IMAGE LICENSED BY
INGRAM PUBLISHING

Melody Extraction from Polyphonic Music Signals

[Approaches, applications, and challenges]

[Justin Salamon,
Emilia Gómez, Daniel P.W. Ellis,
and Gaël Richard]

Melody extraction algorithms aim to produce a sequence of frequency values corresponding to the pitch of the dominant melody from a musical recording. Over the past decade, melody extraction has emerged as an active research topic, comprising a large variety of proposed algorithms spanning a wide range of techniques. This article provides an overview of these techniques, the applications for which melody extraction is useful, and the challenges that remain. We start with a discussion of "melody" from both musical and signal processing perspectives and provide a case

Digital Object Identifier 10.1109/MSP.2013.2271648
Date of publication: 12 February 2014

1053-5888/14/$31.00©2014IEEE

study that interprets the output of a melody extraction algorithm for specific excerpts. We then provide a comprehensive comparative analysis of melody extraction algorithms based on the results of an international evaluation campaign. We discuss issues of algorithm design, evaluation, and applications that build upon melody extraction. Finally, we discuss some of the remaining challenges in melody extraction research in terms of algorithmic performance, development, and evaluation methodology.

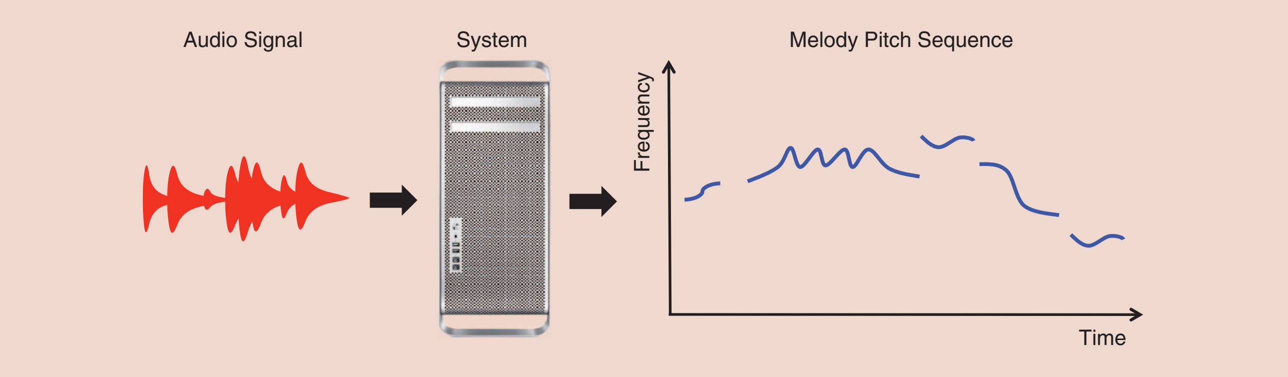
INTRODUCTION

Music was the first mass-market industry to be completely restructured by digital technology starting with the compact disc and leading to today's situation where typical consumers may have access to thousands of tracks stored locally on their smartphone or music player, and millions of tracks instantly available through cloud-based music services. This vast quantity of music demands novel methods of description, indexing, searching, and interaction. Recent advances in audio processing have led to technologies that can help users interact with music by directly analyzing the musical content of audio files. The extraction of melody from polyphonic music signals is such a technology and has received substantial attention from the audio signal processing and music information retrieval (MIR) research communities. Known as *melody extraction, audio melody extraction, predominant melody extraction, predominant melody estimation,* or *predominant fundamental frequency estimation*, the task involves automatically obtaining a sequence of frequency values representing the pitch of the dominant melodic line from recorded music audio signals (Figure 1).

