Proceedings of International Computer Music Conference*. Citeseer, 2001.

[18] R. C. Snell and F. Milinazzo, "Formant location from LPC analysis data," *Speech and Audio Processing, IEEE Transactions on*, vol. 1, no. 2, pp. 129–134, 1993.

[19] C. Glaser, M. Heckmann, F. Joublin, and C. Goerick, "Combining auditory preprocessing and Bayesian estimation for robust formant tracking," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 2, pp. 224–236, 2010.

[20] P. Boersma and D. Weenink, "Praat: doing phonetics by computer (version 5.1. 05)[computer program]," *online: http://www. praat. org*, 2009.

[21] S. A. Fulop, "Accuracy of formant measurement for synthesized vowels using the reassigned spectrogram and comparison with linear prediction," *The Journal of the Acoustical Society of America*, vol. 127, pp. 2114, 2010.

[22] J. Bonada, X. Serra, X. Amatriain, and A. Loscos, "Spectral processing," *DAFX Digital Audio Effects*, pp. 393–444, 2011.

[23] E. Molina, A. Barbancho, L. Tardon, and I. Barbancho, "Dissonance reduction in polyphonic audio using harmonic reorganization," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 2013.

[24] J. Sundberg and M. Nordenberg, "Effects of vocal loudness variation on spectrum balance as reflected by the alpha measure of long-term-average spectra of speech," *The Journal of the Acoustical Society of America*, vol. 120, pp. 453, 2006.

[25] H. Ohmura and K. Tanaka, "Speech synthesis using a nonlinear energy damping model for the vocal folds vibration effect," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*. IEEE, 1996, vol. 2, pp. 1241–1244.

[26] J. Bonada, "High quality voice transformations based on modeling radiated voice pulses in frequency domain," in *Proc. Digital Audio Effects (DAFx)*, 2004.

B.5

Molina, E., Tardón, L. J., Barbancho, I., and Barbancho, A. M. (2014). The importance of F0 tracking in query-by-singing-humming. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 277-282, Taipei (Taiwan).

THE IMPORTANCE OF F0 TRACKING IN QUERY-BY-SINGING-HUMMING

Emilio Molina, Lorenzo J. Tardón, Isabel Barbancho, Ana M. Barbancho
Universidad de Málaga, ATIC Research Group, Andalucía Tech,
ETSI Telecomunicación, Campus de Teatinos s/n, 29071 Málaga, SPAIN
emm@ic.uma.es, lorenzo@ic.uma.es, ibp@ic.uma.es, abp@ic.uma.es

ABSTRACT

In this paper, we present a comparative study of several state-of-the-art F0 trackers applied to the context of query-by-singing-humming (QBSH). This study has been carried out using the well known, freely available, MIR-QBSH dataset in different conditions of added pub-style noise and smartphone-style distortion. For audio-to-MIDI melodic matching, we have used two state-of-the-art systems and a simple, easily reproducible baseline method. For the evaluation, we measured the QBSH performance for 189 different combinations of F0 tracker, noise/distortion conditions and matcher. Additionally, the overall accuracy of the F0 transcriptions (as defined in MIREX) was also measured. In the results, we found that F0 tracking overall accuracy correlates with QBSH performance, but it does not totally measure the suitability of a pitch vector for QBSH. In addition, we also found clear differences in robustness to F0 transcription errors between different matchers.

1. INTRODUCTION

Query-by-singing-humming (QBSH) is a music information retrieval task where short hummed or sung audio clips act as queries. Nowadays, several successful commercial applications for QBSH have been released, such as MusicRadar [1] or SoundHound [2], and it is an active field of research. Indeed, there is a task for QBSH in MIREX since 2006, and every year novel and relevant approaches can be found.

Typically, QBSH approaches firstly extract the F0 contour and/or a note-level transcription for a given vocal query, and then a set of candidate melodies are retrieved from a large database using a melodic matcher module. In the literature, many different approaches for matching in QBSH can be found: statistical, note vs. note, frame vs. note, frame vs. frame. Generally, state-of-the-art systems for QBSH typically combines different approaches in order to achieve more reliable results [3, 12].

[1] www.doreso.com
[2] www.soundhound.com

However, even state-of-the-art systems for QBSH have not a totally satisfactory performance in many real-world cases [1], so there is still room for improvement. Nowadays, some challenges related to QBSH are [2]: reliable pitch tracking in noisy environments, automatic song database preparation (predominant melody extraction and transcription), efficient search in very large music collections, dealing with errors of intonation and rhythm in amateur singers, etc.

In this paper, we analyse the performance of various state-of-the-art F0 trackers for QBSH in different conditions of background noise and smartphone-style distortion. For this study, we have considered three different melodic matchers: two state-of-the-art systems (one of which obtained the best results in MIREX 2013), and a simple, easily reproducible baseline method based on frame-to-frame matching using dynamic time warping (DTW). In Figure 1, we show a scheme of our study.

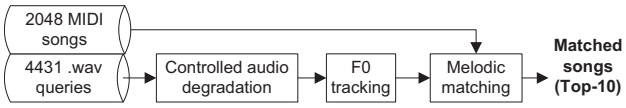

Figure 1. Overall scheme of our study

This paper is organized as follows: Section 2 and Section 3 present the studied algorithms for F0 tracking and melodic matching, respectively. The evaluation strategy is presented in Section 4. Section 5 presents the obtained results and Section 6 draws some conclusions about the present study.

2. F0 TRACKERS

In this section, we describe the F0 trackers considered in our study, together with their specific set of parameters. The literature reports a wide set of algorithms oriented to either monophonic or polyphonic audio, so we have focused on well-known, commonly used algorithms (e.g. Yin [4] or Praat-AC [8]), and some recently published algorithms for F0 estimation (e.g. pYin [6] or MELODIA [15]). Most of the algorithms analysed address F0 estimation in monophonic audio, but we have also studied the performance of MELODIA, which is a method for predominant melody extraction in polyphonic audio, using monophonic audio in noisy conditions. Regarding the used set of parameters, when possible, they have been adjusted by trial and error using ten audio queries. The considered methods for F0 tracking are the following ones:

2.1 YIN

The Yin algorithm was developed by de Cheveigné and Kawahara in 2002 [4]. It resembles the idea of the autocorrelation method [5] but it uses the cumulative mean normalized difference function, which peaks at the local period with lower error rates than the traditional autocorrelation function. In our study, we have used Matthias Mauch's VAMP plugin [3] in Sonic Annotator tool [4].

Parameters used in YIN: step size = 80 samples (0.01 seconds), Block size = 512 samples, Yin threshold = 0.15.

2.2 pYIN

The pYin method has been published by Mauch in 2014 [6], and it basically adds a HMM-based F0 tracking stage in order to find a "smooth" path through the fundamental frequency candidates obtained by Yin. Again, we have used the original Matthias Mauch's VAMP plugin [3] in Sonic Annotator tool [4].

Parameters used in PYIN: step size = 80 samples (0.01 seconds), Block size = 512 samples, Yin threshold distribution = Beta (mean 0.15).

2.3 AC-DEFAULT and AC-ADJUSTED (Praat)

Praat is a well-known tool for speech analysis [7], which includes several methods for F0 estimation. In our case, we have chosen the algorithm created by P. Boersma in 1993 [8]. It is based on the autocorrelation method, but it improves it by considering the effects of the window during the analysis and by including a F0 tracking stage based on dynamic programming. This method has 9 parameters that can be adjusted to achieve a better performance for a specific application. According to [9], this method significantly improves its performance when its parameters are adapted to the input signal. Therefore, we have experimented not only with the default set of parameters (AC-DEFAULT), but also with an adjusted set of parameters in order to limit octave jumps and false positives during the voicing process (AC-ADJUSTED). In our case, we have used the implementation included in the console Praat tool.

Parameters used in AC-DEFAULT: Time step = 0.01 seconds, Pitch floor = 75Hz, Max. number of candidates = 15, Very accurate = off, Silence threshold = 0.03, Voicing threshold = 0.45, Octave cost = 0.01, Octave-jump cost = 0.35, Voiced / unvoiced cost = 0.15, Pitch ceiling = 600 Hz.

Parameters used in AC-ADJUSTED: Time step = 0.01 seconds, Pitch floor = 50Hz, Max. number of candidates = 15, Very accurate = off, Silence threshold = 0.03, Voicing threshold = 0.45, Octave cost = 0.1, Octave-jump cost = 0.5, Voiced / unvoiced cost = 0.5, Pitch ceiling = 700 Hz.

2.4 AC-LEIWANG

In our study we have also included the exact F0 tracker used in Lei Wang's approach for QBSH [3], which obtained the best results for most of the datasets in MIREX 2013. It is based on P. Boersma's autocorrelation method [8], but it uses a finely tuned set of parameters and a post-processing stage in order to mitigate spurious and octave errors. This F0 tracker is used in the latest evolution of a set of older methods [11, 12] also developed by Lei Wang (an open source C++ implementation is available [5]).

2.5 SWIPE'

The Swipe' algorithm was published by A. Camacho in 2007 [10]. This algorithm estimates the pitch as the fundamental frequency of the sawtooth waveform whose spectrum best matches the spectrum of the input signal. The algorithm proved to outperform other well-known F0 estimation algorithms, and it is used in the F0 estimation stage of some state-of-the-art query-by-humming systems [13]. In our study, we have used the original author's Matlab implementation [6]. The Matlab code does not provide a voiced / unvoiced classification of frames, but it outputs a strength vector `S` which has been used for it. Specifically, a frame is considered voiced if its strength is above a threshold S_{th}, otherwise they are considered unvoiced.

Parameters used in SWIPE': DT (hop-size) = 0.01 seconds, pmin = 50 Hz, pmax = 700Hz, dlog2p = 1/48 (default), dERBs = 0.1 (default), woverlap = 0.5 (default), voicing threshold S_{th} = 0.3.

2.6 MELODIA-MONO and MELODIA-POLY

MELODIA is a system for automatic melody extraction in polyphonic music signals developed by Salamon in 2012 [15]. This system is based on the creation and characterisation of pitch contours, which are time continuous sequences of pitch candidates grouped using auditory streaming cues. Melodic and non-melodic contours are distinguished depending on the distributions of its characteristics. The used implementation is MELODIA VAMP plugin [7] in Sonic Annotator tool [4]. This plugin has two default sets of parameters, adapted to deal with monophonic or polyphonic audio. We have experimented with both of them, and therefore we have defined two methods: MELODIA-MONO and MELODIA-POLY.

Parameters used in MELODIA-MONO: Program = Monophonic, Min Frequency = 55Hz, Max Frequency = 700Hz, Voicing Tolerance = 3,00, Monophonic Noise Filter = 0,00, Audio block size = 372 (not configurable), Window increment = 23 (not configurable).

Parameters used in MELODIA-POLY: Program = Polyphonic, Min Frequency = 55Hz, Max Frequency = 700Hz, Voicing Tolerance = 0,20, Monophonic Noise Filter = 0,00, Audio block size = 372 (not configurable), Window increment = 23 (not configurable).

Note that the time-step in this case can not be directly set to 0.01 seconds. Therefore, we have linearly interpolated the pitch vector in order to scale it to a time-step of 0.01 seconds.

[3] http://code.soundsoftware.ac.uk/projects/pyin
[4] http://www.vamp-plugins.org/sonic-annotator/
[5] http://www.atic.uma.es/ismir2014qbsh/
[6] http://www.cise.ufl.edu/ acamacho/publications/swipep.m
[7] http://mtg.upf.edu/technologies/melodia

3. AUDIO-TO-MIDI MELODIC MATCHERS

In this section, we describe the three considered methods for audio-to-MIDI melodic matching: a simple baseline (Section 3.1) and two state-of-the-art matchers (Sections 3.2 and 3.3).

3.1 Baseline approach

We have implemented a simple, freely available[5] baseline approach based on dynamic time warping (DTW) for melodic matching. Our method consists of four steps (a scheme is shown in Figure 2):

(1) Model building: We extract one pitch vector $\mathbf{P^k}$ (in MIDI number) for every target MIDI song $k \in 1 \ldots N_{\text{songs}}$ using a hop-size of 0.01 seconds. Then we replace unvoiced frames (rests) in $\mathbf{P^k}$ by the pitch value of the previous note, except for the case of initial unvoiced frames, which are directly removed (these processed pitch vectors are labelled as $\mathbf{P^{*k}}$). Then, each pitch vector $\mathbf{P^{*k}}$ $\forall k \in 1 \ldots N_{\text{songs}}$ is truncated to generate 7 pitch vectors with lengths [500, 600, 700, 800, 900, 1000, 1100] frames (corresponding to the first 5, 6, 7, 8, 9, 10 and 11 seconds of the target MIDI song, which are reasonable durations for an user query). We label these pitch vectors as $\mathbf{P^{*k}_{5s}}$, $\mathbf{P^{*k}_{6s}}$, $\ldots \mathbf{P^{*k}_{11s}}$. Finally, all these pitch vectors are resampled (through linear interpolation) to a length of 50 points, and then zero-mean normalized (for a common key transposition), leading to $\mathbf{P^{50*k}_{\mathit{Duration}}}$ $\forall \mathit{Duration} \in 5s \ldots 11s$ and $\forall k \in 1 \ldots N_{\text{songs}}$. These vectors are then stored for later usage. Note that this process must be done only once.

(2) Query pre-processing: The pitch vector $\mathbf{P^Q}$ of a given .wav query is loaded (note that all pitch vectors are computed with a hopsize equal to 0.01 seconds). Then, as in step (1), unvoiced frames are replaced by the pitch value of the previous note, except for the case of initial unvoiced frames, which are directly removed. This processed vector is then converted to MIDI numbers with 1 cent resolution, and labelled as $\mathbf{P^{*Q}}$. Finally, $\mathbf{P^{*Q}}$ is resampled (using linear interpolation) to a length $L = 50$ and zero-mean normalized (for a common key transposition), leading to $\mathbf{P^{50*Q}}$.

(3) DTW-based alignment: Now we find the optimal alignment between $\mathbf{P^{50*Q}}$ and all pitch vectors $\mathbf{P^{50*k}_{\mathit{Duration}}}$ $\forall \mathit{Duration} \in 5s \ldots 11s$ and $\forall k \in 1 \ldots N_{\text{songs}}$ using dynamic time warping (DTW). In our case, each cost matrix $\mathbf{C^{Duration,k}}$ is built using the squared difference:

$$C^{Duration,k}(i,j) = (P^{50*Q}(i) - P^{50*k}_{Duration}(j))^2 \quad (1)$$

Where k is the target song index, $Duration$ represents the truncation level (from 5s to 11s), and i, j are the time indices of the query pitch vector $\mathbf{P^{50*Q}}$ and the target pitch vector $\mathbf{P^{50*k}_{\mathit{Duration}}}$, respectively. The optimal path is now found using Dan Ellis' Matlab implementation for DTW [16] (`dpfast.m` function), with the following allowed steps and associated cost weights $[\Delta i, \Delta j, W]$: $[1,1,1]$, $[1,0,30]$, $[0,1,30]$, $[1,2,5]$, $[2,1,5]$. The allowed steps and weights have been selected in order to penalize 0 or 90 angles in the optimal path (associated to unnatural alignments), and although they lead to acceptable results, they may not be optimal.

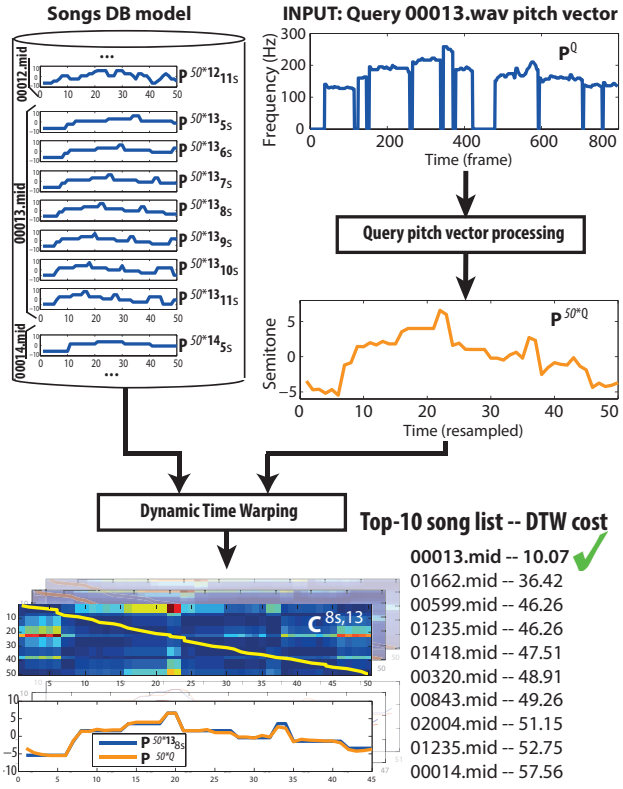

Figure 2. Scheme of the proposed baseline method for audio-to-MIDI melody matching.

(4) Top-10 report: Once the $\mathbf{P^{50*Q}}$ has been aligned with all target pitch vectors (a total of $7 \times N_{\text{songs}}$ vectors, since we use 7 different durations), the matched pitch vectors are sorted according to their alignment total cost (this value consists of the matrix `D` produced by `dpfast.m` evaluated in the last position of the optimal path, `Tcost = D(p(end),q(end))`). Finally, the 10 songs with minimum cost are reported.

3.2 Music Radar's approach

MusicRadar [3] is a state-of-the-art algorithm for melodic matching, which participated in MIREX 2013 and obtained the best accuracy in all datasets, except for the case of IOACAS[8]. It is the latest evolution of a set of systems developed by Lei Wang since 2007 [11, 12]. The system takes advantage of several matching methods to improve its accuracy. First, Earth Mover's Distance (EMD), which is note-based and fast, is adopted to eliminate most unlikely candidates. Then, Dynamic Time Warping (DTW), which is frame-based and more accurate, is executed on these surviving candidates. Finally, a weighted voting fusion strategy is employed to find the optimal match. In our study, we have used the exact melody matcher tested in MIREX 2013, provided by its original author.

3.3 NetEase's approach

NetEase's approach [13] is a state-of-the-art algorithm for melodic matching, which participated in MIREX 2013 and

[8] http://www.music-ir.org/mirex/wiki/2013:Query_by_Singing/Humming

obtained the first position for IOACAS dataset [8], as well as relevant results in the rest of datasets. This algorithm adopts a two-stage cascaded solution based on Locality Sensitive Hashing (LSH) and accurate matching of frame-level pitch sequence. Firstly, LSH is employed to quickly filter out songs with low matching possibilities. In the second stage, Dynamic Time Warping is applied to find the N (set to 10) most matching songs from the candidate list. Again, the original authors of NetEase's approach (who also authored some older works on query-by-humming [14]) collaborated in this study, so we have used the exact melody matcher tested in MIREX 2013.

4. EVALUATION STRATEGY

In this section, we present the datasets used in our study (Section 4.1), the way in which we have combined F0 trackers and melody matchers (Section 4.2) and the chosen evaluation measures (Section 4.3).

4.1 Datasets

We have used the public corpus MIR-QBSH [8] (used in MIREX since 2005), which includes 4431 .wav queries corresponding to 48 different MIDI songs. The audio queries are 8 seconds length, and they are recorded in mono 8 bits, with a sample rate of 8kHz. In general, the audio queries are monophonic with no background noise, although some of them are slightly noisy and/or distorted. This dataset also includes a manually corrected pitch vector for each .wav query. Although these annotations are fairly reliable, they may not be totally correct, as stated in MIR-QBSH documentation.

In addition, we have used the Audio Degradation Toolbox [17] in order to recreate common environments where a QBSH system could work. Specifically, we have combined three levels of pub-style added background noise (`PubEnvironment1` sound) and smartphone-style distortion (`smartPhoneRecording` degradation), leading to a total of seven evaluation datasets: (1) Original MIR-QBSH corpus (2) 25 dB SNR (3) 25 dB SNR + smartphone distortion (4) 15 dB SNR (5) 15 dB SNR + smartphone distortion (6) 5 dB SNR (7) 5 dB SNR + smartphone distortion. Note that all these degradations have been checked in order to ensure perceptually realistic environments.

Finally, in order to replicate MIREX conditions, we have included 2000 extra MIDI songs (randomly taken from ESSEN collection [9]) to the original collection of 48 MIDI songs, leading to a songs collection of 2048 MIDI songs. Note that, although these 2000 extra songs fit the style of the original 48 songs, they do not correspond to any .wav query of Jang's dataset.

4.2 Combinations of F0 trackers and melody matchers

For each of the 7 datasets, the 4431 .wav queries have been transcribed using the 8 different F0 trackers mentioned in Section 2. Additionally, each dataset also includes the 4431 manually corrected pitch vectors of MIR-QBSH as a reference, leading to a total of 7 datasets $\times$ (8 F0 trackers + 1 manual annotation) $\times$ 4431 queries = 63 $\times$ 4431 queries = 279153 pitch vectors. Then, all these pitch vectors have been used as input to the 3 different melody matchers mentioned in Section 3, leading to 930510 lists of top-10 matched songs. Finally, these results have been used to compute a set of meaningful evaluation measures.