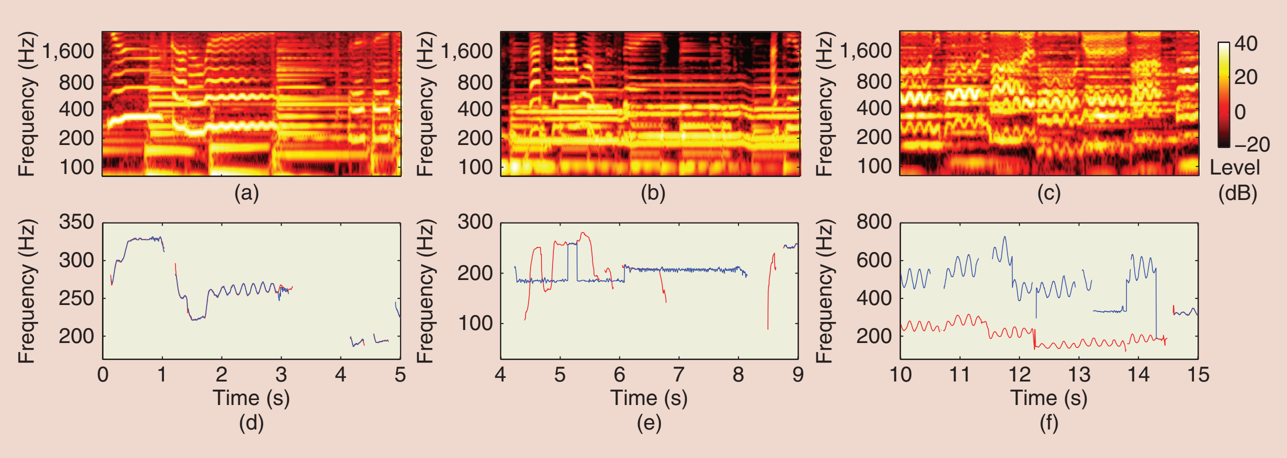
Music transcription, i.e., converting an audio signal into a description of all the notes being played, is a task that can usually be achieved by a trained student of music and has long been a topic of computational research. It has, however, proven to be very difficult due to the complex and deliberately overlapped spectral structure of musical harmonies. In one of the earliest works in the field, Masataka Goto pointed out that many interesting music tasks, such as melody-based retrieval or melody line suppression for karaoke, could be achieved with a much more limited transcription that recovered only a single melody line as the "strongest" pitch in the likely melody range at any time [1]. This idea was picked up by Emilia Gómez, Beesuan Ong, and Sebastian Streich, who put together a melody extraction task as part of the Audio Description Contests associated with the 2004 International Conference on Music Information Retrieval (ISMIR), organized by the Music Technology Group at Pompeu Fabra University, Barcelona [2]. This activity was followed by the Music Information Retrieval Evaluation eXchange (MIREX) evaluation campaign for MIR technologies [3] and has in subsequent years resulted in a series of well-organized international evaluations with broad participation, described in the section "Algorithm Overview: 2005 to Date."

OVER THE PAST DECADE, MELODY EXTRACTION HAS EMERGED AS AN ACTIVE RESEARCH TOPIC, COMPRISING A LARGE VARIETY OF PROPOSED ALGORITHMS SPANNING A WIDE RANGE OF TECHNIQUES.

To frame the technical task of melody extraction, we should start by examining the musicological concept of "melody," which ultimately relies on the judgement of human listeners [2] and will therefore tend to vary across application contexts (e.g., symbolic melodic similarity [4] or music transcription [5]). Centuries of musicological study [6] have resulted in no clear consensus regarding the definition of "melody" but, faced with the need for a common interpretation, the MIR community has opted for simplified, pragmatic definitions that result in a task amenable to signal processing. One popular definition [2] holds that "the melody is the single (monophonic) pitch sequence that a listener might reproduce if asked to whistle or hum a piece of polyphonic music, and that a listener would recognize as being the essence of that music when heard in comparison." This definition is still open to a considerable degree of subjectivity, since different listeners might hum different parts after listening to the same song (e.g., lead vocals versus guitar solo). In practice, research has focused on what we



[FIG1] Melody extraction: obtaining a sequence of frequency values representing the pitch of the melody from the audio signal of polyphonic music.



[FIG2] Case study examples: (a)–(c) show the log-frequency spectrogram of three excerpts in the genres of (a) vocal jazz, (b) pop, and (c) opera. Parts (d)–(f) show the extracted melody [16] (blue) and ground truth (red) for each excerpt, respectively.

term *single source predominant fundamental frequency estimation*. That is, the melody is constrained to belong to a single sound source throughout the piece being analyzed, where this sound source is considered to be the most predominant instrument or voice in the mixture. While the subjective element can not be completely eliminated even in this definition (for instance, how do we define predominant?), the problem is avoided in practice by working with musical material that contains a clear lead singer or instrument. Thus, our modified task definition becomes "single source predominant fundamental frequency estimation from musical content with a lead voice or instrument." While this definition is too limited to encompass everything one might consider as melody, its solution would nonetheless lead to extremely powerful technologies. Note that we have used the term *fundamental frequency* (henceforth f_0) to refer to the physical property most closely related to the perceptual property of pitch [7]. Still, the terms *pitch* and f_0 are often used interchangeably in the melody extraction literature, and for the sake of readability we shall do the same here. The final musical term we must define is "polyphonic music." Although musicology draws distinctions between monophonic, homophonic, heterophonic, and polyphonic musical textures, in this article "polyphonic" is simply used to refer to any type of music in which two or more notes can sound simultaneously, be it on different instruments (e.g., voice, guitar, and bass) or a single instrument capable of playing more than one note at a time (e.g., the piano).