4.3 Evaluation measures

In this section, we present the evaluation measures used in this study:

(1) Mean overall accuracy of F0 tracking ($\overline{\text{Acc}_{\text{ov}}}$): For each pitch vector we have computed an evaluation measures defined in MIREX Audio Melody Extraction task: *overall accuracy* (Acc_{ov}) (a definition can be found in [15]). The *mean overall accuracy* is then defined as $\overline{\text{Acc}_{\text{ov}}} = (1/N)\sum_{i=1}^{N} \text{Acc}_{\text{ov}i}$, where N is the total number of queries considered and $\text{Acc}_{\text{ov}i}$ is the overall accuracy of the pitch vector of the i:th query. We have selected this measure because it considers both voicing and pitch, which are important aspects in QBSH. For this measure, our ground truth consists of the manually corrected pitch vectors of the .wav queries, which are included in the original MIR-QBSH corpus.

(2) Mean Reciprocal Rank (MRR): This measure is commonly used in MIREX Query By Singing Humming task [8], and it is defined as: $\text{MRR} = (1/N)\sum_{i=1}^{N} r_i^{-1}$, where N is the total number of queries considered and r_i is the rank of the correct answer in the retrieved melodies for i:th query.

5. RESULTS & DISCUSSION

In this section, we present the obtained results and some relevant considerations about them.

5.1 $\overline{\text{Acc}_{\text{ov}}}$ and MRR for each F0 tracker - Dataset - Matcher

In Table 1, we show the $\overline{\text{Acc}_{\text{ov}}}$ and the MRR obtained for the whole dataset of 4431 .wav queries in each combination of F0 tracker-dataset-matcher (189 combinations in total). Note that these results are directly comparable to MIREX Query by Singing/Humming task [8] (Jang Dataset). As expected, the manually corrected pitch vectors produce the best MRR in most cases (the overall accuracy is 100% because it has been taken as the ground truth for such measure). Note that, despite manual annotations are the same in all datasets, NetEase and MusicRadar matchers do not produce the exact same results in all cases. It is due to the generation of the indexing model (used to reduced the time search), which is not a totally deterministic process.

Regarding the relationship between $\overline{\text{Acc}_{\text{ov}}}$ and MRR in the rest of F0 trackers, we find a somehow contradictory result: the best $\overline{\text{Acc}_{\text{ov}}}$ does not always correspond with the best MRR. This fact may be due to two different reasons. On the one hand, the meaning of $\overline{\text{Acc}_{\text{ov}}}$ may be distorted due to annotation errors in the ground truth (as mentioned in Section 4.1), or to eventual intonation errors in the dataset. However, the manual annotations produce the best MRR, what suggests that the amount of these types

[9] www.esac-data.org/

F0 tracker	Clean dataset	25dB SNR	25 dB SNR + distortion	15dB SNR	15 dB SNR + distortion	5dB SNR	5 dB SNR + distortion
(A)	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.89 / 0.95	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.88 / 0.95
(B)	89 / **0.80** / **0.89** / **0.96**	89 / **0.80** / **0.89** / **0.96**	**88** / **0.80** / **0.88** / **0.95**	88 / **0.79** / **0.88** / **0.94**	84 / 0.71 / 0.86 / 0.94	78 / 0.50 / 0.73 / 0.85	67 / 0.33 / 0.57 / 0.73
(C)	**90** / 0.74 / 0.85 / 0.94	90 / 0.71 / 0.85 / 0.92	86 / 0.72 / 0.84 / 0.92	89 / 0.71 / 0.84 / 0.92	85 / 0.66 / 0.81 / 0.89	72 / 0.49 / 0.58 / 0.70	64 / 0.26 / 0.39 / 0.51
(D)	90 / 0.71 / 0.83 / 0.92	**90** / 0.74 / 0.85 / 0.93	85 / 0.74 / 0.85 / 0.94	**90** / 0.78 / 0.87 / 0.94	**85** / **0.77** / **0.87** / **0.94**	79 / **0.69** / **0.79** / **0.87**	72 / **0.58** / **0.69** / **0.81**
(E)	89 / 0.71 / 0.83 / 0.92	89 / 0.71 / 0.84 / 0.92	84 / 0.66 / 0.80 / 0.91	88 / 0.72 / 0.84 / 0.93	83 / 0.65 / 0.80 / 0.91	75 / 0.67 / 0.67 / 0.82	66 / 0.48 / 0.53 / 0.73
(F)	86 / 0.62 / 0.81 / 0.89	86 / 0.70 / 0.83 / 0.92	81 / 0.64 / 0.78 / 0.89	82 / 0.60 / 0.77 / 0.88	75 / 0.50 / 0.67 / 0.82	48 / 0.03 / 0.08 / 0.04	44 / 0.04 / 0.04 / 0.03
(G)	88 / 0.56 / 0.81 / 0.88	87 / 0.47 / 0.79 / 0.86	83 / 0.47 / 0.76 / 0.85	86 / 0.39 / 0.78 / 0.87	81 / 0.35 / 0.73 / 0.82	70 / 0.11 / 0.32 / 0.52	63 / 0.04 / 0.20 / 0.38
(H)	87 / 0.66 / 0.83 / 0.87	87 / 0.67 / 0.82 / 0.87	83 / 0.64 / 0.78 / 0.84	86 / 0.66 / 0.81 / 0.84	82 / 0.58 / 0.74 / 0.80	83 / 0.51 / 0.73 / 0.75	73 / 0.32 / 0.55 / 0.62
(I)	84 / 0.62 / 0.76 / 0.86	84 / 0.62 / 0.76 / 0.86	79 / 0.50 / 0.64 / 0.74	84 / 0.63 / 0.76 / 0.86	79 / 0.50 / 0.65 / 0.75	**83** / 0.60 / 0.73 / 0.83	**75** / 0.39 / 0.55 / 0.65

Table 1: F0 overall accuracy and MRR obtained for each case. F0 trackers: (A) *MANUALLY CORRECTED* (B) *AC-LEIWANG* (C) *AC-ADJUSTED* (D) *PYIN* (E) *SWIPE'* (F) *YIN* (G) *AC-DEFAULT* (H) *MELODIA-MONO* (I) *MELODIA-POLY*. The format of each cell is: $\overline{\text{Acc}_{\text{ov}}}(\%)$ / *MRR-baseline* / *MRR-NetEase* / *MRR-MusicRadar*.

of errors are low. On the other hand, the measure $\overline{\text{Acc}_{\text{ov}}}$ itself may not be totally representative of the suitability of a pitch vector for QBSH. Indeed, after analysing specific cases, we observed that two pitch vectors with same F0 tracking accuracy (according to MIREX measures) may not be equally suitable for query-by-humming. For instance, we analysed the results produced by the baseline matcher using two different pitch vectors (Figure 3) with exactly the same evaluation measures in MIREX Audio Melody Extraction task: *vocing recall* $= 99.63\%$, *voicing false-alarm* $= 48.40\%$, *raw pitch accuracy*$= 97.41\%$, *raw-chroma accuracy* $= 97.41\%$ and *overall accuracy* $= 82.91\%$. However, we found that pitch vector (a) matches the right song with rank $r_i = 1$ whereas pitch vector (b) does not matches the right song at all ($r_i \geq 11$). The reason is that MIREX evaluation measures do not take into account the pitch values of false positives, but in fact they are important for QBSH. Therefore, we conclude that the high MRR achieved by some F0 trackers (AC-LEIWANG when background noise is low, and PYIN for highly degraded signals), is not only due to the amount of errors made by them, but also to the type of such errors.

Additionally, we observed that, in most cases, the queries are matched either with rank $r_i = 1$ or $r_i \geq 11$ (intermediate cases such as rank $r_i = 2$ or $r_i = 3$ are less frequent). Therefore, the variance of ranks is generally high, their distribution is not Gaussian.

5.2 MRR vs. $\overline{\text{Acc}_{\text{ov}}}$ for each matcher

In order to study the robustness of each melodic matcher to F0 tracking errors, we have represented the MRR obtained by each one for different ranges of $\overline{\text{Acc}_{\text{ov}}}$ (Figure 4). For this experiment, we have selected only the .wav queries which produce the right answer in first rank for the three matchers considered (baseline, Music Radar and NetEase) when manually corrected pitch vectors are used (around a 70% of the dataset matches this condition). In this way, we ensure that bad singing or a wrong manual annotation is not affecting the variations of MRR in the plots. Note that, in this case, the results are not directly comparable to the ones computed in MIREX (in contrast to the results shown in Section 5.1).

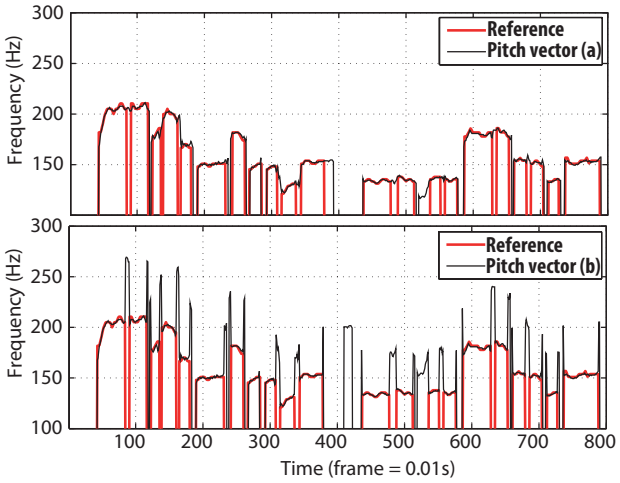

Figure 3. According to MIREX measures, these two pitch vectors (manually manipulated) are equally accurate; however, they are not equally suitable for QBSH.

Regarding the obtained results (shown in Figure 4), we observe clear differences in the robustness to F0 estimation errors between matchers, which is coherent with the results presented in Table 1. The main difference is found in the baseline matcher with respect to both NetEase and Music Radar. Given that the baseline matcher only uses DTW, whereas the other two matchers use a combination of various searching methods (see Sections 3.2 and 3.3), we hypothesise that such combination may improve their robustness to F0 tracking errors. However, further research is needed to really test this hypothesis.

6. CONCLUSIONS

In this paper, eight different state-of-the-art F0 trackers were evaluated for the specific application of query-by-humming-singing in different conditions of pub-style added noise and smartphone-style distortion. This study was carried out using three different matching methods: a simple, freely available baseline (a detailed description has been provided in Section 3.1) and two state-of-the-art matchers. In our results, we found that Boersma's AC method [8], with an appropriate adjustment and a smoothing stage

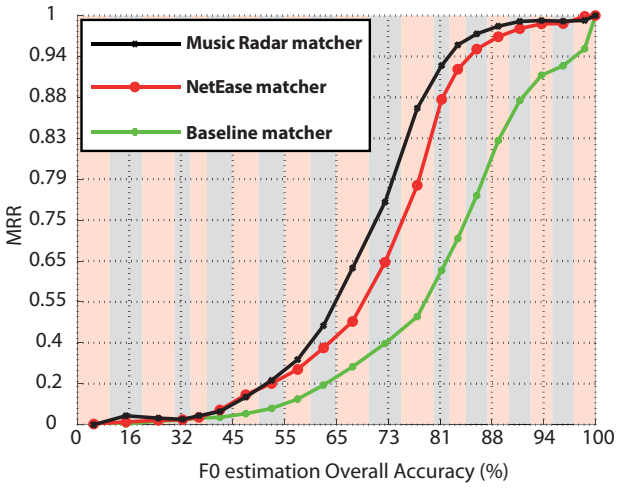

Figure 4. MRR obtained for each range of Overall Accuracy (each range is marked with coloured background rectangles). We have considered only the .wav queries which, using manually corrected F0 vectors, produce MRR $= 1$ in all matchers.

achieves the best results when the audio is not very degraded. In contrast, when the audio is highly degraded, the best results are obtained with pYIN [6], even without further smoothing. Considering that pYIN is a very recent, open source approach, this result is promising in order to improve the noise robustness of future QBSH systems. Additionally, we found that F0 trackers perform differently on QBSH depending on the type of F0 tracking errors made. Due to this, MIREX measures do not fully represent the suitability of a pitch vector for QBSH purposes, so the development of novel evaluation measures in MIREX is encouraged to really measure the suitability of MIR systems for specific applications. Finally, we observed clear differences between matchers regarding their robustness to F0 estimation errors. However, further research is needed for a deeper insight into these differences.

7. ACKNOWLEDGEMENTS

Special thanks to Doreso [1] team (especially to Lei Wang and Yuhang Cao) and to Peng Li for their active collaboration in this study. This work has been funded by the Ministerio de Economía y Competitividad of the Spanish Government under Project No. TIN2013-47276-C6-2-R and by the Junta de Andalucía under Project No. P11-TIC-7154. The work has been done at Universidad de Málaga. Campus de Excelencia Internacional Andalucía Tech.

8. REFERENCES

[1] A. D. Brown and Brighthand staff: "SoundHound for Android OS Review: 'Name That Tune,' But At What Price?", *Brighhand Smartphone News & Review*, 2012. Online: www.brighthand.com [Last Access: 28/04/2014]

[2] J. -S. Roger Jang: "QBSH and AFP as Two Successful Paradigms of Music Information Retrieval" Course in *RuSSIR*, 2013. Available at: http://mirlab.org/jang/ [Last Access: 28/04/2014]

[3] Doreso Team (www.doreso.com): "MIREX 2013 QBSH Task: Music Radar's Solution" *Extended abstract for MIREX*, 2013.

[4] A. De Cheveigné and H. Kawahara: "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustic Society of America*, Vol. 111, No. 4, pp. 1917-1930, 2002.

[5] L. Rabiner: "On the use of autocorrelation analysis for pitch detection," *Acoustics, Speech and Signal Processing, IEEE Transactions on,*, Vol. 25, No.1, pp. 24-33. 1977.

[6] M. Mauch, and S. Dixon, "pYIN: A fundamental frequency estimator using probabilistic threshold distributions," *Proceedings of ICASSP*, 2014.

[7] P. Boersma and D. Weenink: "Praat: a system for doing phonetics by computer," *Glot international*, Vol. 5, No. 9/10, pp. 341-345, 2002. Software available at: www.praat.org [Last access: 28/04/2014]

[8] P. Boersma: "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," *Proceedings of the Institute of Phonetic Sciences*, Vol. 17, No. 1193, pp 97-110, 1993.

[9] E. Keelan, C. Lai, K. Zechner: "The importance of optimal parameter setting for pitch extraction," *Journal of Acoustical Society of America*, Vol. 128, No. 4, pp. 2291–2291, 2010.

[10] A. Camacho: "SWIPE: A sawtooth waveform inspired pitch estimator for speech and music," PhD dissertation, University of Florida, 2007.

[11] L. Wang, S. Huang, S. Hu, J. Liang and B. Xu: "An effective and efficient method for query-by-humming system based on multi-similarity measurement fusion," *Proceedings of ICALIP*, 2008.

[12] L. Wang, S. Huang, S. Hu, J. Liang and B. Xu: "Improving searching speed and accuracy of query by humming system based on three methods: feature fusion, set reduction and multiple similarity measurement rescoring," *Proceedings of INTERSPEECH*, 2008.

[13] P. Li, Y. Nie and X. Li: "MIREX 2013 QBSH Task: Netease's Solution" *Extended abstract for MIREX*, 2013.

[14] P. Li, M. Zhou, X. Wang and N. Li: "A novel MIR system based on improved melody contour definition," *Proceedings of the International Conference on Multi-Media and Information Technology (MMIT)*, 2008.

[15] J. Salamon and E. Gómez: "Melody Extraction from Polyphonic Music Signals using Pitch Contour Characteristics," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 20, No. 6, pp. 1759–1770, 2012.

[16] D. Ellis: "Dynamic Time Warp (DTW) in Matlab", 2003. Web resource, available: www.ee.columbia.edu/ dpwe/resources/matlab/dtw/ [Last Access: 28/04/2014]

[17] M. Mauch and S. Ewert: "The Audio Degradation Toolbox and its Application to Robustness Evaluation," *Proceedings of ISMIR*, 2013.

B.6

Molina, E., Tardón, L. J., Barbancho, A. M., and Barbancho, I. (2015). SiPTH: Singing transcription based on hysteresis defined on the pitch-time curve. *IEEE/ACM Transactions on Acoustics, Speech and Language Processing*, 23(2):252-263.

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.

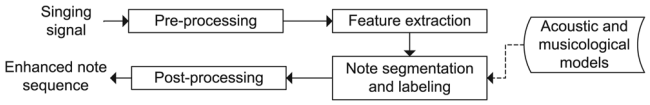

Fig. 1. Diagram of a singing transcription algorithm shown in [3].

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

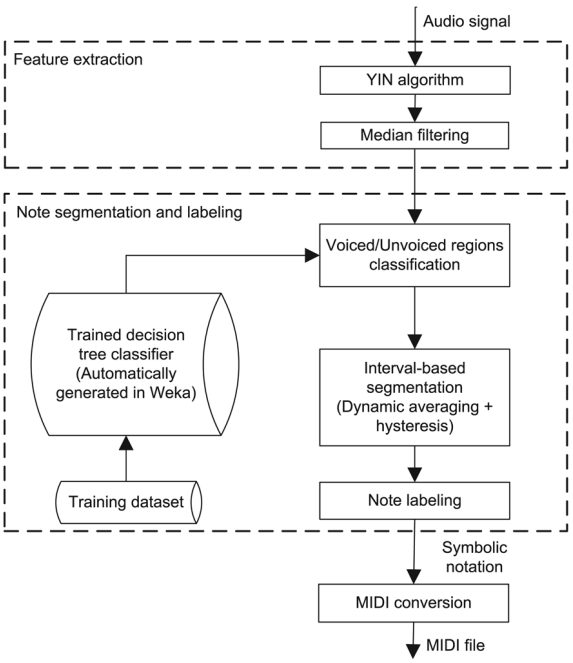

Fig. 2. Scheme of the proposed algorithm for note segmentation and labeling.

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:

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

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.

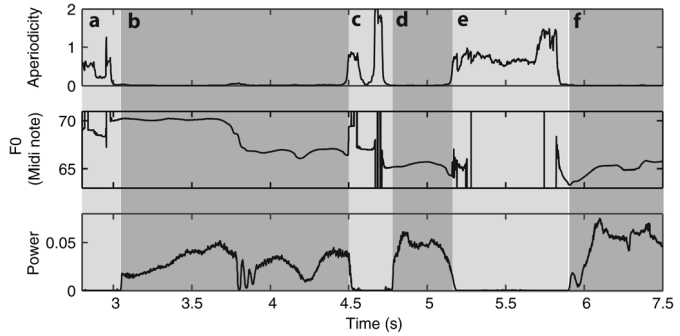

Fig. 3. Output of the YIN algorithm for a child singing performance: fundamental frequency, power, and aperiodicity over time. In this figure, actual sung notes have been marked with shadowed rectangles.

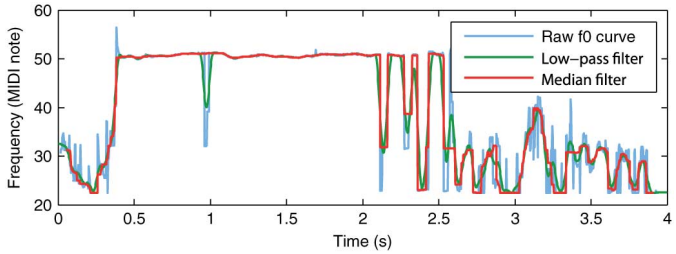

Fig. 4. Sample of application of low-pass and median filtering to a raw pitch curve. The spurious gap at second 1 is removed by the median filter whilst it remains after low-pass filtering.

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

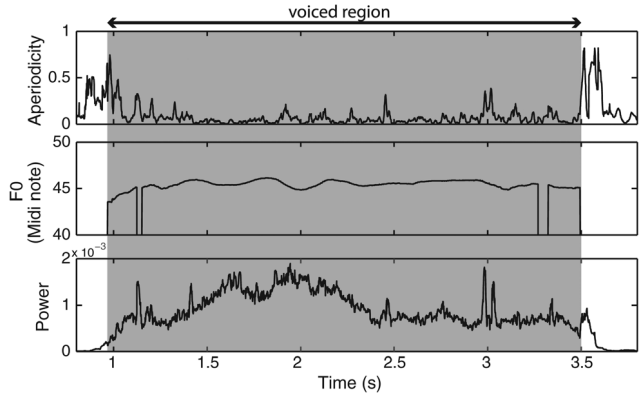

Fig. 5. Extracted features for the case of a rough timbre voice (old male voice). Pitch stability is a better criterion to identify the voiced segment than aperiodicity.

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.

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

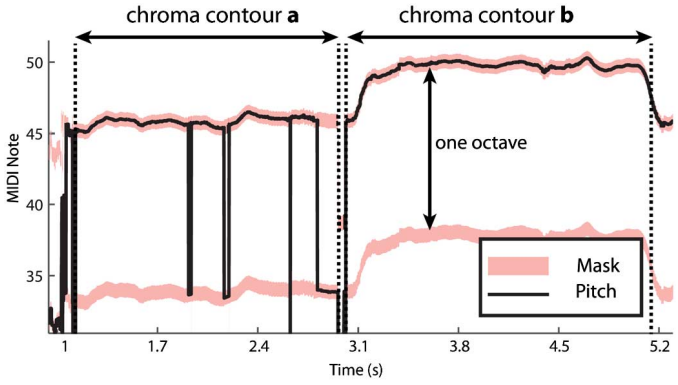

Fig. 6. Stable pitch detection. The black curve represents the estimated pitch value; red regions represent the mask where pitch values can vary between consecutive frames. The use of a mask that allows fast octave jumps avoids fake note changes if octave errors happen.

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

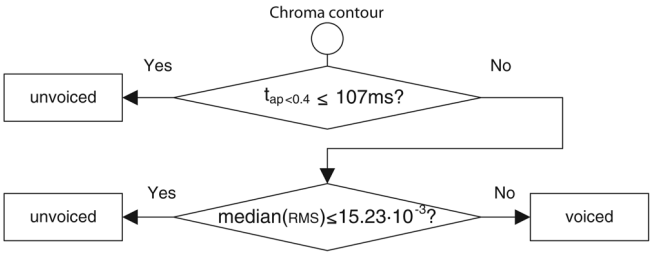

Fig. 7. Decision tree generated by using the C4.5 algorithm implemented in the Weka data-mining software with a very low confidence factor (strong pruning) for the classification of voice/unvoiced frames.

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,

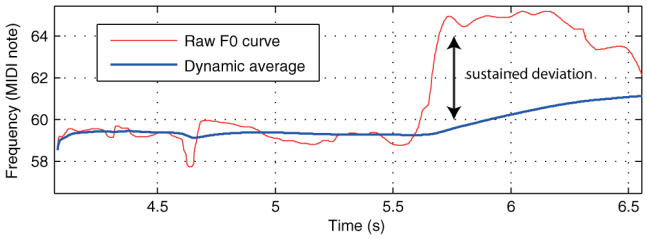

Fig. 8. Dynamic averaging of the pitch curve. Fast variations of pitch at the beginning of the note are tracked, whereas later strong changes can be easily detected.

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 $l > l_0$:

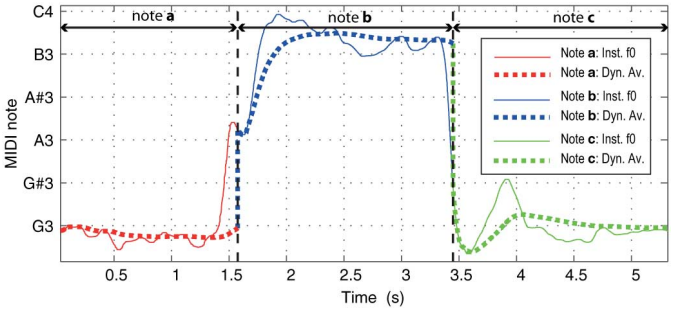

Fig. 9. Representation of the hysteresis process for the detection of note changes. Samples are taken from real data: from $\approx G3$ to $\approx B3$ to $\approx G3$. The instantaneous F0 and the dynamic average $\mathrm{F0}_A$ for each note are shown. Strong and/or sustained deviations of the instantaneous 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

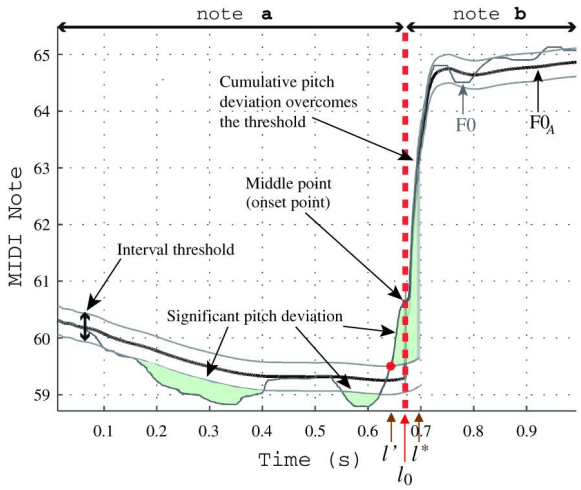

Fig. 10. Segmentation process. The dynamic averaging and the hysteresis effect are used to detect note changes. In this example, the interval threshold δ_{th} has been set to 0.25 semitones.

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 $M_{\gamma \to \phi_{i_0 j}} = 0 \ \forall j \in \{1 \dots N_\phi\}$. 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 $M_{\phi \to \gamma_{j_0 i}} = 0 \ \forall i \in \{1 \dots N_\gamma\}$. That means that there is no overlap between the note j_0 from the transcription and the whole ground truth.
- **Split note (S) ($i_0 \to j_0 \dots j_n$)**: We consider that a note i_0 in the ground truth is split into a set of notes $j_0 \dots j_n$ in the transcription if $M_{\phi \to \gamma_{j_{0\dots n} i_0}} > 0$ with $n > 1$, $od_{i_0 j_0} < 50$ ms and $\sum_0^n M_{\phi \to \gamma_{j_{0\dots n} i_0}} > t$.
- **Merged note (M) ($i_0 \dots i_n \to j_0$)**: We consider that several notes $i_0 \dots i_n$ in the ground truth are merged into a single note j_0 in the transcription if $M_{\gamma \to \phi_{i_{0\dots n} j_0}} > 0$ with $n > 1$, $od_{i_0 j_0} < 50$ ms and $\sum_0^n M_{\gamma \to \phi_{i_{0\dots n} j_0}} > t$. 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].

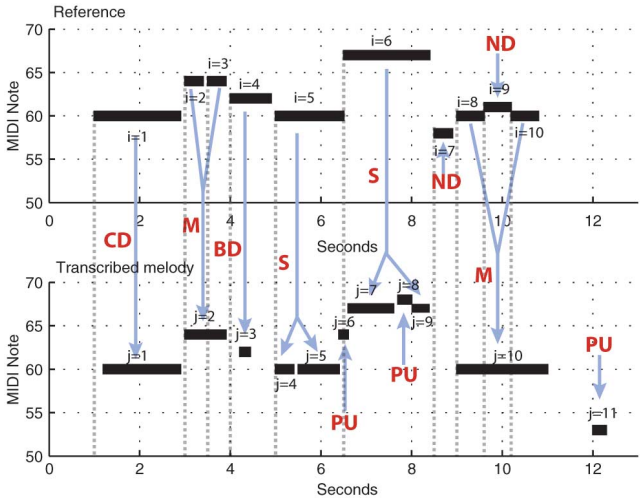

Fig. 11. Example of comparison between the ground truth and the transcribed melody. All the error types defined are illustrated using a sample outcome of the proposed evaluation algorithm.

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-

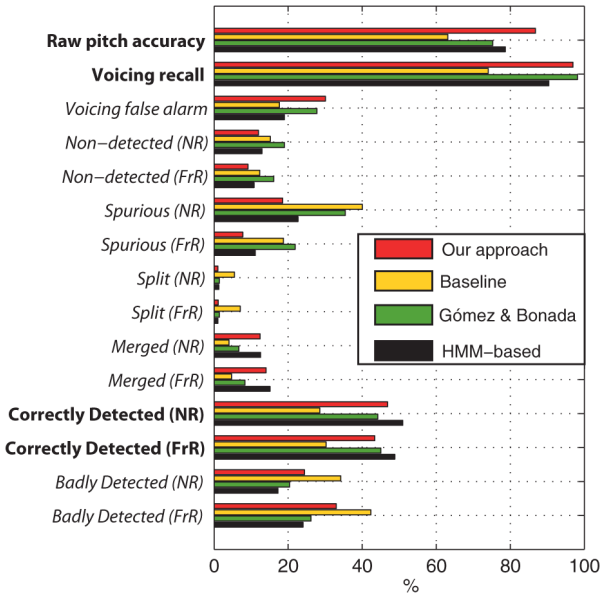

Fig. 12. Detailed performance evaluation of the monophonic singing transcription system proposed, the baseline approach, the HMM approach and the Gómez and Bonada transcription scheme [4] (confidence interval set to 0.5 semitones). The measures that should be maximized are in bold and the measures that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

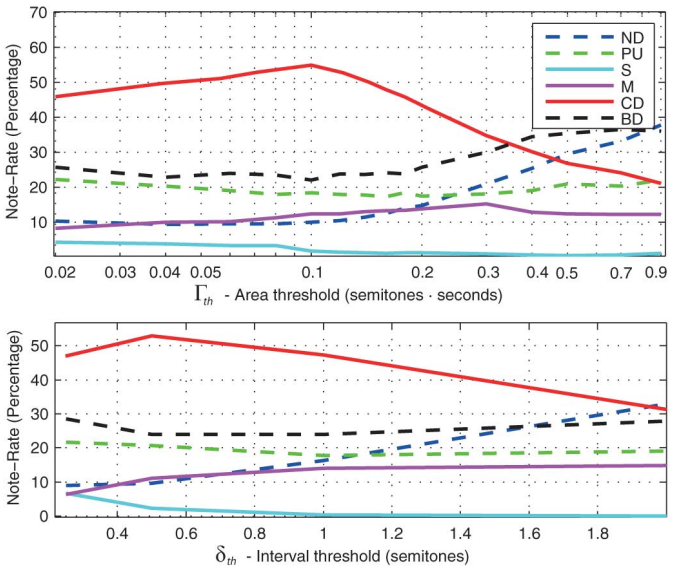

Fig. 13. Illustration of the influence of the system parameters δ_{th} and Γ_{th} on the pitch-based segmentation process. The optimal performance is achieved when the rate of correctly detected notes (CD) is maximum, and the rates of the rest of errors (ND, PU, S, M and BD) are minimum. (Top: area threshold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

Bibliography

[Al-Naymat et al., 2009] Al-Naymat, G., Chawla, S., and Taheri, J. (2009). SparseDTW: a novel approach to speed up dynamic time warping. In *Proceedings of the 8th Australasian Data Mining Conference*, pages 117–127. Australian Computer Society, Inc. ↑25

[Alku et al., 2013] Alku, P., Pohjalainen, J., Vainio, M., Laukkanen, A. M., and Story, B. H. (2013). Formant frequency estimation of high-pitched vowels using weighted linear prediction. *Journal of the Acoustical Society of America*, 134(2):1295–1313. ↑36

[Anguera, 2012] Anguera, X. (2012). Expert Talk for Time Machine Session in ICME 2012: Dynamic Time Warping New Youth. `http://videolectures.net/icme2012_anguera_time_warping/`. Last access: 2016/03/01. ↑25

[Anguera and Ferrarons, 2013] Anguera, X. and Ferrarons, M. (2013). Memory efficient subsequence DTW for Query-by-Example Spoken Term Detection. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME 2013)*, pages 1–6. ↑25

[ANSI, 2004] ANSI (2004). S1.1-1994 (R2004) Acoustical Terminology. ↑34

[Baker et al., 2009] Baker, J. M., Li, D., Glass, J., Khudanpur, S., Lee, C. H., Morgan, N., and O'Shaughnessy, D. (2009). Research developments and directions in speech recognition and understanding, Part 1. *IEEE Signal Processing Magazine*, 26(3):75–80. ↑37

[Barbancho et al., 2010] Barbancho, A. M., Tardón, L. J., and Barbancho, I. (2010). PIC Detector for Piano Chords. *EURASIP Journal on Advances in Signal Processing*, 2010(1). ↑18

[Barbancho et al., 2013] Barbancho, A. M., Barbancho, I., Tardón, L. J., and Molina, E. (2013). *A Database of Piano Chords: An Engineering View of Harmony*. Springer. ↑93

[Becker et al., 2008] Becker, T., Jessen, M., and Grigoras, C. (2008). Forensic speaker verification using formant features and Gaussian mixture models. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association (INTERSPEECH 2008)*, pages 1505–1508. ↑36

[Bednar and Watt, 1984] Bednar, J. and Watt, T. (1984). Alpha-trimmed means and their relationship to median filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 32(1):145–153. ↑62

[Bello and Sandler, 2000] Bello, J. P. and Sandler, M. (2000). Blackboard system and top-down processing for the transcription of simple polyphonic music. In *Proceedings of the 3rd International Conference on Digital Audio Effects (DAFx 2000)*, pages 7–11. ↑19

[Bergee, 2003] Bergee, M. J. (2003). Faculty Interjudge Reliability of Music Performance Evaluation. *Journal of Research in Music Education*, 51(2):137. ↑28

[Böck et al., 2012] Böck, S., Arzt, A., Krebs, F., and Schedl, M. (2012). Online realtime onset detection with recurrent neural networks. In *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx 2012)*, pages 15–18. ↑39, ↑40

[Boersma, 1993] Boersma, P. (1993). Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. *Proceedings of the Institute of Phonetic Sciences*, 17(1193):97–110. ↑iii, ↑v, ↑14, ↑16, ↑52, ↑99

[Bogdanov et al., 2013] Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J., and Serra, X. (2013). ESSENTIA: An audio analysis library for music information retrieval. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR 2013)*, pages 493–498. ↑97

[Bonada et al., 2001] Bonada, J., Celma, Ò., Loscos, A., Ortolà, J., and Serra, X. (2001). Singing voice synthesis combining excitation plus resonance and sinusoidal plus residual models. In *Proceedings of the International Computer Music Conference (ICMC 2001)*, pages 139–146. ↑74

[Bonada and Serra, 2007] Bonada, J. and Serra, X. (2007). Synthesis of the singing voice by performance sampling and spectral models. *IEEE Signal Processing Magazine*, 24(2):67–79. ↑2, ↑36

[Borges et al., 2008] Borges, J., Couto, I., Oliveira, F., Imbiriba, T., Klautau, A., and Bruckert, E. (2008). GASpeech: A framework for automatically estimating

input parameters of Klatt's speech synthesizer. In *Proceedings of the 10th Brazilian Symposium on Neural Networks (SBRN 2008)*, pages 81–86, Salvador, Bahia, Brazil. ↑36

[Bozkurt et al., 2004] Bozkurt, B., Dutoit, T., Doval, B., and D'Alessandro, C. (2004). Improved differential phase spectrum processing for formant tracking. In *Proceedings of the 8th International Conference on Spoken Language Processing (INTERSPEECH 2004 - ICSLP)*, Jeju Island, Korea. ↑36

[Bruderlin and Williams, 1995] Bruderlin, A. and Williams, L. (1995). Motion signal processing. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 97–104, New York, New York, USA. ACM Press. ↑25

[Burges, 1998] Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167. ↑39

[Busby and Plant, 1995] Busby, P. A. and Plant, G. L. (1995). Formant frequency values of vowels produced by preadolescent boys and girls. *Journal of the Acoustical Society of America*, 97(4):2603–2606. ↑36

[Camacho and Harris, 2008] Camacho, A. and Harris, J. G. (2008). A sawtooth waveform inspired pitch estimator for speech and music. *Journal of the Acoustical Society of America*, 124(3):1638–1652. ↑15, ↑52

[Cañadas-Quesada et al., 2008] Cañadas-Quesada, F. J., Vera-Candeas, P., Ruiz-Reyes, N., Mata-Campos, R., and Carabias-Orti, J. J. (2008). Note-event detection in polyphonic musical signals based on harmonic matching pursuit and spectral smoothness. *Journal of New Music Research*, 37(3):167–183. ↑19

[Cancela, 2008] Cancela, P. (2008). Tracking melody in polyphonic audio. In *Extended Abstract for Music Information Retrieval Evaluation eXchange (MIREX 2008)*. ↑17

[Carlsson and Sundberg, 1992] Carlsson, G. and Sundberg, J. (1992). Formant frequency tuning in singing. *Journal of Voice*, 6(3):256–260. ↑36

[Cemgil et al., 2006] Cemgil, A., Kappen, H., and Barber, D. (2006). A generative model for music transcription. *IEEE Transactions on Audio, Speech and Language Processing*, 14(2):679–694. ↑19

[Childers et al., 1977] Childers, D. G., Skinner, D. P., and Kemerait, R. C. (1977). The Cepstrum: A Guide to Processing. *Proceedings of the IEEE*, 65(10):1428–1443. ↑33

[Clarisse et al., 2002] Clarisse, L. P., Martens, J. P., Lesaffre, M., Baets, B. D., Meyer, H. D., and Leman, M. (2002). An Auditory Model Based Transcriber of Singing Sequences. In *Proceedings of the 3rd International Society for Music Information Retrieval Conference (ISMIR 2002)*, pages 116–123, Paris, France. ↑21

[Cont, 2006] Cont, A. (2006). Realtime multiple pitch observation using sparse non-negative constraints. In *Proceedings of the 7th International Society for Music Information Retrieval Conference (ISMIR 2006)*, pages 206–211, Victoria, Canada. ↑19

[Cook, 1991] Cook, P. (1991). *Identification of Control Parameters in an Articulatory Vocal Tract Model, with Applications to the Synthesis of Singing*. PhD thesis, Stanford University. ↑2, ↑12

[Cook, 1996] Cook, P. (1996). Singing voice synthesis: history, current work, and future directions. *Computer Music Journal*, 20(3):38–46. ↑2

[Cook, 1999] Cook, P. (1999). Pitch, periodicity and noise in voice. In *Music, Cognition, and Computerized Sound*, pages 195–208. MIT Press. ↑12

[Dahl et al., 2010] Dahl, G., Mohamed, A.-R., and Hinton, G. (2010). Phone recognition with the mean-covariance restricted Boltzmann machine. In *Advances in Neural Information Processing Systems (NIPS)*, pages 469–477. ↑40

[De Cheveigné and Kawahara, 2002] De Cheveigné, A. and Kawahara, H. (2002). YIN, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America*, 111(4):1917. ↑14, ↑16, ↑52, ↑61, ↑67, ↑69, ↑99

[De Mulder et al., 2003] De Mulder, T., Martens, J.-P., Lesaffre, M., Leman, M., Baets, B. D., and Meyer, H. D. (2003). An auditory model based transcriber of vocal queries. In *Proceedings of the 4th International Society for Music Information Retrieval Conference (ISMIR 2003)*, pages 26–30. ↑21

[De Mulder et al., 2004] De Mulder, T., Martens, J. P., Lesaffre, M., Leman, M., Baets, B. D., and Meyer, H. D. (2004). Recent improvements of an auditory model based front-end for the transcription of vocal queries. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2004)*, pages 257–260, Montreal, Quebec, Canada. ↑21, ↑67, ↑99

[Deng et al., 2006] Deng, L. D. L., Cui, X. C. X., Pruvenok, R., Chen, Y. C. Y., Momen, S., and Alwan, A. (2006). A Database of Vocal Tract Resonance Trajectories for Research in Speech Processing. In *Proceedings of the IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP 2006), Toulouse, France. ↑36, ↑37

[Dittmar et al., 2010] Dittmar, C., Großmann, H., Cano, E., and Al., E. (2010). Songs2See and GlobalMusic2One: two applied research projects in music information retrieval at Fraunhofer IDMT. In *Proceedings of the 7th International Symposium on Computer Music Modeling and Retrieval (CMMR 2010)*, pages 259 – 272, Málaga, Spain. ↑3, ↑20, ↑27, ↑99

[Dixon and Widmer, 2005] Dixon, S. and Widmer, G. (2005). MATCH: a music alignment tool chest. In *Proceedings of the 6th International Society for Music Information Retrieval Conference (ISMIR 2005)*, pages 492–497, London, UK. ↑25

[Doreso, 2013] Doreso (2013). MIREX 2013 QBSH Task: MusicRadar's solution. In *Extended Abstract for MIREX Query by Singing/Humming (QBSH) Task*. ↑19, ↑52, ↑53

[Dressler, 2006] Dressler, K. (2006). Sinusoidal extraction using an efficient implementation of a multi-resolution FFT. In *Proceedings of the 9th International Conference on Digital Audio Effects (DAFx 2006)*, pages 247–252, Montreal, Quebec, Canada. ↑17

[Dressler, 2011] Dressler, K. (2011). Pitch estimation by the pair-wise evaluation of spectral peaks. In *Proceedings of the Audio Engineering Society 42th International Conference (AES 2011)*. ↑17

[Duda et al., 2007] Duda, A., Nürnberger, A., and Stober, S. (2007). Towards query by singing / humming on audio databases. In *Proceedings of the 8th International Society for Music Information Retrieval Conference (ISMIR 2007)*, pages 331–334, Vienna, Austria. ↑3

[Durrieu, 2010] Durrieu, J. (2010). *Automatic transcription and separation of main melody in polyphonic music signals*. PhD thesis, Télécom ParisTech. ↑18

[Ekholm et al., 1998] Ekholm, E., Papagiannis, G. C., and Chagnon, F. P. (1998). Relating objective measurements to expert evaluation of voice quality in Western classical singing: critical perceptual parameters. *Journal of Voice*, 12(2):182–196. ↑28

[Ellis, 2003] Ellis, D. (2003). Dynamic Time Warp (DTW) in Matlab. `http://www.ee.columbia.edu/$\sim$dpwe/resources/matlab/dtw`. ↑24

[Ellis, 2005] Ellis, D. (2005). PLP and RASTA (and MFCC, and inversion) in Matlab. `http://labrosa.ee.columbia.edu/matlab/rastamat/`. ↑37, ↑38

[Ellis, 1996] Ellis, D. P. W. (1996). *Prediction-driven computational auditory scene analysis*. PhD thesis, Massachusetts Institute of Technology. ↑19