With these definitions of melody and polyphony, it becomes easier to define melody extraction as a signal processing challenge: given a recording of polyphonic music, we want to automatically estimate the sequence of f_0 values that corresponds to the pitch of the lead voice or instrument. Furthermore, we must estimate the time intervals when this voice is not present in the mixture (known as the "voicing detection" problem). For a human listener, this task might seem almost trivial—many of us can sing the melodies of our favorite songs even without any musical training. Those with musical training can even transcribe a melody into musical notation. However, when we try to automate this task, it turns out to be highly challenging. The complexity of the task is mainly due to two factors: first, a polyphonic music signal is composed of the superposition of the sound waves produced by all instruments in the recording, and much of the time these instruments play simultaneously. When considering the spectral content of the signal, the frequency components of different sources superimpose making it very hard to attribute specific energy levels in specific frequency bands to the notes of individual instruments. This is further complicated by mixing and mastering techniques which can add reverberation (thus blurring note onsets and offsets and increasing the overlap of sound sources) or apply dynamic range compression (thus reducing the difference between soft and loud sources, increasing interference). Second, even after we obtain a pitch-based representation of the audio signal, we still need to determine which pitch values belong to the predominant melody and which are merely accompaniment. The challenge is illustrated in Figure 2, which displays the spectrograms of three polyphonic excerpts (a)–(c) and the target melody sequence [(d)–(f), in red] together with the estimate (in blue) of a melody extraction algorithm (see the next section).

As we discuss in the section "Software and Applications," melody extraction has many potential applications, including query-by-humming (QBH) (searching for a song by singing or humming part of its melody) and cover song identification (detecting whether two recordings are different renditions of the same musical piece) [8], [9], genre classification (automatically sorting your music collection based on genre) [10], music desoloing for the automatic generation of karaoke accompaniment [11], and singer characterization [12]. It also has a wide range of applications in computational musicology and ethnomusicology, such as music transcription [13], intonation analysis [14], and automatic melodic motif and pattern analysis [15]. Determining the melody of a song could also be used as an intermediate step toward the derivation of other semantic labels from music signals. Finally, melody extraction also has a variety of uses outside

FOR MORE INFORMATION

- Adobe Audition: http://www.adobe.com/products/audition/html
- Melodyne: http://www.celemony.com/cms
- SMSTools: http://mtg.upf/edu/technologies/sms
- Wavesurfer: http://www.speech.kth.se/wavesurfer
- LabROSAmelodyextract2005: http://labrosa.ee.columbia.edu/projects/melody/
- FChT: http://iie.fing.edu/uy/investigacion/grupos/gpa/fcht.html
- separateLeadStereo: http://www.durrieu.ch/research/jstsp2010.html
- IMMF0salience: https://github.com/wslihgt.IMMF0salience
- Vamp audio analysis plug-in system: http://www.vamp-plugins.org
- MELODIA: http://mtg.upf.edu/technologies/melodia
- Melody Extraction for Music Games: http://www.idmt.fraunhofer.de/en/Service_Offerings/products_and_technologies/m_p/melody_extraction.html
- SoundHound: http://www.soundhound.com
- ADC2004 and MIREX05 data sets: http://labrosa.ee.columbia.edu/projects/melody/
- MIR-1K data set: https://sites.google.com/site/unvoicedsoundseparation/mir-1k
- RWC pop data set: http://staff.aist.go.jp/m/goto/RWC-MDB/
- Audio Melody Extraction Annotation Initiative: http://ameannotationinitiative.wikispaces.com

the realm of research, such as electroacoustic composition and music education. Melody extraction technologies are beginning to be incorporated into professional music production tools such as Adobe Audition and Melodyne (see "For More Information").

CASE STUDY

To better understand the challenges of melody extraction and the types of errors afflicting melody extraction algorithms, we start with a closer look at the actual melody extraction results for some musical excerpts. For conciseness, we limit ourselves to one state-of-the-art algorithm [16], but the types of errors we observe (and the challenges they represent) are common to all methods.

Figure 2 shows the output of the algorithm for three excerpts in the genres of vocal jazz [(d)], pop music [(e)], and opera [(f)]. In (a)–(c), we display a log-frequency spectrogram of each excerpt, showing the complex pattern of harmonics associated with these polyphonic musical signals. Plots (d)–(f) display the final melody line estimated by the algorithm (blue) overlaid on top of the ground truth annotation (red).

Before we can interpret different types of errors in the plots, it is useful to know what a correct extraction looks like, provided in Figure 2(d). We see that the blue (estimated) and red (ground truth) melody sequences overlap almost perfectly, and there are practically no frames where only one sequence is present. The perfect overlap means the pitch estimation of the algorithm is correct. The fact that there are no frames where only one sequence is present indicates we have not made any voicing detection mistakes—a red sequence on its own would mean we wrongly estimated the frame as unvoiced when the melody is actually present. A blue sequence on its own would mean a case of voicing false alarm, i.e., a frame where we mistakenly included some other pitched source in the melody when the melody is in fact not present in that frame. In (d), we see that the algorithm correctly estimates the pitch of the lead singer while excluding the notes of the piano chord played between seconds three and four.

In Figure 2(e), we provide an example that contains both pitch errors (seconds four to seven) and voicing errors (seconds seven to nine). The excerpt is taken from a pop song whose arrangement includes a lead singer, guitar accompaniment, and backing vocals. Here the source of both types of errors are the backing vocals, who sing a stable pitch in the same range as the melodic line of the lead singer. As a result, the algorithm mistakenly tracks the backing vocals, resulting in a wrong pitch estimate (up to the seventh second) followed by a voicing false alarm, since the backing vocals continue after the lead singer has paused.

Finally, in Figure 2(f), we provide an example where the algorithm makes octave errors. In this excerpt, taken from an opera aria sung by a male singer, the pitch class of the melody is correctly estimated but in the wrong octave (one octave above the actual pitch of the singer). Here the octave errors most likely stem from the actual singing technique used by the singer. Unlike pop or jazz singers, classical singers are trained to produce a highly resonant sound (allowing them to be heard over the orchestra). In the low frequencies this resonance results in the second harmonic often having a larger amplitude than the fundamental frequency, and in the high frequencies the appearance (especially in male singers) of a clear formant around 3 kHz (the "singer's formant") [17]. Combined, these phenomena can cause the algorithm to give more weight to $2f_0$ than to f_0 (f_0 being the correct fundamental frequency), as seen in the spectrogram in Figure 2(c) between seconds ten and 12. The increased salience at double the true f_0 combined with the relatively low pitch range of the melody (algorithms often bias the tracking against low frequencies) results in the algorithm tracking the melody one octave above the correct pitch, thus producing the observed octave errors.

ALGORITHM OVERVIEW: 2005 TO DATE

Melody extraction is strongly linked to pitch (fundamental frequency) estimation, which has a long research tradition. Early approaches for pitch estimation in music dealt with the estimation of the f_0 of monophonic music recordings and were adopted from the speech processing literature [18]. Since then, various approaches specifically tailored for f_0 estimation in monophonic music signals have been proposed [19]. More recently, algorithms have also been proposed for estimating the f_0 of multiple concurrent instruments in polyphonic recordings (multipitch estimation). For a detailed review, the reader is referred to [20]. As seen in the section "Introduction," melody extraction differs from both monophonic and multipitch estimation in two important ways. Unlike monophonic pitch estimation, here we are dealing with polyphonic material and the challenges it entails. Unlike multipitch estimation, melody extraction requires the identification of the specific voice that carries the melody within the polyphony, but does not involve estimating the pitch values of the remaining sources.