[Fujihara et al., 2011] Fujihara, H., Goto, M., Ogata, J., and Okuno, H. G. (2011). Lyric synchronizer : Automatic synchronization system between musical audio signals and lyrics. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1252–1261. ↑2

[Garofolo, 1993] Garofolo, J. (1993). *TIMIT: acoustic-phonetic continuous speech corpus*. Linguistic Data Consortium. ↑36

[Giese and Poggio, 2000] Giese, M. A. and Poggio, T. (2000). Morphable Models for the Analysis and Synthesis of Complex Motion Patterns. *International Journal of Computer Vision*, 38(1):59–73. ↑25

[Gläser et al., 2010] Gläser, C., Heckmann, M., Joublin, F., and Goerick, C. (2010). Combining auditory preprocessing and bayesian estimation for robust formant tracking. *IEEE Transactions on Audio, Speech and Language Processing*, 18(2):224–236. ↑36

[Gold and Rabiner, 1969] Gold, B. and Rabiner, L. (1969). Parallel processing techniques for estimating pitch periods of speech in the time domain. *Journal of the Acoustical Society of America*, 46(2):442–448. ↑15

[Gómez et al., 2003a] Gómez, E., Peterschmitt, G., Amatriain, X., and Herrera, P. (2003a). Content-based melodic transformations of audio material for a music processing application. In *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx 2003)*, pages 1–6, London, UK. ↑71

[Gómez et al., 2003b] Gómez, E., Klapuri, A., and Meudic, B. (2003b). Melody description and extraction in the context of music content processing. *Journal of New Music Research*, 32(1):23–40. ↑13, ↑15

[Gómez et al., 2013] Gómez, E., Bonada, J., and Emilia, G. (2013). Towards computer-assisted flamenco transcription: an experimental comparison of automatic transcription algorithms as applied to a cappella singing. *Computer Music Journal*, 37(2):73–90. ↑2, ↑21, ↑23, ↑67

[Goto, 2000] Goto, M. (2000). A robust predominant-F0 estimation method for real-time detection of melody and bass lines in CD recordings. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2000)*, pages 757–760. ↑19

[Goto et al., 2003] Goto, M., Hashiguchi, H., Nishimura, T., and Oka, R. (2003). RWC Music Database: Music Genre Database and Musical Instrument Sound Database. In *Proceedings of the 4th International Society for Music Information Retrieval Conference (ISMIR 2003)*, pages 229–230, Baltimore, Maryland, USA. ↑84

[Goto, 2004] Goto, M. (2004). A real-time music-scene-description system: predominant-F0 estimation for detecting melody and bass lines in real-world audio signals. *Speech Communication*, 43(4):311–329. ↑17

[Goto et al., 2010] Goto, M., Saitou, T., Nakano, T., and Fujihara, H. (2010). Singing information processing based on singing voice modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2010)*, pages 5506–5509. ↑2

[Goto, 2014] Goto, M. (2014). Singing Information Processing. In *Proceedings of the 12th International Conference on Signal Processing (ICSP 2004)*, pages 7–14, Hangzhou, China. ↑2

[Gray and Wong, 1980] Gray, A. J. and Wong, D. (1980). The Burg algorithm for LPC speech analysis/Synthesis. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(6):609–615. ↑31

[Gribonval and Bacry, 2003] Gribonval, R. and Bacry, E. (2003). Harmonic decomposition of audio signals with matching pursuit. *IEEE Transactions on Signal Processing*, 51(1):101–111. ↑19

[Griffiths and Davidson, 2006] Griffiths, N. and Davidson, J. (2006). The effects of concert dress and physical appearance on perceptions of female solo performers. In *Proceedings of the International Conference on Music Perception and Cognition (ICMPC 2006)*, pages 1723–1726. ↑27

[Grollmisch et al., 2011] Grollmisch, S., Cano Cerón, E., and Dittmar, C. (2011). Songs2See: Learn to Play by Playing. In *Proceedings of Audio Engineering Society Conference: 41st International Conference: Audio for Games.* ↑3

[Guaus, 2009] Guaus, E. (2009). *Audio content processing for automatic music genre classification: descriptors, databases, and classifiers.* PhD thesis, Universitat Pompeu Fabra (Barcelona). ↑39

[Hasan et al., 2004] Hasan, R., Jamil, M., Rabbani, G., and Rahman, S. (2004). Speaker identification using mel frequency cepstral coefficients. In *Proceedings of the 3rd International Conference on Electrical & Computer Engineering (ICECE 2004)*, pages 28–30, Dhaka, Bangladesh. ↑37

[Haus and Pollastri, 2001] Haus, G. and Pollastri, E. (2001). An Audio Front End for Query-by-Humming Systems. In Downie, J. S. and Bainbridge, D., editors, *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval (ISMIR 2001)*, pages 65–72, Bloomington, Indiana, USA. Indiana University. ↑16, ↑21

[Hermansky, 1990] Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America*, 87(4):1738–1752. ↑37, ↑38

[Hermansky and Morgan, 1994] Hermansky, H. and Morgan, N. (1994). RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589. ↑38

[Hinton et al., 2012] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, 35(8):82–97. ↑37, ↑39, ↑40

[Hiroaki, 1978] Hiroaki, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26(1):43–46. ↑24, ↑25, ↑69

[Holmes et al., 1997] Holmes, J. N., Holmer, W. J., and Garner, P. N. (1997). Using formant frequencies in speech recognition. In *Proceedings of European Conference on Speech Communication and Technology (EUROSPEECH 1997)*, pages 2083–2086, Rhodes, Greece. ↑36

[Howard et al., 2004] Howard, D. M., Welch, G., Brereton, J., Himonides, E., Decosta, M., Williams, J., and Howard, A. (2004). WinSingad: a real-time display for the singing studio. *Logopedics Phoniatrics Vocology*, 29(3):135–144. ↑3, ↑27

[Hsu and Jang, 2010] Hsu, C.-l. and Jang, J.-s. R. (2010). Singing pitch extraction by voice vibrato/tremolo estimation and instrument partial deletion. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, pages 525–530, Utrecht, Netherlands. ↑17

[Hsu et al., 2005] Hsu, E., Pulli, K., and Popović, J. (2005). Style translation for human motion. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2005)*, pages 1082–1089. ↑25

[Hu et al., 2003] Hu, N., Dannenberg, R. B., and Tzanetakis, G. (2003). Polyphonic audio matching and alignment for music retrieval. In *Proceedings of the IEEE*

Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), pages 185–188. ↑25

[Ibañez, 2010] Ibañez, A. P. (2010). *Computationally efficient methods for polyphonic music transcription*. PhD thesis, University of Alicante. ↑18

[Imai and Abe, 1979] Imai, S. and Abe, Y. (1979). Spectral envelope extraction by improved cepstral method. *Journal of IEICE (in japanese)*, 62(4):10–17. ↑29, ↑33

[Jansen et al., 2010] Jansen, A., Church, K., and Hermansky, H. (2010). Towards spoken term discovery at scale with zero resources. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH 2010)*, pages 1676–1679. ↑25

[Jansen and Church, 2011] Jansen, A. and Church, K. (2011). Towards unsupervised training of speaker independent acoustic models. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH 2011)*. ↑25

[Jespersen, 1922] Jespersen, O. (1922). *Language: its nature, development and origin*. London : G. Allen & Unwin, ltd. ↑1

[Kameoka et al., 2007] Kameoka, H., Nishimoto, T., and Sagayama, S. (2007). A multipitch analyzer based on harmonic temporal structured clustering. *IEEE Transactions on Audio, Speech and Language Processing*, 15(3):982–994. ↑19

[Kan et al., 2008] Kan, M. Y., Wang, Y., Iskandar, D., Nwe, T. L., and Shenoy, A. (2008). LyricAlly: Automatic synchronization of textual lyrics to acoustic music signals. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2):338–349. ↑2

[Kanato et al., 2014] Kanato, A., Nakano, T., Goto, M., and Kikuchi, H. (2014). An automatic singing impression estimation method using factor analysis and multiple regression. In *Proceedings of the Joint International Computer Music Conference and Sound and Music Computing Conference (ICMCSMC2014)*, pages 1244–1251, Athens, Greece. ↑3

[Karneback, 2001] Karneback, S. (2001). Discrimination between speech and music based on a low frequency modulation feature. In *Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH 2001)*, pages 1891–1894. ↑38

[Kedem, 1986] Kedem, B. (1986). Spectral analysis and discrimination by zero-crossings. *Proceedings of the IEEE*, 74(11):1477–1493. ↑15, ↑38

[Kenmochi and Ohshita, 2007] Kenmochi, H. and Ohshita, H. (2007). VOCALOID - commercial singing synthesizer based on sample concatenation. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH 2007)*, pages 4009–4010. ↑2

[Keogh and Ratanamahatana, 2004] Keogh, E. and Ratanamahatana, C. A. (2004). Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386. ↑25

[Kim, 2003] Kim, Y. E. (2003). *Singing Voice Analysis / Synthesis*. PhD thesis, MIT. ↑12

[Klapuri, 2003] Klapuri, A. (2003). Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *IEEE Transactions on Speech and Audio Processing*, 11(6):804–816. ↑18

[Klapuri, 2005] Klapuri, A. (2005). A perceptually motivated multiple-F0 estimation method. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2005)*, pages 291–294. IEEE. ↑18, ↑80

[Klapuri, 2008] Klapuri, A. (2008). Multipitch Analysis of Polyphonic Music and Speech Signals Using an Auditory Model. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2):255–266. ↑17, ↑18

[Klatt, 1980] Klatt, D. H. (1980). Software for a cascade/paralell formant synthesizer. *Journal of the Acoustical Society of America*, 67(3):971–995. ↑28, ↑74

[Kovar and Gleicher, 2003] Kovar, L. and Gleicher, M. (2003). Flexible automatic motion blending with registration curves. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, pages 214–224. Eurographics Association. ↑25

[Krige et al., 2008] Krige, W., Herbst, T., and Niesler, T. (2008). Explicit transition modelling for automatic singing transcription. *Journal of New Music Research*, 37(4):311–324. ↑15, ↑22, ↑23

[Lahat et al., 1987] Lahat, M., Niederjohn, R., and Krubsack, D. (1987). A spectral autocorrelation method for measurement of the fundamental frequency of noise-corrupted speech. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(6). ↑15

[LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444. ↑40

[Leveau et al., 2008] Leveau, P., Vincent, E., Richard, G., and Daudet, L. (2008). Instrument-specific harmonic atoms for mid-level music representation. *IEEE Transactions on Audio, Speech and Language Processing*, 16(1):116–128. ↑19

[Levinson, 1947] Levinson, N. (1947). The Wiener RMS error criterion in filter design and prediction. *Journal of Mathematics and Physics*, 25:261–278. ↑32

[Li et al., 2008] Li, P., Wang, X., Zhou, M., and Li, N. (2008). A novel MIR system based on improved melody contour definition. In *Proceedings of the International Conference on MultiMedia and Information Technology (MMIT 2008)*, pages 409–412. ↑3, ↑55

[Li et al., 2013] Li, P., Nie, Y., and Li, X. (2013). Query-by-singing-humming Task : Netease ' S Solution. In *Extended Abstract for MIREX Query by Singing/Humming (QBSH) Task*. ↑16, ↑19, ↑53

[Logan, 2000] Logan, B. (2000). Mel frequency cepstral coefficients for music modeling. In *Proceedings of the 1st International Symposium on Music Information Retrieval (ISMIR 2000)*, Plymouth, Massachusetts. Cambridge Research Laboratory. ↑37

[Maher and Beauchamp, 1994] Maher, R. C. and Beauchamp, J. W. (1994). Fundamental frequency estimation of musical signals using a two-way mismatch procedure. *Journal of the Acoustical Society of America*, 95(4):2254–2263. ↑15, ↑17

[Makhoul, 1975] Makhoul, J. (1975). Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4). ↑30

[Mallat, 1993] Mallat, S. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415. ↑19

[Markel and Gray, 1976] Markel, J. D. and Gray, A. J. (1976). *Linear Prediction of Speech*. Springer Verlag, Berlin. ↑31

[Marolt, 2004a] Marolt, M. (2004a). A connectionist approach to automatic transcription of polyphonic piano music. *IEEE Transactions on Multimedia*, 6(3):439–449. ↑18

[Marolt, 2004b] Marolt, M. (2004b). Networks of adaptive oscillators for partial tracking and transcription of music recordings. *Journal of New Music Research*, 33(1):49–59. ↑18

[Martin, 1996] Martin, K. D. (1996). Automatic transcription of simple polyphonic music. Technical report, MIT Media Lab. ↑19

[Mauch and Ewert, 2013] Mauch, M. and Ewert, S. (2013). The audio degradation toolbox and its application to robustness evaluation. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR 2013)*, pages 83–88, Curitiba, PR, Brazil. ↑51, ↑55

[Mauch, 2014] Mauch, M. (2014). PYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, pages 659–663. ↑iii, ↑v, ↑14, ↑16, ↑52, ↑90

[Mauch et al., 2015a] Mauch, M., Cannam, C., Bittner, R., Fazekas, G., Salamon, J., Dai, J., Bello, J. P., and Dixon, S. (2015a). Computer-aided melody note transcription using the Tony software: accuracy and efficiency. In *Proceedings of the First International Conference on Technologies for Music Notation and Representation (TENOR)*. ↑19, ↑22, ↑23, ↑67, ↑90, ↑94

[Mauch et al., 2015b] Mauch, M., Dixon, S., and Goto, M. (2015b). Why singing is interesting? `http://ismir2015.uma.es/docs/ISMIR2015tutorial_Singing.pdf` Last access: 12-03-2016. ↑2

[Mayergoyz, 1986] Mayergoyz, I. (1986). Mathematical models of hysteresis. *IEEE Transactions on Magnetics*, 22(5):603–608. ↑61

[Mayor et al., 2006] Mayor, O., Bonada, J., and Loscos, A. (2006). The singing tutor: expression categorization and segmentation of the singing voice. *Proceedings of the AES 121st Convention*. ↑26

[McFee et al., 2015] McFee, B., Raffel, C., Liang, D., Ellis, D. P. W., McVicar, M., Battenberg, E., and Nieto, O. (2015). librosa: Audio and Music Signal Analysis in Python. In *Proceedings of the 14th Python science conference (SCIPY 2015)*, pages 18–25. ↑98

[McLoughlin, 2008] McLoughlin, I. V. (2008). Line spectral pairs. *Signal Processing*, 88(3):448–467. ↑32

[McNab et al., 1996] McNab, R. J., Smith, L. A., and Witten, I. H. (1996). Signal Processing for Melody Transcription. *Proceedings of the 19th Australasian Computer Science Conference*, 18(4):301–307. ↑21

[Molina, 2012] Molina, E. (2012). *Automatic scoring of singing voice based on melodic similarity measures*. MSc Thesis. Universitat Pompeu Fabra (Barcelona), Barcelona. ↑3

[Molina et al., 2013] Molina, E., Barbancho, I., Gómez, E., Barbancho, A. M., and Tardón, L. J. (2013). Fundamental frequency alignment vs. note-based melodic similarity for singing voice assessment. In *Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, pages 744–748, Vancouver (Canada). ↑5, ↑50, ↑69, ↑92

[Molina et al., 2014a] Molina, E., Barbancho, A. M., Tardón, L. J., and Barbancho, I. (2014a). Dissonance reduction in polyphonic music using harmonic reorganization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(2):325–334. ↑5, ↑50, ↑79, ↑81, ↑83

[Molina et al., 2014b] Molina, E., Barbancho, A. M., Tardón, L. J., and Barbancho, I. (2014b). Evaluation framework for automatic singing transcription. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 567–572, Taipei (Taiwan). ↑5, ↑21, ↑49, ↑63, ↑92

[Molina et al., 2014c] Molina, E., Barbancho, I., Barbancho, A. M., and Tardón, L. J. (2014c). Parametric model of spectral envelope to synthesize realistic intensity variations in singing voice. In *Proceedings of the 39th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, pages 634–638, Florence (Italy). ↑2, ↑5, ↑50, ↑73, ↑76, ↑92

[Molina et al., 2014d] Molina, E., Tardón, L. J., Barbancho, I., and Barbancho, A. M. (2014d). The importance of F0 tracking in query-by-singing-humming. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 277–282, Taipei (Taiwan). ↑5, ↑16, ↑49, ↑51, ↑52, ↑92

[Molina et al., 2015] Molina, E., Tardón, L. J., Barbancho, A. M., and Barbancho, I. (2015). SiPTH: Singing transcription based on hysteresis defined on the pitch-time curve. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(2):252–263. ↑5, ↑49, ↑61, ↑66, ↑92

[Morris and Clements, 2002] Morris, R. W. and Clements, M. A. (2002). Modification of formants in the line spectrum domain. *IEEE Signal Processing Letters*, 9(1):19–21. ↑31

[Müller et al., 2004] Müller, M., Kurth, F., and Röder, T. (2004). Towards an efficient algorithm for automatic score-to-audio synchronization. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 365–373, Barcelona, Spain. ↑25

[Müller et al., 2006] Müller, M., Mattes, H., and Kurth, F. (2006). An efficient multiscale approach to audio synchronization. In *Proceedings of the 7th International Society for Music Information Retrieval Conference (ISMIR 2006)*, pages 192–197, Victoria, Canada. ↑25

[Müller and Röder, 2006] Müller, M. and Röder, T. (2006). Motion templates for automatic classification and retrieval of motion capture data. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, pages 137–146. Eurographics Association. ↑25

[Müller, 2007] Müller, M. (2007). Dynamic Time Warping. In *Information Retrieval for Music and Motion*. Springer Berlin Heidelberg, Berlin, Heidelberg. ↑24

[Muñoz-Expósito et al., 2005] Muñoz-Expósito, J., Garcia-Galán, S., Ruiz-Reyes, N., Vera-Candeas, P., and Rivas-Pena, F. (2005). Speech / Music Discrimination Using a Single Warped Lpc-Based Feature. In *Proceedings of the 6th International Society for Music Information Retrieval Conference (ISMIR 2005)*, pages 614–617, London, UK. ↑32

[Nakano et al., 2005] Nakano, T., Goto, M., Ogata, J., and Hiraga, Y. (2005). Voice Drummer : A Music Notation Interface of Drum Sounds Using Voice Percussion Input. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*. ↑3

[Nakano et al., 2006] Nakano, T., Goto, M., and Hiraga, Y. (2006). Subjective evaluation of common singing skills using the rank ordering method. In *Proceedings of the International Conference on Music Perception and Cognition (ICMPC 2006)*, pages 1507–1512. ↑28

[Nakano et al., 2007] Nakano, T., Goto, M., and Hiraga, Y. (2007). MiruSinger: a singing skill visualization interface using real-time feedback and music CD recordings as referential data. In *Proceedings of the IEEE International Symposium on Multimedia Workshops (ISMW 2007)*, pages 75–76. ↑27

[Nakano et al., 2009] Nakano, T., Goto, M., and Hiraga, Y. (2009). An automatic singing skill evaluation method for unknown melodies using pitch interval accuracy and vibrato features. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH 2009)*, pages 1706–1709. ↑26, ↑95

[Nichols et al., 2012] Nichols, E., DuHadway, C., Aradhye, H., and Lyon, R. F. (2012). Automatically discovering talented musicians with acoustic analysis of YouTube videos. In *Proceedings of the IEEE International Conference Data Mining*, pages 559–565. IEEE. ↑26

[Noll, 1967] Noll, A. M. (1967). Cepstrum pitch determination. *Journal of the Acoustical Society of America*, 41(2):293–309. ↑15

[Noll, 1969] Noll, A. M. (1969). Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate. In *Proceedings of the Symposium on Computer Processing and Communications*, pages 779–797. ↑15

[Paiva et al., 2006] Paiva, R. P., Mendes, T., and Cardoso, A. (2006). Melody detection in polyphonic musical signals: exploiting perceptual rules, note salience, and melodic smoothness. *Computer Music Journal*, 30(4):80–98. ↑17

[Pardo et al., 2004] Pardo, B., Shifrin, J., and Birmingham, W. (2004). Name that tune: a pilot study in finding a melody from a sung query. *Journal of the American Society for Information Science and Technology*, 55(4):283–300. ↑3, ↑19

[Peeters, 2006] Peeters, G. (2006). Music Pitch Representation by Periodicity Measures Based on Combined Temporal and Spectral Representations. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, volume 5, pages 53–56. IEEE. ↑18

[Plumbley et al., 2002] Plumbley, M. D., Abdallah, S. a., Bello, J. P., Davies, M. E., Monti, G., and Sandler, M. B. (2002). Automatic music transcription and audio source separation. *Cybernetics and Systems*, 33(6):603–627. ↑19

[Poliner et al., 2007] Poliner, G. E., Ellis, D. P. W., Ehmann, A. F., Gomez, E., Streich, S., and Ong, B. (2007). Melody transcription from music audio: approaches and evaluation. *IEEE Transactions on Audio, Speech and Language Processing*, 15(4):1247–1256. ↑18

[Potamianos and Maragos, 1996] Potamianos, A. and Maragos, P. (1996). Speech formant frequency and bandwidth tracking using multiband energy demodulation. *Journal of the Acoustical Society of America*, 99(6):3795–3806. ↑36

[Rabiner, 1977] Rabiner, L. (1977). On the use of autocorrelation analysis for pitch detection. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 25(1). ↑14, ↑16, ↑52

[Rabiner and Schafer, 1978] Rabiner, L. R. and Schafer, R. W. (1978). *Digital processing of speech signals*. Prentice Hall. ↑14

[Rabiner, 1989] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286. ↑23, ↑25

[Raczyński and Ono, 2007] Raczyński, S. and Ono, N. (2007). Multipitch analysis with harmonic nonnegative matrix approximation. In *Proceedings of the 8th International Society for Music Information Retrieval Conference (ISMIR 2007)*, pages 281–386, Vienna, Austria. ↑19

[Raffel et al., 2014] Raffel, C., McFee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D., and Ellis, D. P. W. (2014). mir_eval: A Transparent Implementation of Common MIR Metrics. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 367–372, Taipei, Taiwan. ↑94, ↑98

[Rao and Rao, 2010] Rao, V. and Rao, P. (2010). Vocal melody extraction in the presence of pitched accompaniment in polyphonic music. *IEEE Transactions on Audio, Speech and Language Processing*, 18(8):2145–2154. ↑17

[Röbel and Rodet, 2005] Röbel, A. and Rodet, X. (2005). Efficient Spectral Envelope Estimation and its Application to Pitch Shifting and Envelope Preservation. In *Proceedings of the 8th International Conference on Digital Audio Effects (DAFx 2005)*, Madrid, Spain. ↑29

[Rossiter and Howard, 1996] Rossiter, D. and Howard, D. M. (1996). ALBERT: a real-time visual feedback computer tool for professional vocal development. *Journal of Voice*, 10(4):321–336. ↑3, ↑27

[Ryynänen and Klapuri, 2004] Ryynänen, M. and Klapuri, A. (2004). Modelling of note events for singing transcription. In *Proceedings of the Workshop on Statistical and Perceptual Audio Processing (SAPA 2004)*. ↑23, ↑67

[Ryynänen, 2006] Ryynänen, M. (2006). Singing Transcription. In *Signal Processing Methods for Music Transcription*. Springer. ↑3, ↑16, ↑19, ↑22, ↑23

[Ryynänen, 2008] Ryynänen, M. (2008). *Automatic Transcription of Pitch Content in Music and Selected Applications*. PhD thesis, Tampere University of Technology. ↑67

[Ryynänen and Klapuri, 2008] Ryynänen, M. P. and Klapuri, A. (2008). Query by humming of midi and audio using locality sensitive hashing. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*, pages 2249–2252. ↑17

[Saino et al., 2006] Saino, K., Zen, H., Nankaku, Y., Lee, A., and Tokuda, K. (2006). An HMM-based singing voice synthesis system. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH 2006)*, pages 2274–2277. ↑3

[Sakoe and Chiba, 1971] Sakoe, H. and Chiba, S. (1971). A dynamic programming approach to continuous speech recognition. In *Proceedings of the International Congress on Acoustics*, volume C-13. ↑24

[Salamon and Gómez, 2012] Salamon, J. and Gómez, E. (2012). Melody Extraction From Polyphonic Music Signals Using Pitch Contour Characteristics. *IEEE Transactions on Audio, Speech and Language Processing*, 20(6):1759–1770. ↑16, ↑17, ↑52, ↑56

[Salamon, 2013] Salamon, J. (2013). *Melody Extraction from Polyphonic Music Signals*. PhD thesis, Universitat Pompeu Fabra (Barcelona). ↑99

[Salvador and Chan, 2007] Salvador, S. and Chan, P. (2007). FastDTW: toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11:561–580. ↑25

[Scheirer, 1998] Scheirer, E. D. (1998). Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America*, 103(1):588–601. ↑38, ↑39

[Schlüter and Osendorfer, 2011] Schlüter, J. and Osendorfer, C. (2011). Music similarity estimation with the mean-covariance restricted Boltzmann machine. In *Proceedings of the International Conference on Machine Learning and Applications (ICMLA 2011)*, volume 2, pages 118–123. ↑40

[Schlüter and Sonnleitner, 2012] Schlüter, J. and Sonnleitner, R. (2012). Unsupervised feature learning for speech and music detection in radio broadcasts. In *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx 2012)*, pages 112–115, York, UK. ↑40

[Schramm et al., 2015] Schramm, R., Nunes, H. D. S., and Jung, C. R. (2015). Automatic solfège assessment. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR 2015)*, pages 183–189, Málaga, Spain. ↑26, ↑73

[Schwarz, 2007] Schwarz, D. (2007). Corpus-based concatenative synthesis. *IEEE Signal Processing Magazine*, 24(2):92–104. ↑2

[Serra, 1989] Serra, X. (1989). *A System for Sound Analysis / Transformation / Synthesis based on a Deterministic plus Stochastic Decomposition*. PhD thesis, Stanford University. ↑40, ↑79, ↑95

[Serra and Smith, 2014] Serra, X. and Smith, J. O. (2014). Audio Signal Processing for Music Applications. Coursera. `https://www.coursera.org/course/audio`. ↑40, ↑75

[Shimamura and Kobayashi, 2001] Shimamura, T. and Kobayashi, H. (2001). Weighted autocorrelation for pitch extraction of noisy speech. *IEEE Transactions on Speech and Audio Processing*, 9(7):727–730. ↑14

[Slifka and Anderson, 1995] Slifka, J. and Anderson, T. R. (1995). Speaker modification with LPC pole analysis. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 1995)*, volume 1, pages 644–647, Detroit, Michigan (USA). ↑31, ↑32

[Smaragdis and Brown, 2003] Smaragdis, P. and Brown, J. C. (2003). Non-negative matrix factorization for polyphonic music transcription. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2003)*, pages 177–180. ↑19

[Snell and Milinazzo, 1993] Snell, R. C. and Milinazzo, F. (1993). Formant location from LPC analysis data. *IEEE Transactions on Speech and Audio Processing*, 1(2):129–134. ↑32, ↑36

[Soulez et al., 2008] Soulez, F., Rodet, X., and Schwarz, D. (2008). Improving polyphonic and poly-instrumental music to score alignment. In *Proceedings of the 9th International Society for Music Information Retrieval Conference (ISMIR 2008)*, pages 143–148. ↑25

[Sundberg, 1977] Sundberg (1977). *The Acoustics of the Singing Voice*. Scientific American. ↑10, ↑11

[Sundberg, 1987] Sundberg, J. (1987). *The Science of Singing Voice*. Northern Illinois University Press. ↑10, ↑11

[Sundberg, 2001] Sundberg, J. (2001). Level and center frequency of the singer's formant. *Journal of Voice*, 15(2):176–186. ↑35

[Suzuki et al., 2007] Suzuki, M., Hosoya, T., and Ito, A. (2007). Music information retrieval from a singing voice using lyrics and melody information. *EURASIP Journal on Advances in Signal Processing*, 2007. ↑3

[Tachibana et al., 2010] Tachibana, H., Ono, T., Ono, N., and Sagayama, S. (2010). Melody line estimation in homophonic music audio signals based on temporal-variability of melodic source. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2010)*, pages 425–428. IEEE. ↑18

[Talkin, 1995] Talkin, D. (1995). A robust algorithm for pitch tracking (RAPT). In *Speech Coding and Synthesis*, chapter 14, pages 495–518. Elsevier Science, New York. ↑16

[Titze, 2000] Titze, I. R. (2000). *Principles of Voice Production*. National Center for Voice and Speech. ↑11, ↑34

[Toda et al., 2007] Toda, T., Black, A. W., and Tokuda, K. (2007). Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. *IEEE Transactions on Audio, Speech and Language Processing*, 15(8):2222–2235. ↑3

[Tolonen and Karjalainen, 2000] Tolonen, T. and Karjalainen, M. (2000). A computationally efficient multipitch analysis model. *IEEE Transactions on Speech and Audio Processing*, 8(6):708–716. ↑18

[Tzanetakis and Cook, 2002] Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302. ↑39

[Viitaniemi et al., 2003] Viitaniemi, T., Klapuri, A., and Eronen, A. (2003). A probabilistic model for the transcription of single-voice melodies. In *Proceedings of the 2003 Finnish Signal Processing Symposium FINSIG'03*, pages 59–63. Tampere University of Technology. ↑22, ↑23, ↑67

[Villavicencio et al., 2006] Villavicencio, F., Robel, A., and Rodet, X. (2006). Improving Lpc Spectral Envelope Extraction Of Voiced Speech By True-Envelope Estimation. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, 1. ↑34

[Vintsyuk, 1968] Vintsyuk, T. K. (1968). Speech discrimination by dynamic programming. *Cybernetics*, 4(1):52–57. ↑24

[Virtanen, 2007] Virtanen, T. (2007). Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Transactions on Audio, Speech and Language Processing*, 15(3):1066–1074. ↑19

[Wallin and Merker, 2001] Wallin, N. L. and Merker, B. (2001). *The Origins of Music*. MIT Press. ↑1

[Wang et al., 2010] Wang, C.-C., Jang, J.-s. R., and Wang, W. (2010). An Improved Query by Singing/Humming System Using Melody and Lyrics Information. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, pages 45–50. ↑3, ↑53

[Wang et al., 2008] Wang, L., Huang, S., Hu, S., Liang, J., and Xu, B. (2008). An effective and efficient method for query by humming system based on multi-similarity measurement fusion. In *Proceedings of the International Conference*

on Audio, Language and Image Processing, Proceedings (ICALIP 2008), pages 471–475. ↑3, ↑53

[Wapnick and Ekholm, 1997] Wapnick, J. and Ekholm, E. (1997). Expert consensus in solo voice performance evaluation. *Journal of Voice*, 11(4):429–436. ↑28, ↑72

[Welch et al., 1988] Welch, G. F., Rush, C., and Howard, D. M. (1988). The SINGAD (SINGing Assessment and Development) system: First applications in the classroom. *Proceedings of the Institute of Acoustics*, 10(2):179–185. ↑1

[Xia and Espy-Wilson, 2000] Xia, K. and Espy-Wilson, C. (2000). A new strategy of formant tracking based on dynamic programming. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP / INTERSPEECH 2000)*, pages 10–13, Beijing (China). ↑36

[Yeh, 2008] Yeh, C. (2008). *Multiple Fundamental Frequency Estimation of Polyphonic Recordings*. PhD thesis, Université Paris VI - Pierre et Marie Curie. ↑18

[Yeh et al., 2012] Yeh, T.-c., Wu, M.-j., Jang, J.-s. R., Chang, W.-l., and Liao, I.-b. (2012). A hybrid approach to singing pitch extraction based on trend estimation and hidden Markov models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)*, pages 457–460. ↑17

[Young et al., 2009] Young, S. J., Evermann, G., Gales, M. J. F., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. C. (2009). *The HTK Book (for HTK Version 3.4)*. University of Cambridge. ↑22, ↑38

[Zabell, 2008] Zabell, S. L. (2008). On Student's 1908 article "The probable error of a mean". *Journal of the American Statistical Association*, 103(481):1–7. ↑87

[Zhang, 2003] Zhang, T. Z. T. (2003). Automatic singer identification. In *Proceedings of the International Conference on Multimedia and Expo (ICME 2003)*, pages 33–36. ↑3

[Zheng and Hasegawa-Johnson, 2004] Zheng, Y. Z. Y. and Hasegawa-Johnson, M. (2004). Formant tracking by mixture state particle filter. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2004)*, volume 1, pages 565–568, Montreal (Canada). ↑36

[Zhou, 2006] Zhou, R. (2006). *Feature extraction of musical content for automatic music transcription*. PhD thesis, École Polytechnique Fédérale de Lausanne. ↑18

[Zhou and Mattavelli, 2007] Zhou, R. and Mattavelli, M. (2007). A new time-frequency representation for music signal analysis: resonator time-frequency image. In *Proceedings of the International Symposium on Signal Processing and its Applications*. ↑18

[Zhou et al., 2009] Zhou, R., Reiss, J. D., Mattavelli, M., and Zoia, G. (2009). A computationally efficient method for polyphonic pitch estimation. *EURASIP Journal on Advances in Signal Processing*, 28. ↑18

[Zölzer, 2011] Zölzer, U. (2011). *DAFX: Digital Audio Effects: Second Edition*. John Wiley and Sons. ↑28, ↑29, ↑34

Procesado de Información de Voz Cantada: Técnicas y Aplicaciones

Emilio Molina Martínez

Resumen en Castellano de Tesis Doctoral

Programa de Doctorado en Ingeniería de Telecomunicación
Escuela Técnica Superior de Ingeniería de Telecomunicación
Universidad de Málaga, 2017

Tutor
Lorenzo José Tardón García

Directores
Lorenzo José Tardón García
Ana María Barbancho Pérez

Resumen

La voz cantada es una componente esencial de la música en todas las culturas del mundo, ya que se trata de una forma increíblemente natural de expresión musical. En consecuencia, el procesado automático de voz cantada tiene un gran impacto desde la perspectiva de la industria, la cultura y la ciencia. En este contexto, esta Tesis contribuye con un conjunto variado de técnicas y aplicaciones relacionadas con el procesado de voz cantada, así como con un repaso del estado del arte asociado en cada caso.

En primer lugar, se han comparado varios de los mejores estimadores de tono conocidos para el caso de uso de recuperación por tarareo. Los resultados demuestran que [Boersma, 1993] (con un ajuste no obvio de parámetros) y [Mauch, 2014], tienen un muy buen comportamiento en dicho caso de uso dada la suavidad de los contornos de tono extraídos.

Además, se propone un novedoso sistema de transcripción de voz cantada basada en un proceso de histéresis definido en tiempo y frecuencia, así como una herramienta para evaluación de voz cantada en Matlab. El interés del método propuesto es que consigue tasas de error cercanas al estado del arte con un método muy sencillo. La herramienta de evaluación propuesta, por otro lado, es un recurso útil para definir mejor el problema, y para evaluar mejor las soluciones propuestas por futuros investigadores.

En esta Tesis también se presenta un método para evaluación automática de la interpretación vocal. Usa alineamiento temporal dinámico para alinear la interpretación del usuario con una referencia, proporcionando de esta forma una puntuación de precisión de afinación y de ritmo. La evaluación del sistema muestra una alta correlación entre las puntuaciones dadas por el sistema, y las puntuaciones anotadas por un grupo de músicos expertos.

Por otro lado, se presenta un método para el cambio realista de intensidad de voz cantada. Esta transformación se basa en un modelo paramétrico de la envolvente espectral, y mejora sustancialmente la percepción de realismo al compararlo con software comerciales como Melodyne o Vocaloid. El inconveniente del enfoque propuesto es que requiere intervención manual, pero los resultados conseguidos arrojan importantes conclusiones hacia la modificación automática de intensidad con resultados realistas.

Por último, se propone un método para la corrección de disonancias en acordes aislados. Se basa en un análisis de múltiples F0, y un desplazamiento de la frecuencia de su componente sinusoidal. La evaluación la ha realizado un grupo de músicos entrenados, y muestra un claro incremento de la consonancia percibida después de la transformación propuesta.

Contenido

Sección 1

Introducción

La voz cantada es una componente esencial de la música en todas las culturas del mundo, ya que se trata de una natural y genuina forma de expresión musical. En la actualidad, las tecnologías de grabación de audio y la amplificación han contribuido a la aparición de estilos de canto muy diversos con recursos expresivos variados (e.g. susurros), y en parte debido a ello, la voz cantada tiene un rol claramente protagonista en la mayor parte de los estilos musicales modernos (e.g. pop). En consecuencia, el procesado digital de voz cantada tiene un gran impacto en la sociedad desde el punto de vista de la industria, la cultura y la ciencia.

Sin embargo, al contrario de lo que sucede con el ámbito de procesado de voz hablada, el procesado de voz cantada es un campo de investigación aún inmaduro, y los retos asociados aún estan lejos de ser solucionados para aplicaciones válidas en el mundo real: transcripción a nivel de nota, modificación realista del timbre, transcripción y alineamiento de letra, etc. Muchos de estos problemas están prácticamente resueltos para instrumentos musicales, pero las soluciones empleadas suelen fracasar cuando se aplican a voz cantada. La razón es la gran variabilidad de la voz cantada, la cual se ve afectada por factores como: cantante (género, timbre, formación...), estilo musical (e.g. rap es completamente diferente a ópera), presencia o no de letra, etc. En consecuencia, es necesaria mucha investigación aún para superar estos retos asociados al procesado de voz cantada.

Contexto científico: Procesamiento de Información de Voz Cantada

El área de investigación llamado Procesamiento de Información de Voz Cantada (Singing Information Processing) [Goto et al., 2010] [Goto, 2014] se define como

"procesamiento de información musical para voz cantada". Algunos de los problemas abordados en este ámbito de investigación son:

- Síntesis de voz cantada [Cook, 1991] [Cook, 1996] [Bonada and Serra, 2007] [Schwarz, 2007] [Kenmochi and Ohshita, 2007]
- Transcripción y sincronización de letra [Kan et al., 2008] [Fujihara et al., 2011]
- Análisis y procesado de timbre: conversión de voz [Toda et al., 2007], identificación de cantante [Zhang, 2003], reconocimiento de emoción [Kanato et al., 2014], etc.
- Sistemas de recuperación de información musical (*Music Information Retrieval*), como por ejemplo sistemas de búsqueda musical por canto o tarareo (query-by-singing-humming) [Wang et al., 2008] [Li et al., 2008], o búsqueda por percusión vocal [Nakano et al., 2005].
- Transcripción de voz cantada a notas [Ryynänen, 2006] [Pardo et al., 2004] [Dittmar et al., 2010].
- Modificación de la curva de tono, sobre todo estudiada en el ámbito comercial (por ejemplo Melodyne[1] o Auto-tune[2]).
- Evaluación automática de la habilidad de canto [Rossiter and Howard, 1996] [Howard et al., 2004] [Saino et al., 2006] [Grollmisch et al., 2011] [Molina, 2012].

Temas abordados en esta Tesis

Esta Tesis aborda varios temas específicos relacionados con este ámplio campo de investigación.

Se analiza la importancia de la estimación de tono en sistemas de búsqueda por canto o tarareo. Para este análisis se ha llevado a cabo un estudio comparativo con estimadores de tono del estado del arte con una colección de datos ampliamente usada para estudiar sistemas de búsqueda por canto o tarareo [Molina et al., 2014d].

Además, se presenta un sistema de transcripción de voz cantada a notas utilizando un proceso de histéresis en la curva tiempo-tono [Molina et al., 2015], así como un marco de evaluación en Matlab para transcripción de voz cantada [Molina

[1] www.celemony.com
[2] www.antarestech.com

et al., 2014b].

También se presenta un método para evaluación automática de la habilidad de canto basado en el uso de alineamiento temporal dinámico (dynamic time warping) para obtener información de alineamiento entre la interpretación vocal del usuario y una referencia [Molina et al., 2013].

Por otro lado, se presenta un estudio acerca de la evolución de la envolvente espectral de voz cantada en función de la intensidad, junto con un método para producir variaciones realistas de intensidad en voz cantada [Molina et al., 2014c].

Finalmente, se propone un método para reducir la disonancia de acordes grabados (vocales o instrumentales) mediante estimación de múltiples frecuencias fundamentales y el posterior procesado de su componente sinusoidal. [Molina et al., 2014a]

1.1 Objetivos de investigación

Los objetivos de investigación de esta Tesis incluyen tanto técnicas como aplicaciones en el ámbito del Procesado de Información de Voz Cantada. Estos objetivos son:

- Revisar el estado del arte de los problemas abordados en esta Tesis. Esta revisión debe ser especialmente profunda para los temas principales de esta Tesis: estimación de tono, transcripción de voz cantada a notas, evaluación automática de la habilidad de canto y procesado de timbre de voz.
- Desarrollar un sistema de transcripción de voz cantada a notas con una tasa de error al nivel, al menos, del estado del arte. Este objetivo se puede subdividir en varios sub-objetivos:
  - Definir una metodología de investigación clara para abordar el problema de transcripción de voz cantada: decidir qué tipo de datos deben utilizarse, qué métricas de evaluación son relevantes y cuáles son los métodos del estado del arte disponibles para comparar.
  - Crear una colección de voz cantada monofónica con anotaciones a nivel de nota.
  - Coleccionar otros métodos del estado del arte para transcripción de voz cantada para comparar con ellos.
  - Contruir una herramienta para la evaluación de transcripción de voz cantada y hacerla públicamente disponible.

  - Investigar y desarrollar un método para transcripción automática de voz cantada a nivel de nota.

- Investigar y desarrollar un sistema para evaluación automática de la habilidad de canto basado en una comparación de curvas de tono, y de secuencia de notas con respecto a una referencia.

- Investigar y desarrollar un sistema para modelar los cambios tímbricos producidos en voz cantada en función de la intensidad. Este objetivo también incluye desarrollar una herramienta software para visualizar y anotar la envolvente espectral de una colección de vocales cantadas.