It is instructive to consider melody extraction systems as elaborations of monophonic pitch trackers. Monophonic pitch trackers usually take the audio signal $x(t)$ and calculate a function $S_x(f_\tau, \tau)$ evaluated across a range of candidate pitch frequencies f that indicates the relative score or likelihood of the pitch candidates at each time frame τ. The function can be calculated either in the time domain (e.g., the autocorrelation evaluated over a range of lags) or the frequency domain (e.g., some function of the magnitude spectrum evaluated over a range of frequencies). The local estimates of period are then typically subject to sequential constraints, for instance, via dynamic programming. Thus, the estimated sequence of pitch values $\hat{\mathrm{f}}$, represented as a vector with one value for each time frame, is derived as

$$\hat{\mathrm{f}}_{\mathrm{mon}} = \arg\max_{\mathrm{f}} \sum_{\tau} S_x(f_\tau, \tau) + C(\mathrm{f}), \tag{1}$$

where f_τ is the τth element of f, and $C(\mathrm{f})$ accounts for the temporal constraints. For example, a common choice for $S_x(f, \tau)$ is an autocorrelation function such as

$$S_x(f, \tau) = r_{xx}\left(\frac{1}{f}; \tau\right) = \frac{1}{W}\int_{\tau - W/2}^{\tau + W/2} x(t)x\left(t + \frac{1}{f}\right)dt, \tag{2}$$

where W is the length of the autocorrelation analysis window. In melody extraction, the observed signal $y(t)$ consists of a target monophonic melody signal $x(t)$ with added accompaniment "noise"

$$y(t) = x(t) + n(t). \tag{3}$$

There are two paths to extending monophonic trackers to succeed in such conditions: we could improve the robustness of the underlying pitch candidate scoring function, so it continues to reflect the desired pitch even in the presence of other periodicities; we call this *salience-based* melody extraction

$$\hat{\mathrm{f}}_{\mathrm{sal}} = \arg\max_{\mathrm{f}} \sum_{\tau} S'_y(f_\tau, \tau) + C'(\mathrm{f}), \tag{4}$$

where S'_y is the modified pitch salience function calculated over the mixed signal y. There are many different approaches for calculating the salience function (cf. the section "Salience Function"). For instance, some functions compute the salience of a candidate frequency f as the weighted sum of its harmonics

$$S'_y(f_\tau, \tau) = \sum_{h=1}^{N_h} g(f_\tau, h)\,|\,Y(h \cdot f, \tau)\,|, \tag{5}$$

where N_h is the number of harmonics in the summation, $g(f_\tau, h)$ is a harmonic weighting function [5], and $Y(f, \tau)$ is the short-time Fourier transform (STFT),

$$Y(f, \tau) = \int_{-W/2}^{W/2} w(t)y(\tau + t)e^{-j2\pi ft}dt, \tag{6}$$

where $w(t)$ is a windowing function.

Note that in (4) we now use $C'(\mathrm{f})$ to represent the temporal constraints instead of $C(\mathrm{f})$, since for the polyphonic case this is a far more complex problem: even with a modified salience function there is no guarantee that the frequency of the melody will always be found at the maximum of the function. As shall be seen in the section "Tracking," this is addressed by employing tracking techniques such as Viterbi decoding, tracking agents, clustering, etc.

Alternatively, we could attempt to decompose the mixed signal into separate sources, at least one of which, $\hat{x}(t)$, is dominated by the melody signal to a degree that makes it suitable for a largely unmodified pitch tracker; we call this *source separation* melody extraction

$$\hat{\mathrm{f}}_{\mathrm{sep}} = \arg\max_{\mathrm{f}} \sum_{\tau} S_{\hat{x}}(f_\tau, \tau) + C'(\mathrm{f}), \tag{7}$$

where $\hat{x}(t)$ is estimated using decomposition or matrix factorization techniques (cf. the section "Source Separation-Based Approaches").

THE MIREX MELODY EXTRACTION EVALUATIONS

Since its initiation in 2005, over 50 melody extraction algorithms have been submitted to MIREX [3]. In this annual campaign, different algorithms are evaluated against the same set of music collections to obtain a quantitative comparison between methods and assess the accuracy of the current state of the art in melody extraction. We believe MIREX is a good point of reference for this review, given that the large majority of melody extraction algorithms that have had an impact on the research community have participated in MIREX at some point. Due to space limitations, approaches predating 2005 (e.g., [1]) are not discussed in this article, and we refer the reader to [20] for further information on earlier work.