Sección 2

Resumen de resultados

En esta sección se resume el capítulo 3 de la Tesis completa, donde se presentan los aspectos más relevantes de cada resultado conseguido durante esta investigación. Específicamente, se presenta un análisis comparativo de estimadores de tono para sistemas de búsqueda de música por canto o tarareo (sección 2.1), un método transcripción de voz cantada a notas y una herramienta de evaluación (sección 2.2), un método para evaluación automática de habilidad de canto (sección 2.3), un método para procesar automáticamente el timbre de la voz y producir variaciones realistas de intensidad (sección 2.4) y por último un método para reducir la disonancia en acordes desafinados (sección 2.5).

2.1 Análisis comparativo de estimadores de tono para búsqueda de música por canto o tarareo

En esta sección se resume el contenido de la sección 3.1 de la Tesis completa, que corresponde con la publicación [Molina et al., 2014d], donde se presenta un estudio comparativo de varios estimadores de tono del estado del arte aplicados al contexto de búsqueda musical por canto o tarareo.

2.1.1 Enfoque utilizado

Este estudio se ha llevado a cabo utilizando la base de datos MIR-QBSH [1], que es bien conocida y está disponible públicamente, con diferentes condiciones de ruido ambiental y distorsión. Para el estudio se han evaluado 8 algoritmos:

1. YIN [De Cheveigné and Kawahara, 2002]

[1] http://mirlab.org/dataset/public/

2. pYIN [Mauch, 2014]
3. AC-DEFAULT [Boersma, 1993]
4. AC-ADJUSTED [Boersma, 1993]
5. AC-LEIWANG [Wang et al., 2008]
6. SWIPE' [Camacho and Harris, 2008]
7. MELODIA-MONO [Salamon and Gómez, 2012]
8. MELODIA-POLY [Salamon and Gómez, 2012]

Para la evaluación, se han utilizado tres algoritmos de emparejamiento melódico audio-a-MIDI, dos de los cuales son estado del arte: MusicRadar [Doreso, 2013] y NetEase [Li et al., 2013], y el tercero es un sencillo algoritmo base que utiliza alineamiento temporal dinámico.

Para la evaluación se ha medido la tasa de acierto en búsqueda de música por canto o tarareo usando 189 combinaciones diferentes de estimador de tono, condiciones de ruido y distorsión y algoritmo de emparejamiento melódico. La tasa de acierto se ha medido utilizando la medida Rango Recíproco Medio (Mean Reciprocal Rank o MRR), definido como:

$$\mathrm{MRR} = (1/N) \sum_{i=1}^{N} r_i^{-1} \tag{2.1}$$

where: N = número total de búsquedas
r_i = posición (o rango) de la respuesta correcta

Además, de cada curva de tono se ha obtenido la exactitud media de la estimación de tono $\overline{\mathrm{Acc}_{\mathrm{ov}}}$ con respecto a una referencia corregida manualmente, tal y como se define en [Salamon and Gómez, 2012].

2.1.2 Resultados y discusión

Los resultados obtenidos se presentan en la tabla 2.1:

F0 tracker	Clean dataset	25dB SNR	25 dB SNR + distortion	15dB SNR	15 dB SNR + distortion	5dB SNR	5 dB SNR + distortion
(A)	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.89 / 0.95	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.88 / 0.95
(B)	89 / **0.80** / **0.89** / **0.96**	89 / **0.80** / **0.89** / **0.96**	**88** / **0.80** / **0.88** / **0.95**	88 / **0.79** / **0.88** / **0.94**	84 / 0.71 / 0.86 / 0.94	78 / 0.50 / 0.73 / 0.85	67 / 0.33 / 0.57 / 0.73
(C)	**90** / 0.74 / 0.85 / 0.94	90 / 0.71 / 0.85 / 0.92	86 / 0.72 / 0.84 / 0.92	89 / 0.71 / 0.84 / 0.92	85 / 0.66 / 0.81 / 0.89	72 / 0.49 / 0.58 / 0.70	64 / 0.26 / 0.39 / 0.51
(D)	90 / 0.71 / 0.83 / 0.92	**90** / 0.74 / 0.85 / 0.93	85 / 0.74 / 0.85 / 0.94	**90** / 0.78 / 0.87 / 0.94	**85** / **0.77** / **0.87** / **0.94**	79 / **0.69** / **0.79** / **0.87**	72 / **0.58** / **0.69** / **0.81**
(E)	89 / 0.71 / 0.83 / 0.92	89 / 0.71 / 0.84 / 0.92	84 / 0.66 / 0.80 / 0.91	88 / 0.72 / 0.84 / 0.93	83 / 0.65 / 0.80 / 0.91	75 / 0.67 / 0.67 / 0.82	66 / 0.48 / 0.53 / 0.73
(F)	86 / 0.62 / 0.81 / 0.89	86 / 0.70 / 0.83 / 0.92	81 / 0.64 / 0.78 / 0.89	82 / 0.60 / 0.77 / 0.88	75 / 0.50 / 0.67 / 0.82	48 / 0.03 / 0.08 / 0.04	44 / 0.04 / 0.04 / 0.03
(G)	88 / 0.56 / 0.81 / 0.88	87 / 0.47 / 0.79 / 0.86	83 / 0.47 / 0.76 / 0.85	86 / 0.39 / 0.78 / 0.87	81 / 0.35 / 0.73 / 0.82	70 / 0.11 / 0.32 / 0.52	63 / 0.04 / 0.20 / 0.38
(H)	87 / 0.66 / 0.83 / 0.87	87 / 0.67 / 0.82 / 0.87	83 / 0.64 / 0.78 / 0.84	86 / 0.66 / 0.81 / 0.84	82 / 0.58 / 0.74 / 0.80	83 / 0.51 / 0.73 / 0.75	73 / 0.32 / 0.55 / 0.62
(I)	84 / 0.62 / 0.76 / 0.86	84 / 0.62 / 0.76 / 0.86	79 / 0.50 / 0.64 / 0.74	84 / 0.63 / 0.76 / 0.86	79 / 0.50 / 0.65 / 0.75	**83** / 0.60 / 0.73 / 0.83	**75** / 0.39 / 0.55 / 0.65

Table 2.1: Exactitud media de estimación de tono y MRR obtenido para cada caso. Estimadores de tono: (A) *REFERENCIA CORREGIDA MANUALMENTE* (B) *AC-LEIWANG* (C) *AC-ADJUSTED* (D) *PYIN* (E) *SWIPE'* (F) *YIN* (G) *AC-DEFAULT* (H) *MELODIA-MONO* (I) *MELODIA-POLY*. El formato de cada celda es: $\overline{\text{Acc}_{\text{ov}}}(\%)$ / *MRR-algoritmo base* / *MRR-NetEase* / *MRR-MusicRadar*.

Los resultados demuestran que el método basado en autocorrelación de [Boersma, 1993] (con un ajuste no obvio de parámetros) y pYIN [Mauch, 2014], tienen muy buen comportamiento en el contexto de búsqueda musical por canto o tarareo, dada la continuidad y suavidad de los contornos de tono extraídos.

2.2 Transcripción de voz cantada a notas

Esta sección se resume el contenido de la sección 3.2 de la Tesis completa, que corresponde a las publicaciones [Molina et al., 2015] y [Molina et al., 2014b], donde se presenta un nuevo método (llamado SiPTH) para transcripción de voz cantada y una herramienta de evaluación en Matlab.

2.2.1 SiPTH

El enfoque propuesto [Molina et al., 2015] implementa segmentación a nivel de nota basada en intervalos mediante un proceso de histéresis definito en la curva tono-tiempo, la cual es obtenida usando el algoritmo Yin [De Cheveigné and Kawahara, 2002]. Concretamente, el algoritmo consta de varios pasos:

1. Estimación de contornos de croma: Primero se extraen regiones con contornos de croma estables.

2. Clasificación en segmentos vocales/no-vocales: Los contornos de croma se clasifican en vocales / no-vocales.

3. Transcripción basada en intervalos: Un promediado dinámico de la curva de tono se lleva a cabo, y se utiliza la desviación instantánea de la curva con respecto a esta para determinar los cambios de nota. Para establecer un cambio de nota se aplica un proceso de histéresis que favorece que sólo se contabilicen cambios de tono importantes, o sostenidos en el tiempo.

4. Etiquetado de notas: Finalmente cada nota es etiquetada con tres valores: inicio, fin, y frecuencia.

La evauación del algoritmo se ha llevado a cabo utilizando el marco de evaluación propuesto, que se presenta en la siguiente sección.

2.2.2 Herramienta de evaluación para transcripción de voz cantada

Dada la ausencia de una metodología de evaluación standard para transcripción de voz cantada, en esta Tesis se presenta un marco de evaluación que consiste de una

base de datos anotada de 1554 segundos, y un software en Matlab (con interfaz gráfica) capaz de llevar a cabo una evaluación detallada de la transcripción a ser evaluada [Molina et al., 2014b].

2.2.2.1 Colección de datos propuesta

La colección de datos propuesta consta de 38 melodías cantadas por adultos y niños aficionados (formato mono, 16 bits a 44100Hz con cierto ruido ambiental), que en total suman 1154 segundos. La anotación se ha llevado a cabo manualmente por un músico experto, con una posterior revisión por otro músico experto diferente.

2.2.2.2 Medidas de evaluación propuestas

En el marco de evaluación propuesto se han incluido una serie de métricas de evaluación útiles para entender el comportamiento del transcriptor analizado. Algunas de estas medidas han sido utilizadas previamente en MIREX [2]:

- Inicio, fin y tono de nota correctos (COnPOff)
- Inicio y tono de nota correctos (COnp)
- Inicio de nota correcto (COn)

Además, se han incorporado métricas propias sobre el tipo de errores cometidos:

- Tasa de notas con sólo inicio incorrecto (OBOn)
- Tasa de notas con sólo tono incorrecto (OBP)
- Tasa de notas con sólo fin incorrecto (OBOff)
- Tasa de notas divididas innecesariamente en varias notas (S)
- Tasa de notas unificadas indebidamente en una sola nota (M)
- Tasa de notas espúreas (PU)
- Tasa de notas no detectadas (ND)

[2] http://www.music-ir.org/mirex/wiki/MIREX_HOME

2.2.2.3 Resultados y discusión

En esta sección se proporcionan los resultados de la evaluación del método SiPTH junto a varios métodos del estado del arte, usando el marco de evaluación descrito anteriormente.

Algoritmos analizados:

- SiPTH [Molina et al., 2015]
- [Gómez et al., 2013]
- [Ryynänen, 2008]
- Melotranscript [3]
- Algoritmo de base muy sencillo

Los resultados obtenidos se muestran en la figura 2.1.

La primera observación es que ninguno de los métodos analizados tienen un muy buen comportamiento. En efecto, el valor-F más alto en la medida COnPOff (inicio, fin y tono de nota correcto) es menor de 0.5, por lo que el problema de transcripción de voz cantada aún está lejos de ser resuelto. En cualquier caso, el sistema que ofrece mejores resultados es Melotranscript, seguido por SiPTH [Molina et al., 2015] y [Gómez et al., 2013], que tienen un comportamiento similar. Finalmente [Ryynänen, 2008] tiene una precisión menor, probablemente debido al uso de valores de tono enteros para la transcripción (como sugiere [Mauch et al., 2015]).

[3] https://www.samplesumo.com/melody-transcription

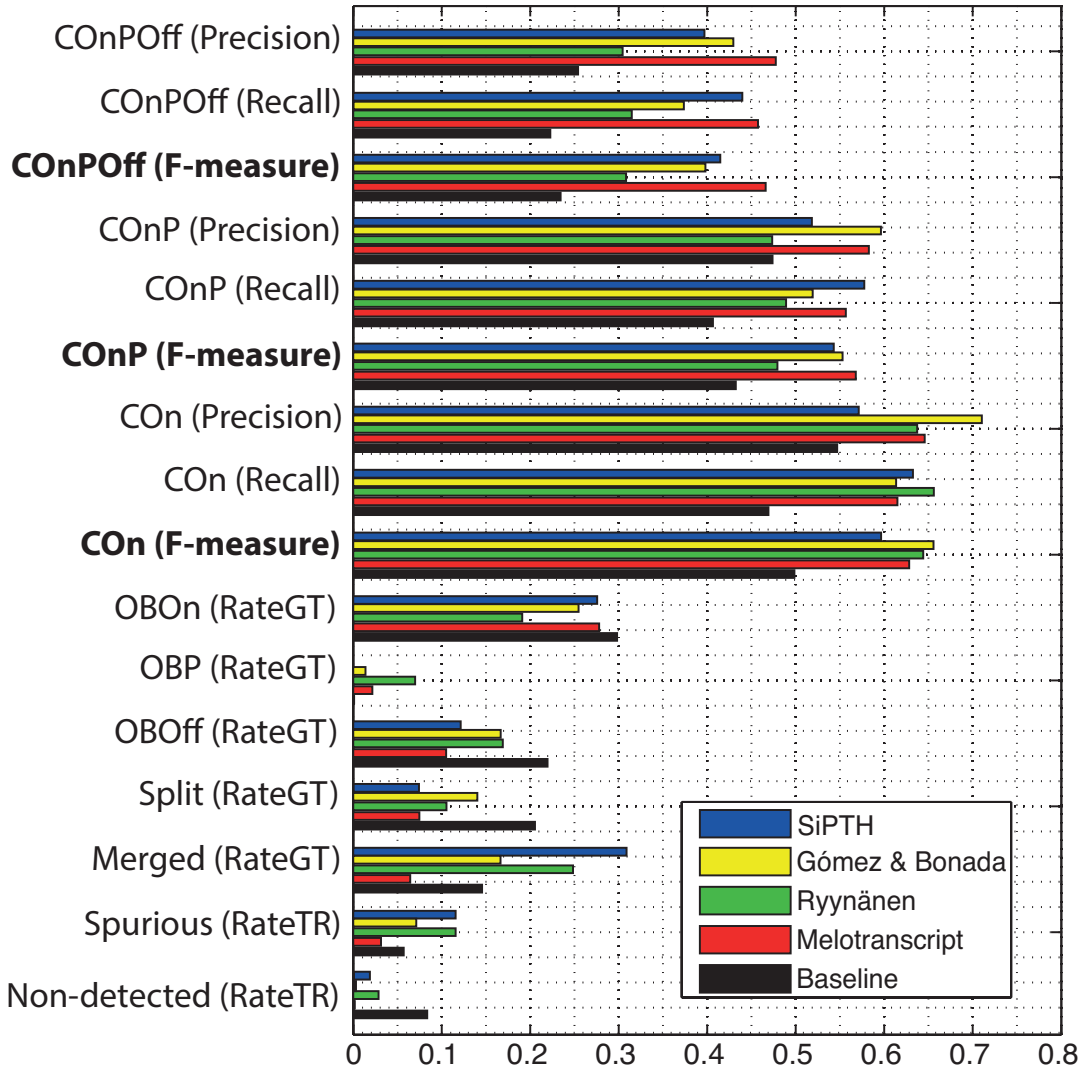

Figure 2.1: Comparación entre métodos del estado del arte para transcripción de voz cantada utilizando la herramienta de evaluación propuesta.

2.3 Evaluación automática de la habilidad de canto

En esta sección se resumen el contenido de la sección 3.3 de la Tesis completa, que corresponde a la publicación [Molina et al., 2013], donde se exploran dos variantes para la evaluación automática de la interpretación de voz cantada: similitud de las curvas de tono utilizando alineamiento temporal dinámico, y similitud a nivel de

nota utilizando transcripción.

2.3.1 Descripción del sistema

2.3.1.1 Enfoque basado en similitud de curvas de tono

Las curvas de tono de la interpretación vocal y de una referencia es extraido usando el algoritmo Yin [De Cheveigné and Kawahara, 2002]. En estas curvas, se asigna f0 = 0 a las ventanas temporales correspondientes a sonidos no vocálicos. Posteriormente, se utiliza alineamiento temporal dinámico [Hiroaki, 1978] para encontrar el alineamiento óptimo entre la interpretación a evaluar y la referencia. La base del enfoque propuesto se basa en la idea de que el camino óptimo de alineamiento ofrece información sobre entonación y ritmo. El error total acumulado en el camino óptimo se relaciona con el error de entonación, y la irregularidad del camino óptimo a lo largo de la matrix de coste se asocia a errores rítmicos.

2.3.1.2 Enfoque basado en similitud a nivel de nota

La interpretación a evaluar es transcrita a nivel de nota usando el transcriptor SiPTH (descrito en sección 2.2.1). Posteriormente, se aplica alineamiento temporal dinámico para encontrar la correspondencia entre cada nota de la interpretación a evaluar y la referencia. Disponiendo de esta información, se utiliza la desviación en los inicios de nota como descriptor de precisión rítmica, y las desviaciones de tono en las notas como descriptor de precisión de entonación.

2.3.1.3 Cálculo de la puntuación

Ambos enfoques utilizan la información disponible para ofrecer tres medidas: precisión rítmica, precisión de entonación y precisión global. Para ello, se utiliza un sistema de regresión polinomial de orden dos entrenado con puntuaciones ofrecidas por músicos expertos sobre una colección de 27 melodías (22 minutos de audio).

2.3.2 Evaluación y resultados

Para evaluación se han calculado las siguientes métricas utilizando el dataset mencionado:

- Confianza interjuicio de los músicos expertos
- Correlación entre las puntuaciones automáticas y las ofrecidas por los músicos
- Error de regresión polinomial

En table 2.2, 2.3 y 2.4 se muestran los errores obtenidos.

Tipo de puntuación	**Coeficiente de correlación medio**
Entonación	0.93
Ritmo	0.82
General	0.90

Table 2.2: Confianza interjuicio de los músicos expertos

Medida de similitud	**Corr. con puntuación entonación**	**Corr. con puntuación ritmo**	**Corr. co puntuación general**
TIE	0.92	0.21	0.81
ε_{RMS}	0.0012	0.81	0.52
$\overline{\Delta O}$	0.026	0.68	0.48
$\overline{\Delta O}_W$	0.037	0.68	0.48
$\overline{\Delta f}$	0.96	0.2	0.82
$\overline{\Delta f}_W$	0.89	0.23	0.82
$\overline{\Delta I}$	0.94	0.34	0.9
$\overline{\Delta I}_W$	0.87	0.35	0.87

Table 2.3: Valores de correlación de cada medida de similitud con las puntuaciones proporcionadas por músicos expertos.

Tipo de error	**Entonación**	**Ritmo**	**General**
Coeficiente de correlación	0.988	0.969	0.976
Raíz de error cuadrático medio	0.4167	0.58	0.44

Table 2.4: Error de regresión polinomial

Como conclusión, el enfoque propuesto modela adecuadamente el criterio de músicos expertos y ofrece una serie de puntuaciones de las cuales, la que más confianza ofrece, es la puntuación de entonación. Como aspecto negativo, el sistema no es capaz de indicar dónde están los errores, sino sólamente una puntuación para toda la interpretación.

2.4 Análisis y procesado de timbre

En esta sección se resumen el contenido de la sección 3.4 de la Tesis completa, que corresponde a la publicación [Molina et al., 2014c], donde se describe un método para modelar las variaciones de la envolvente espectral en función de la intensidad de la voz cantada.

2.4.1 Procedimiento

La investigación se ha llevado a cabo en varios pasos:

1. Definición de un modelo paramétrico de envolvente espectral: Se propone un modelo paramétrico que utiliza filtros de 4o orden para modelar formantes, además de otros parámetros para modelas la pendiente, etc. En total se utilizan 12 parámetros.

2. Anotación manual de 60 vocales de voz cantada: Utilizando una herramienta software especialmente diseñada para ello, se han anotado los parámetros de la envolvete espectral de 60 vocales cantadas en diferentes intensidades. Los parámetros varían con respecto a la intensidad según se muestra en la figura 2.2.

3. Modelado de la variación de parámetros en función de la intensidad: Utilizando un modelo de regresión lineal, se ha modelado la variación de cada parámetro del modelo en función de la intensidad: $\Delta p_x = \Delta I \cdot w_x$ donde w_x es el peso obtenido mediante regresión lineal sobre la colección de vocales descrita en el paso anterior.

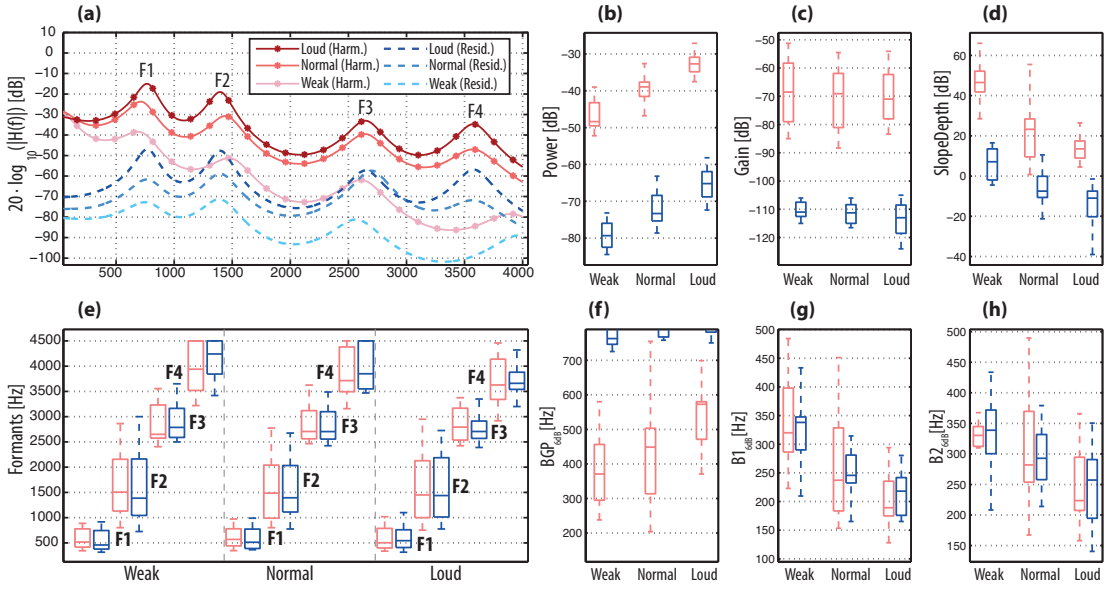

Figure 2.2: Información sobre la componente harmónica (color rojo claro) y la componente residual (color azul oscuro) con diferentes niveles de intensidad. **(a)** Envolvente espectral de una vocal /a/ cantada por un cantante masculino **(b)** Potencia media **(c)** Ganancia media **(d)** Pendiente media **(e)** Frecuencia media de los primeros cuatro formantes **(f)** Ancho de banda medio del resonador glotal R_{GP} **(g)** Ancho de banda medio del primer formante R_1 **(h)** Ancho de banda medio del segundo formante R_2

2.4.2 Evaluación

Para evaluación se han utilizado 12 pares de vocales cantadas débil-fuerte cantadas por cantantes masculinos y femeninos. De estos 12 pares, 4 han sido sintetizados utilizando el software Vocaloid.

Utilizando una variación de intensidad de +/-10, se ha comparado el resultado obtenido en las transformaciones débil-a-fuerte y fuerte-a-débil con el producido por Melodyne Editor y por Vocaloid (para el caso de las vocales sintéticas). Esto da un lugar a un total de 48 pares a evaluar.

Posteriormente, cuatro músicos aficionados han escuchado el resultado de cada transformación, indicando en un cuestionario cómo de similar a un cambio real de intensidad han escuchado el procesado resultante.

2.4.3 Resultados y discusión

En la figura 2.3 se muestra el resultado de los cuestionarios siguiendo el método de evaluación descrito.

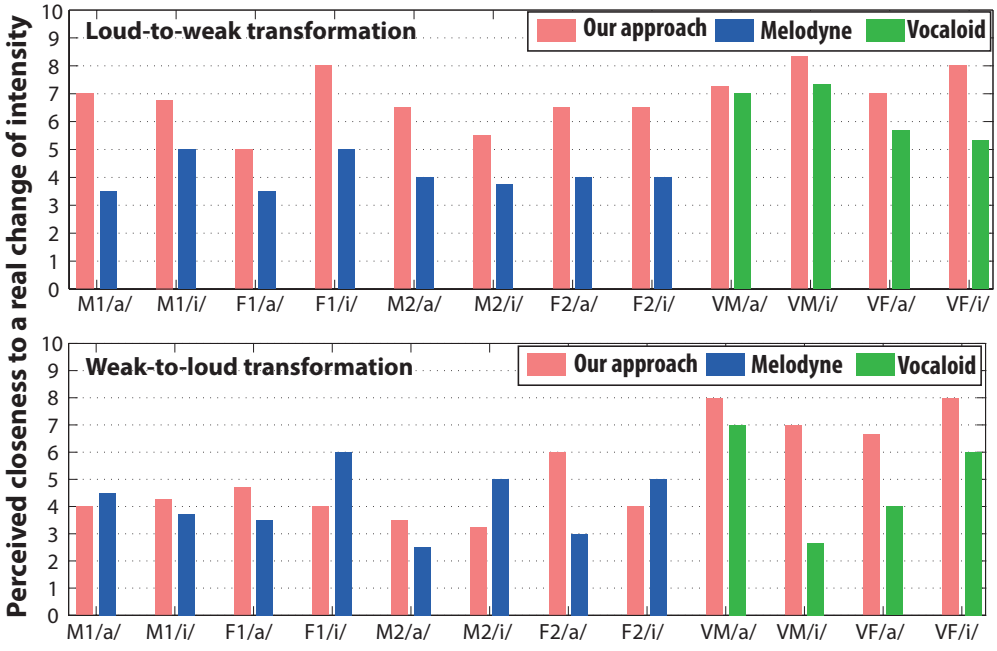

Figure 2.3: Similitud percibida media a un cambio real de intensidad. Cada combinación de cantante / vocal se ha evaluado con varios métodos, mostrados con diferentes colores.

Como conclusión, el método propuesto ofrece resultados mejores que Melodyne Editor y Vocaloid para manipular la intensidad de voz cantada, obteniendo resultados especialmente buenos para transformaciones fuerte-a-débil. Esto puede deberse a que las vocales fuertes requieren formantes bien definidos, y cualquier error en este proceso es fácilmente percibido. La desventaja del método propuesto es que requiere intervención manual para procesar los audios, pero provee de observaciones prometedoras para conseguir un sistema más práctico en futuras investigaciones.

2.5 Reducción de disonancia en audio polifónico

En esta sección se resume el contenido de la sección 3.5 de la Tesis completa, que corresponde con la publicación [Molina et al., 2014a], donde se propone un método para la reducción automática de disonancias en acordes aislados. El enfoque propuesto se basa en un esquema análisis-resíntesis, y se divide en tres bloques: análisis, reorganización harmónica y síntesis.

La etapa de análisis realiza un modelado sinusoidal-más-residual de la señal musical, para poder manipular la componente sinusoidal del acorde sin afectar al resto del sonido. En esta etapa, además, se utiliza un algoritmo para estimación de múltiples f0s, ya que dicha información se usa en etapas posteriores para determinar la versión consonante del acorde de entrada.

A continuación, las múltiples f0s estimadas se desplazan hasta el acorde consonante más cercano (según unos criterios musicales parametrizables por el usuario) y se recalculan las frecuencias de cada componente sinusoidal de este nuevo acorde afinado. Posteriormente, cada componente sinusoidal del acorde disonance se desplaza para ajustarse a las frecuencias del nuevo acorde afinado.

Además de esta reorganización harmónica, se aplica una reducción de batidos de amplitud y frecuencia de cada componente sinusoidal, ya que es uno de los efectos que tiene la suma de sinusoides con amplitudes y frecuencias similares pero no iguales (algo frecuente en acordes disonantes). Esta reducción se basa en la reducción del rizado en la envolvente temporal de la component sinusoidal, así como de la estabilización de la frecuencia de la componente.

Por último, las componentes sinusoidales desplazadas se resintetizan y se combinan con la componente residual inicialmente estimada. En la figura 2.4 se muestran los diferentes pasos del proceso.

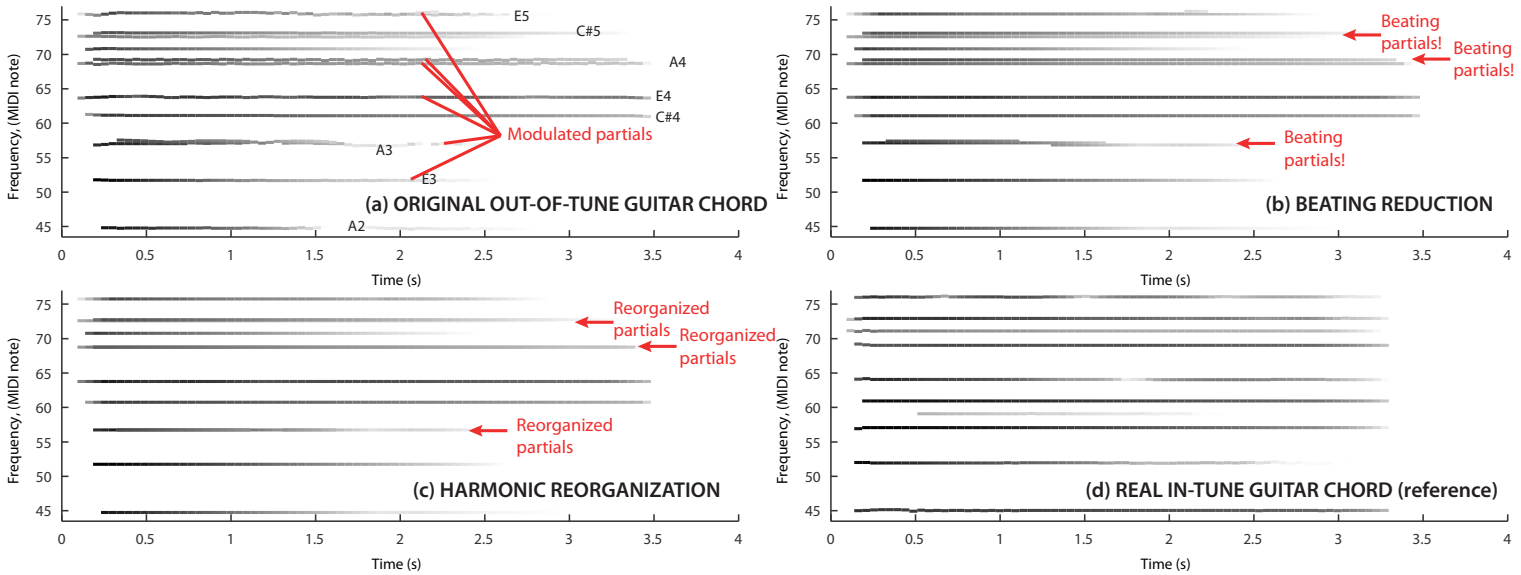

Figure 2.4: Espectrogramas de picos de frecuencias en varias etapas del sistema para un acorde desafinado La mayor tocado con una guitarra acústica. **(a)** Acorde original **(b)** Acorde tras etapa de reducción de batidos **(c)** Acorde tras etapa de reducción de batidos y reorganización harmónica. **(d)** Acorde de La mayor tocado con una guitarra afinada real.

2.5.1 Evaluación

La evaluación publicada en [Molina et al., 2015] se ha llevado a cabo mediante cuestionarios que han sido respondidos por 31 músicos expertos, que han puntuado la consonancia percibida de 18 acordes instrumentales. Además, en esta Tesis se ha llevado a cabo una evaluación extra de 9 acordes vocales cantados por un cuarteto de barbería. En este caso la consonancia percibida ha sido evaluada por 12 músicos expertos.

Los acordes instrumentales evaluados en el cuestionario son:

1. Do mayor tocado con 6 tonos sintéticos complejos. Notas: Do4, Mi4 + 11 cents, Sol4 - 21 cents, Do5 + 30 cents.

2. Do menor tocado con 6 tonos sintéticos complejos. Notas: Do4, Mib4 + 13 cents, Sol4 + 17 cents, Do5 - 32 cents.

3. La mayor tocado con una guitarra acústica real. Notas: La2 - 33 cents, Mi2 - 33 cents, La3 + 27cents, Do#4 + 16 cents, Mi4 - 20 cents.

4. Re mayor tocado con una guitarra acústica real. Notas: Re3 - 30 cents, La3 + 28 cents, Re4 + 15 cents, Fa#4 + 3 cents.

5. Sib mayor tocado con un cuarteto de viento real. Notas: Sib2, Fa3 - 44 cents, Sib3 - 50 cents, Re5 + 31 cents.

6. Do mayor tocado con un cuarteto de cuerda real. Notas: Do3 - 6 cents, Mi3 - 7 cents, Do4 + 30 cents, Sol4 - 73 cents.

Los acordes vocales evaluados en el cuestionario son:

1. Reb mayor cantado por un cuarteto de barbería. Notas: Reb2 - 18 cents, Lab2 - 44 cents, Reb3 + 31 cents, Fa3 + 35 cents.

2. Mib mayor cantado por un cuarteto de barbería. Notas: Mib2 - 58 cents, Sol2, Sib2 + 45 cents, Mib3 + 52 cents.

3. Reb mayor (registro alto) cantado por un cuarteto de barbería. Notas: Reb3 - 58 cents, Fa3 + 45 cents, Ab3 + 52 cents, Reb4.

De cada uno de estos acordes se han evaluado tres versiones diferentes:

a) Acorde sin procesar

b) Procesado con el enfoque propuesto

c) Procesado con Melodyne Editor.

Tal y como se ha comentado, un grupo de músicos expertos ha puntuado las siguientes cuestiones:

- Consonancia media percibida μ_c
- Desviación estándar de la consonancia percibida σ_c
- Cantidad de veces que este acorde ha sido elegido como la mejor opción para un contexto musical.

Las respuestas de los cuestionarios se han promediado para el grupo de músicos entrevistados, obteniendo tres valores por cada uno de los audios:

2.5.2 Resultados y discusión

Los resultados obtenidos se muestran en las tablas 2.5 y 2.6.

Los resultados muestran que el enfoque propuesto mejora sustancialmente la consonancia percibida con respecto a los acordes sin procesar, y obtiene mejores resultados que Melodyne Editor para la mayoría de casos. Especialmente notable es el caso de la guitarra acústica, por el buen resultado obtenido en este instrumento tan habitual en el mundo de la producción musical moderna. En el caso del cuarteto de viento, Melodyne tiene un mejor comportamiento que el enfoque propuesto debido a que la reducción de batidos disminuye notablemente la naturalidad del sonido final. En los casos 7 y 9, sin embargo, Melodyne no consigue estimar correctamente las frecuencias del acorde y no consigue resolver correctamente la disonancia, obteniendo una puntuación mucho peor que el enfoque propuesto. En este sentido, el enfoque propuesto ofrece la gran ventaja de ser robusto a fallos leves en la detección de las F0 del acorde.

Acorde version	***Consonancia percibida [1-10]***	***Escogido como mejor resultado***
1.A Original	$\mu_c = 3.48$ $\sigma_c = 1.48$	3.2%
1.B Enfoque propuesto	$\mu_\mathbf{c} = \mathbf{6.64}$ $\sigma_\mathbf{c} = \mathbf{2.05}$	**77.4%**
1.C Melodyne	$\mu_c = 5.48$ $\sigma_c = 1.80$	19.35%
2.A Original	$\mu_c = 2.67$ $\sigma_c = 1.30$	6.45%
2.B Enfoque propuesto	$\mu_\mathbf{c} = \mathbf{5.35}$ $\sigma_\mathbf{c} = \mathbf{2.25}$	**74.2%**
2.C Melodyne	$\mu_c = 3.96$ $\sigma_c = 1.87$	19.3%
3.A Original	$\mu_c = 4.61$ $\sigma_c = 1.89$	3.2%
3.B Enfoque propuesto	$\mu_\mathbf{c} = \mathbf{7.19}$ $\sigma_\mathbf{c} = \mathbf{1.86}$	**83.9%**
3.C Melodyne	$\mu_c = 5.83$ $\sigma_c = 2.35$	9.7%
4.A Original	$\mu_c = 4.32$ $\sigma_c = 1.81$	3.2%
4.B Enfoque propuesto	$\mu_\mathbf{c} = \mathbf{7.09}$ $\sigma_\mathbf{c} = \mathbf{1.68}$	**71%**
4.C Melodyne	$\mu_c = 6.19$ $\sigma_c = 1.99$	25.8%
5.A Original	$\mu_c = 2.19$ $\sigma_c = 1.27$	0%
5.B Enfoque propuesto	$\mu_\mathbf{c} = \mathbf{4.03}$ $\sigma_\mathbf{c} = \mathbf{2.33}$	**32%**
5.C Melodyne	$\mu_c = 4.64$ $\sigma_c = 2.38$	68%
6.A Original	$\mu_c = 1.54$ $\sigma_c = 0.80$	0%
6.B Enfoque propuesto	$\mu_\mathbf{c} = \mathbf{5.54}$ $\sigma_\mathbf{c} = \mathbf{2.15}$	**77.4%**
6.C Melodyne	$\mu_c = 4.77$ $\sigma_c = 1.96$	22.6%

Table 2.5: Resultados de cuestionarios para acordes instrumentales. **x.A:** Acorde original; **x.B:** Acorde procesado con enfoque propuesto; **x.C:** Acorde procesado con Melodyne Editor.

Acorde version	***Consonancia percibida [1-10]***	***Escogido como mejor resultado***
7.A Original	$\mu_c = 6.09 \;\; \sigma_c = 1.7$	0%
7.B Enfoque propuesto	$\mu_{\mathbf{c}} = \mathbf{8.36} \;\; \sigma_{\mathbf{c}} = \mathbf{1.12}$	**100%**
7.C Melodyne	$\mu_c = 4.54 \;\; \sigma_c = 1.81$	0%
8.A Original	$\mu_c = 3.90 \;\; \sigma_c = 1.22$	0%
8.B Enfoque propuesto	$\mu_{\mathbf{c}} = \mathbf{7.09} \;\; \sigma_{\mathbf{c}} = \mathbf{1.04}$	**36%**
8.C Melodyne	$\mu_c = 6.63 \;\; \sigma_c = 1.02$	64 %
9.A Original	$\mu_c = 3.36 \;\; \sigma_c = 1.62$	0%
9.B Enfoque propuesto	$\mu_{\mathbf{c}} = \mathbf{6.0} \;\; \sigma_{\mathbf{c}} = \mathbf{2.04}$	**73%**
9.C Melodyne	$\mu_c = 3.63 \;\; \sigma_c = 2.37$	27%

Table 2.6: Resultados de cuestionarios para acordes vocales. **x.A:** Acorde original; **x.B:** Acorde procesado con enfoque propuesto; **x.C:** Acorde procesado con Melodyne Editor.

Sección 3

Conclusiones y líneas futuras

En esta sección, se presentan algunas conclusiones sobre el trabajo expuesto a lo largo de esta Tesis, y se remarcan los aspectos más importantes de los resultados obtenidos. Además, se enumeran las contribuciones de esta Tesis (sección 3.1) y se exponen algunas sugerencias para continuar con esta línea de investigación en el futuro (sección 3.2).

En esta Tesis, se ha propuesto un conjunto variado de técnicas y aplicaciones en el ámbito del procesado de voz cantada. Específicamente, los objetivos presentados al inicio de esta disertación cubren los tres temas siguientes: transcripción de voz cantada (a nivel de nota, y de curva de tono), evaluación de la interpretación vocal, y transformación de sonido (concretamente: procesado del timbre vocal, y modificación de tono en audio polifónico). Por supuesto, el éxito en estos objetivos pasa por un estudio profundo de las tecnologías existentes y del estado del arte en cada uno de los temas asociados.

Estudio del estado del arte

En primer lugar, se ha presentado un estudio del estado del arte cubriendo todos los temas abordados en esta Tesis (capítulo 2 de la Tesis completa). En él se refleja el conocimiento adquirido durante nuestra investigación, y es útil para contextualizar los resultados obtenidos. Los temas cubiertos por este estudio son: producción de voz cantada, estimación de tono (monofónico, extracción de melodía y en polifonía), transcripción de voz cantada, alineamiento temporal dinámico, evaluación automática de interpretación vocal, procesado de timbre y síntesis basada en modelado espectral.

Análisis comparativo de estimadores de tono para sistemas de búsqueda musical por canto o tarareo

En la sección 3.1 de la Tesis completa, se ha presentado un estudio comparativo de varios estimadores de tono del estado del arte en el contexto de la búsqueda de música por canto o tarareo. Específicamente, se han comparado 8 de los mejores estimadores de tono existentes en 2 de los mejores sistemas de emparejamiento melódico existentes, así como en un sencillo sistema open source. Tres conclusiones se han obtenido de este estudio. Primero, los tres sistemas de emparejamiento melódico obtienen los mejores resultados con los mismos estimadores de tono. Esto sugiere que un sistema simple de emparejamiento melódico puede usarse con éxito para comparar la bondad de un estimador de tono para este caso de uso. Segundo, que el método pYIN para estimación de tono [Mauch, 2014] tiene un sorprendente buen comportamiento en entornos ruidosos. Por último, que la forma en la que los estimadores de tono son evaluados en la literatura no es totalmente representativa de su bondad para su uso en búsqueda musical por canto o tarareo. Esto es debido a que no sólo importa la cantidad de errores cometidos en la estimación, sino la naturaleza de estos errores.

Transcripción de voz cantada

En esta Tesis, se ha propuesto un sistema de transcripción de voz cantada basado en un proceso de histéresis definido en la curva tono-tiempo (sección 3.2.1 de la Tesis completa). Este método aplica un transformación basada en histéresis a las salidas del algoritmo Yin: F0, aperiodicidad y energía, para convertirlas en una secuencia de notas. Los resultados demuestran que este enfoque, que es simple de comprender e implementar, consigue una precisión comparable a otros métodos del estado del arte más complejos.

Además, se ha presentado una aplicación en Matlab para la evaluación de algoritmos de transcipción de voz cantada (sección 3.2.2 de la Tesis completa). Esta aplicación permite visualizar detalles sobre la transcripción, computar métricas de evaluación, y además incluye un base de datos anotada manualmente. Las métricas de evaluación incluidas en esta aplicación reporta información detallada acerca del tipo de errores cometidos por el transcriptor estudiado. Esta aplicación se ha utilizado en varios artículos recientes sobre transcripción de voz cantada (e.g. [Mauch et al., 2015]), y contribuye a la publicación de resultados reproducibles en el ámbito de la transcripción de voz cantada.

Evaluación automática de la interpretación vocal

En la sección 3.3 de la Tesis completa, se han propuesto y comparado dos enfoques diferentes para evaluación automática de interpretación vocal: (1) uso de alineamiento temporal dinámico para comparar la curva de tono de la interpretación del usuario y una referencia (e.g. partitura), (2) medida de similitud a nivel de notas usando transcripción de voz cantada. Ambos enfoques requieren una referencia, la cual se considera la interpretación ideal. Esta intepretación ideal puede ser un fichero MIDI de la canción original, o una interpretación de un músico de referencia (e.g. profesor). El sistema se ha evaluado analizando la correlación entre la puntuación proporcionada por el sistema, y la puntuación proporcionada por músicos expertos. Los resultados de esta comparación demuestran que la comparación de la curva de tono utilizando alineamiento temporal dinámico es un método sencillo y efectivo para la evaluación de interpretación de tono y de ritmo, y que el uso de transcripción a nivel de nota introduce complejidad sin una clara ventaja.

Análisis y procesado de timbre

En la sección 3.4. de la Tesis completa, se ha propuesto un método para modelar las variaciones de la envolvente espectral en función de la intensidad en voz cantada. Este método se basa en un modelo paramétrico de la envolvente espectral, cuyos parámetros se ajustan automáticamente para simular variaciones realistas de intensidad en voz cantada. Tres son las contribuciones principales de esta investigación: (1) un modelo paraémtrico de la envolvente espectral basada en filtros de 4 polos para modelar formantes, (2) una herramienta software para notar vocales cantadas usando dicho modelo paramétrico, y $(\cdot)$ un método para manipular los parámetros de dicho modelo paramétrico automáticamente con el fin de producir variaciones realistas de intensidad. Se ha observado que dos parámetros son los principales responsables en la percepción de intensidad: pendiente de la envolvente espectral, y ancho de banda de los formates. El método propuesto se ha comparado contra Melodyne Editor y Vocaloid 3.0, mediante un cuestionario contestado por cuatro músicos aficionados. Los resultados demuestran que el método propuesto mejora significativamente el realismo de las transformaciones en comparaciones con los otros dos métodos, especialmente para el caso de transformaciones debil-a-fuerte.

Reducción de disonancia en audio polifónico

Por último, en sección 3.5. de la Tesis completa se ha propuesto un método para la reducción automática de disonancia en acordes aislados. Este método realiza una estimación de múltiples F0 para identificar el acorde a ser afinado, y realiza un modelado sinusoidal-más-residual para desplazar la frecuencia de sus parciales.

Estos parciales se desplazan para ajustarse a la estructura armónica de la versión afinada del mismo acorde. La metodología de evaluación se ha basado en tests de audición donde un conjunto de músicos expertos han evaluado la consonancia percibida para un conjunto de acordes antes y después de ser procesados. Los resultados obtenidos demuestran que el método propuesto se comporta en general mejor que Melodyne Editor para mejorar la consonancia de acordes desafinados, en ambos casos, acordes instrumentales y vocales.

3.1 Resumen de contribuciones

En esta sección, se enumeran las contribuciones científicas de esta tesis, junto a los recursos de investigación publicados a lo largo de nuestra investigación.

Contribuciones científicas

- **Estudio de investigación previa:** En el capítulo 2 de la Tesis completa, se proporciona un completo repaso del estado del arte acerca de los métodos y técnicas existentes para procesado de voz cantada. Este repaso cubre los temas siguientes (todas las referencias de secciones se refieren a la Tesis completa): producción de voz cantada (sección 2.1), estimación de tono (sección 2.2), transcripción de voz cantada (sección 2.3), alineamiento temporal dinámico (sección 2.4), evaluación automática de la interpretación vocal (sección 2.5), procesado de timbre (sección 2.6) y síntesis basado en modelado espectral (sección 2.7).

- **Análisis comparativo de estimadores de tono:** Se ha llevado a cabo un estudio comparativo de varios de los mejores estimadores de tono existentes en el contexto de recuperación de música por tarareo. Este estudio ha sido publicado en [Molina et al., 2014d], y ha sido expuesto en la sección 3.1 de la Tesis completa.

- **Método para transcripción de voz cantada:** Un método para transcripción de voz cantada (llamado *SiPTH*) basado en un ciclo de histéresis en la curva tiempo-tono para segmentación a nivel de nota por intervalos. Este método es simple de implementar, y su precisión es cercana a otros métodos del estado del arte. Ha sido publicado en [Molina et al., 2015], y resumido en la sección 3.2.1 de la Tesis completa.

- **Marco de evaluación para transcripción de voz cantada:** Se ha presentado una comparación de las metodologías de evaluación utilizas en trabajos previos sobre transcripción de voz cantada, y se ha propuesto un marco de

evaluación (base de datos anotada y aplicación Matlab). Se ha publicado en [Molina et al., 2014b] y se ha expuesto en la sección 3.2.2 de la Tesis completa.

- **Método para la evaluación automática de la interpretación vocal:** Se ha propuesto un método para la evaluación automática del canto basado en alineamiento dinámico automático de la curva de tono. Se ha publicado en [Molina et al., 2013] y se ha expuesto en la sección 3.3 de la Tesis completa.

- **Método para procesado de timbre:** Se ha presentado un modelo paramétrico de envolvente espectral basado en filtros de 4 polos para modelado de formates, así como un estudio sobre las variaciones de estos parámetros en función de la intensidad del canto. De esta forma, se ha propouesto un método para realizar transformaciones realistas de intensidad en voz cantada. Este método ha sido publicado [Molina et al., 2014c] y se ha expuesto en la sección 3.4 de la Tesis completa.

- **Método para la reducción de disonancia en música polifónica:** Se ha propuesto un método para la reducción de disonancia en acordes desafinados utilizado reorganización armónica. Se ha publicado en [Molina et al., 2015] y se ha resuimdo en la sección 3.5 de la Tesis completa.

Recursos de investigación

- **Sistema básico para recuperación por tarareo:** Se proporciona un sistema Matlab para recuperación por tarareo basado en alineamiento temporal dinámico. Este sistema puede utilizarse como un punto de partido para empezar a desarrollar sistemás más complejos, o para evaluar la bondad de estimadores de tono para el caso de uso de recuperación por tarareo. Puede encontrarse en el siguiente enlace:

  `www.atic.uma.es/ismir2014qbsh`

- **Herramienta para la anotación de envolvente espectral:** Herramienta matlab (con interfaz gráfica) para anotar los parámetros del modelo de la envolvente espectral en vocales cantadas mantenidas. Más detalles pueden encontrarse en la sección 3.4 de la Tesis completa, y puede ser descargada en el enlace:

  `www.atic.uma.es/icassp2014singing`

- **Herramienta para evaluar algoritmos de transcripción de voz cantada:** Herramienta Matlab (con interfaz gráfica) para visualizar y evaluar transcripciones melódicas, junto a una base de datos anotada consistente en

38 melodías (1154 segundos), cantadas por cantantes adultos y niños amateur, anotados por músicos expertos a nivel de nota. Más detalles pueden encontrarse en la sección 3.2.2 de la Tesis completa, y se puede descargar desde el enlace:

`www.atic.uma.es/ismir2014singing`

3.2 Sugerencias para investigación futura

Además del material publicado, muchas otras observaciones e ideas han aparecido a lo largo de nuestra investigación. Algunas de estas consideraciones merecen la pena ser mencionadas porque podrían solucionar debilidades específicas de los enfoques propuestos, o podrían ser incluso prometedoras alternativas para solucionar los problemas abordados. En esta sección se discuten estas ideas y se proponen sugerencias específicas para investigación futura.

Transcripción de voz cantada

- **Mejora del marco de evaluación:** La utilidad del marco de evaluación propuesto en la sección 3.2.2 de la Tesis completa podría mejorarse añadiendo más datos anotados. La anotación manual puede llevarse a cabo eficientemente utilizando el reciente software Tony [Mauch et al., 2015]. En un momento dado, si la cantidad de datos es suficientemente grande, podría definirse una tarea de transcripción de voz cantada en MIREX[1]. Además, las métricas de evaluación propuestas podrían integrarse en el paquete de python `mir_eval` [Raffel et al., 2014] para hacerlas disponibles es un formato estandarizado. Finalmente, este marco de evaluación podría incluir no sólo métricas indendientes del contexto de uso (e.g. precisión de transcripción de notas), sino otras métricas definidas para contextos de uso específicos (e.g. respondiendo a: ¿cómo de bien funciona tu transcriptor para el caso concreto de recuperación de música por tarareo?).

- **Transcripción de voz cantada utilizando modelos ocultos de Markov (HMM) y descriptores de timbre:** Se propone el uso de HMM con descriptores de timbre (e.g. MFCC) para transcripción de voz cantada. Esta idea se basa en tres hechos principalmente: (1) numerosos sistemas de transcripción de voz hablada se basan en esta tecnología con éxito, (2) el problema de transcripción de voz hablada parece compartir una naturaleza similar al

[1] www.music-ir.org/mirex

problema de transcripción de voz hablada (especialmente cuando hay letra), y (3) algunos sistemas de transcripción de voz cantada ya están exitosamente basados en HMM, aunque no se ha encontrado ningún trabajo que utilice descriptores de timbre en este proceso.

Evaluación de interpretación vocal

- **Alineamiento robusto de curva de tono:** El alineamiento de la curva de tono entre la referencia y la intepretación del usuario es la base del enfoque propuesto para evaluación automática de interpretación vocal. Sin embargo, en ocasiones es difícil conseguir un buen alineamiento, sobre todo cuando el usuario comete muchos errores. Para solucionar este problema, se propone realizar un alineamiento audio-a-audio usando no sólo información de tono, sino también de energía, aperiodicidad, o incluso tímbrica.

- **Evaluación de la interpretación vocal de forma independiente a la canción:** El uso de melodías de referencia tiene una clara desventaja: es necesario preparar mucho material para poder llevar a cabo una evaluación adecuada con una gran cantidad de canciones. Sin embargo, tal y como indica [Nakano et al., 2009], la calidad de una interpretación vocal puede a menudo ser evaluada por un humano de forma independiente a la canción que esté siendo cantada. Inspirado por este hecho, un sistema que sea independiente de la canción cantada puede ser más adecuado para una aplicación comercial realista, por lo que se recomienda explorar esta vía de investigación.

Procesado de timbre

- **Enfoque alternativo para modificación automática de intensidad utilizado adaptación de los polos del modelo LPC:** El método propuesto para modificación automática de intensidad en voz cantada (ver sección 3.4 de la Tesis completa) se basa en un modelo paramétrico de envolvente espectral. Se observó que dos parámetros son clave para la percepción de intensidad: pendiente de la envolvente y ancho de banda de los formantes. En vista de este resultado, se sugiere explorar la transformación de polos del filtro LPC para procesar voz cantada, ya que es computacionalmente más sencillo y puede incluso derivar en aplicaciones a tiempo real.

- **Eliminación de la etapa de seguimiento sinusoidal para procesado de audio polifónico:** Nuestro enfoque para reducción de disonancia en música polifónica (ver sección 3.5 de la Tesis completa) usa modelado sinusoidal más residual, y sigue cada sinusoide a lo largo del tiempo para identificar los parciales del sonido, como propone [Serra, 1989]. Sin embargo, esta etapa de seguimiento es computacionalmente costosa y sus beneficios podrían no merecer la pena, así que se sugiere explorar la eliminación de esta etapa para trabajar a nivel de ventana directamente.

Bibliography

[Boersma, 1993] Boersma, P. (1993). Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. *Proceedings of the Institute of Phonetic Sciences*, 17(1193):97–110. ↑2, ↑10, ↑12

[Bonada and Serra, 2007] Bonada, J. and Serra, X. (2007). Synthesis of the singing voice by performance sampling and spectral models. *IEEE Signal Processing Magazine*, 24(2):67–79. ↑6

[Camacho and Harris, 2008] Camacho, A. and Harris, J. G. (2008). A sawtooth waveform inspired pitch estimator for speech and music. *Journal of the Acoustical Society of America*, 124(3):1638–1652. ↑10

[Cook, 1991] Cook, P. (1991). *Identification of Control Parameters in an Articulatory Vocal Tract Model, with Applications to the Synthesis of Singing*. PhD thesis, Stanford University. ↑6

[Cook, 1996] Cook, P. (1996). Singing voice synthesis: history, current work, and future directions. *Computer Music Journal*, 20(3):38–46. ↑6

[De Cheveigné and Kawahara, 2002] De Cheveigné, A. and Kawahara, H. (2002). YIN, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America*, 111(4):1917. ↑9, ↑12, ↑16

[Dittmar et al., 2010] Dittmar, C., Großmann, H., Cano, E., and Al., E. (2010). Songs2See and GlobalMusic2One: two applied research projects in music information retrieval at Fraunhofer IDMT. In *Proceedings of the 7th International Symposium on Computer Music Modeling and Retrieval (CMMR 2010)*, pages 259 – 272, Málaga, Spain. ↑6

[Doreso, 2013] Doreso (2013). MIREX 2013 QBSH Task: MusicRadar's solution. In *Extended Abstract for MIREX Query by Singing/Humming (QBSH) Task*. ↑10

[Fujihara et al., 2011] Fujihara, H., Goto, M., Ogata, J., and Okuno, H. G. (2011). Lyric synchronizer : Automatic synchronization system between musical audio signals and lyrics. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1252–1261. ↑6

[Gómez et al., 2013] Gómez, E., Bonada, J., and Emilia, G. (2013). Towards computer-assisted flamenco transcription: an experimental comparison of automatic transcription algorithms as applied to a cappella singing. *Computer Music Journal*, 37(2):73–90. ↑14

[Goto et al., 2010] Goto, M., Saitou, T., Nakano, T., and Fujihara, H. (2010). Singing information processing based on singing voice modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2010)*, pages 5506–5509. ↑5

[Goto, 2014] Goto, M. (2014). Singing Information Processing. In *Proceedings of the 12th International Conference on Signal Processing (ICSP 2004)*, pages 7–14, Hangzhou, China. ↑5

[Grollmisch et al., 2011] Grollmisch, S., Cano Cerón, E., and Dittmar, C. (2011). Songs2See: Learn to Play by Playing. In *Proceedings of Audio Engineering Society Conference: 41st International Conference: Audio for Games*. ↑6

[Hiroaki, 1978] Hiroaki, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26(1):43–46. ↑16

[Howard et al., 2004] Howard, D. M., Welch, G., Brereton, J., Himonides, E., Decosta, M., Williams, J., and Howard, A. (2004). WinSingad: a real-time display for the singing studio. *Logopedics Phoniatrics Vocology*, 29(3):135–144. ↑6

[Kan et al., 2008] Kan, M. Y., Wang, Y., Iskandar, D., Nwe, T. L., and Shenoy, A. (2008). LyricAlly: Automatic synchronization of textual lyrics to acoustic music signals. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2):338–349. ↑6

[Kanato et al., 2014] Kanato, A., Nakano, T., Goto, M., and Kikuchi, H. (2014). An automatic singing impression estimation method using factor analysis and multiple regression. In *Proceedings of the Joint International Computer Music Conference and Sound and Music Computing Conference (ICMCSMC2014)*, pages 1244–1251, Athens, Greece. ↑6

[Kenmochi and Ohshita, 2007] Kenmochi, H. and Ohshita, H. (2007). VOCALOID - commercial singing synthesizer based on sample concatenation. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH 2007)*, pages 4009–4010. ↑6

[Li et al., 2008] Li, P., Wang, X., Zhou, M., and Li, N. (2008). A novel MIR system based on improved melody contour definition. In *Proceedings of the International Conference on MultiMedia and Information Technology (MMIT 2008)*, pages 409–412. ↑6

[Li et al., 2013] Li, P., Nie, Y., and Li, X. (2013). Query-by-singing-humming Task : Netease ' S Solution. In *Extended Abstract for MIREX Query by Singing/Humming (QBSH) Task*. ↑10

[Mauch, 2014] Mauch, M. (2014). PYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, pages 659–663. ↑2, ↑10, ↑12, ↑28

[Mauch et al., 2015] Mauch, M., Cannam, C., Bittner, R., Fazekas, G., Salamon, J., Dai, J., Bello, J. P., and Dixon, S. (2015). Computer-aided melody note transcription using the Tony software: accuracy and efficiency. In *Proceedings of the First International Conference on Technologies for Music Notation and Representation (TENOR)*. ↑14, ↑28, ↑32

[Molina, 2012] Molina, E. (2012). *Automatic scoring of singing voice based on melodic similarity measures*. MSc Thesis. Universitat Pompeu Fabra (Barcelona), Barcelona. ↑6

[Molina et al., 2013] Molina, E., Barbancho, I., Gómez, E., Barbancho, A. M., and Tardón, L. J. (2013). Fundamental frequency alignment vs. note-based melodic similarity for singing voice assessment. In *Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, pages 744–748, Vancouver (Canada). ↑7, ↑15, ↑31

[Molina et al., 2014a] Molina, E., Barbancho, A. M., Tardón, L. J., and Barbancho, I. (2014a). Dissonance reduction in polyphonic music using harmonic reorganization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(2):325–334. ↑7, ↑20

[Molina et al., 2014b] Molina, E., Barbancho, A. M., Tardón, L. J., and Barbancho, I. (2014b). Evaluation framework for automatic singing transcription. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 567–572, Taipei (Taiwan). ↑7, ↑12, ↑13, ↑31

[Molina et al., 2014c] Molina, E., Barbancho, I., Barbancho, A. M., and Tardón, L. J. (2014c). Parametric model of spectral envelope to synthesize realistic intensity variations in singing voice. In *Proceedings of the 39th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, pages 634–638, Florence (Italy). ↑7, ↑17, ↑31

[Molina et al., 2014d] Molina, E., Tardón, L. J., Barbancho, I., and Barbancho, A. M. (2014d). The importance of F0 tracking in query-by-singing-humming. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 277–282, Taipei (Taiwan). ↑6, ↑9, ↑30

[Molina et al., 2015] Molina, E., Tardón, L. J., Barbancho, A. M., and Barbancho, I. (2015). SiPTH: Singing transcription based on hysteresis defined on the pitch-time curve. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(2):252–263. ↑6, ↑12, ↑14, ↑22, ↑30, ↑31

[Nakano et al., 2005] Nakano, T., Goto, M., Ogata, J., and Hiraga, Y. (2005). Voice Drummer : A Music Notation Interface of Drum Sounds Using Voice Percussion Input. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*. ↑6

[Nakano et al., 2009] Nakano, T., Goto, M., and Hiraga, Y. (2009). An automatic singing skill evaluation method for unknown melodies using pitch interval accuracy and vibrato features. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH 2009)*, pages 1706–1709. ↑33

[Pardo et al., 2004] Pardo, B., Shifrin, J., and Birmingham, W. (2004). Name that tune: a pilot study in finding a melody from a sung query. *Journal of the American Society for Information Science and Technology*, 55(4):283–300. ↑6

[Raffel et al., 2014] Raffel, C., McFee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D., and Ellis, D. P. W. (2014). mir_eval: A Transparent Implementation of Common MIR Metrics. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 367–372, Taipei, Taiwan. ↑32

[Rossiter and Howard, 1996] Rossiter, D. and Howard, D. M. (1996). ALBERT: a real-time visual feedback computer tool for professional vocal development. *Journal of Voice*, 10(4):321–336. ↑6

[Ryynänen, 2006] Ryynänen, M. (2006). Singing Transcription. In *Signal Processing Methods for Music Transcription*. Springer. ↑6

[Ryynänen, 2008] Ryynänen, M. (2008). *Automatic Transcription of Pitch Content in Music and Selected Applications*. PhD thesis, Tampere University of Technology. ↑14

[Saino et al., 2006] Saino, K., Zen, H., Nankaku, Y., Lee, A., and Tokuda, K. (2006). An HMM-based singing voice synthesis system. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH 2006)*, pages 2274–2277. ↑6

[Salamon and Gómez, 2012] Salamon, J. and Gómez, E. (2012). Melody Extraction From Polyphonic Music Signals Using Pitch Contour Characteristics. *IEEE Transactions on Audio, Speech and Language Processing*, 20(6):1759–1770. ↑10

[Schwarz, 2007] Schwarz, D. (2007). Corpus-based concatenative synthesis. *IEEE Signal Processing Magazine*, 24(2):92–104. ↑6

[Serra, 1989] Serra, X. (1989). *A System for Sound Analysis / Transformation / Synthesis based on a Deterministic plus Stochastic Decomposition*. PhD thesis, Stanford University. ↑34

[Toda et al., 2007] Toda, T., Black, A. W., and Tokuda, K. (2007). Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. *IEEE Transactions on Audio, Speech and Language Processing*, 15(8):2222–2235. ↑6

[Wang et al., 2008] Wang, L., Huang, S., Hu, S., Liang, J., and Xu, B. (2008). An effective and efficient method for query by humming system based on multi-similarity measurement fusion. In *Proceedings of the International Conference on Audio, Language and Image Processing, Proceedings (ICALIP 2008)*, pages 471–475. ↑6, ↑10

[Zhang, 2003] Zhang, T. Z. T. (2003). Automatic singer identification. In *Proceedings of the International Conference on Multimedia and Expo (ICME 2003)*, pages 33–36. ↑6