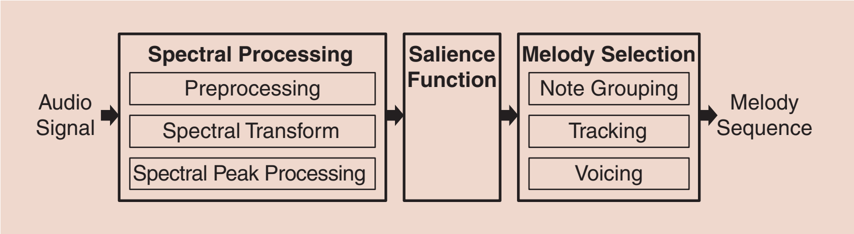
In Table 1, we provide a summary of the characteristics of a selection of 16 representative algorithms out of all the

[TABLE 1] ALGORITHMIC ARCHITECTURE OF 16 MELODY EXTRACTION ALGORITHMS FROM MIREX FROM 2005 TO 2012.

FIRST AUTHOR/ MIREX YEAR	PREPROCESSING	SPECTRAL TRANSFORM AND PROCESSING	MULTIPITCH REP. (SALIENCE FUNCTION)	TRACKING	VOICING	APPROACH TYPE
PAIVA 2005 [25]	—	AUDITORY MODEL + AUTOCORRELATION PEAKS	SUMMARY CORRELOGRAM	MULTIPITCH TRAJECTORIES + NOTE DELETION	SALIENCE VALLEYS	SALIENCE BASED
MAROLT 2005 [26]	—	STFT + SMS HARMONICS PLUS NOISE	EM FIT TO TONE MODELS	FRAGMENTS + FRAGMENT CLUSTERING	LOUDNESS FILTER	SALIENCE BASED
GOTO 2005 [27]	BANDPASS FILTER	MULTIRATE FILTERBANK + IF-BASED PEAK SELECTION	EM FIT TO TONE MODELS	TRACKING AGENTS	—	SALIENCE BASED
CANCELA 2008 [28]	—	CONSTANT-Q + HIGH PASS FILTER + LOG POWER NORM.	HARMONICITY MAP	CONTOUR TRACKING + WEIGHTING + SMOOTHING	ADAPTIVE THRESHOLD	SALIENCE BASED
RYYNÄNEN 2008 [5]	—	STFT + SPECTRAL WHITENING	HARMONIC SUMMATION	NOTE EVENT HMM + GLOBAL HMM	SILENCE MODEL	SALIENCE BASED
DRESSLER 2009 [29]	—	MRFFT + IF PEAK CORRECTION + MAGNITUDE THRESH.	PAIRWISE COMPARISON OF SPECTRAL PEAKS	STREAMING RULES	DYNAMIC THRESHOLD	SALIENCE BASED
RAO 2009 [30]	—	HIGH RESOLUTION FFT + MAIN-LOBE MAG. MATCHING	SMS + TWM	DYNAMIC PROGRAMMING	NHC THRESHOLD	SALIENCE BASED
SALAMON 2011 [16]	EQUAL LOUDNESS FILTER	STFT + IF PEAK CORRECTION	HARMONIC SUMMATION	CONTOUR TRACKING + CONTOUR FILTERING	SALIENCE DISTRIBUTION	SALIENCE BASED
JO 2011 [31]	—	STFT WITH VARYING WINDOW LENGTH	HARMONIC SUMMATION	STABLE CANDIDATES + RULE-BASED SELECTION	IMPLICIT	SALIENCE BASED
ARORA 2012 [32]	—	STFT + LOG SPECTRUM + PEAK SELECTION	IFT OF LOG SPECTRUM	HARMONIC CLUSTER TRACKING + CLUSTER SCORE	HARM. SUM. THRESHOLD	SALIENCE BASED
HSU 2010 [33]	HARM/PERC SOUND SEP.	MRFFT + VOCAL PARTIAL DISCRIMINATION	NORMALIZED SUBHARMONIC SUMMATION	GLOBAL TREND + DYNAMIC PROGRAMMING	CLASSIFICATION	SALIENCE BASED + SOURCE SEP. PREPROCESSING
YEH 2012 [34]	HARM/PERC SOUND SEP.	MRFFT + VOCAL PARTIAL DISCRIMINATION	NORMALIZED SUBHARMONIC SUMMATION	TREND ESTIMATION + HMM	—	SALIENCE BASED + SOURCE SEP. PREPROCESSING
DURRIEU 2009 [22]	SOURCE/FILTER MODEL FOR MELODY SOURCE SEPARATION			VITERBI SMOOTHING	ENERGY THRESHOLD	SOURCE SEPARATION
TACHIBANA 2011 [23]	TWO-STAGE HARMONIC/PERCUSSIVE SOUND SEPARATION			DYNAMIC PROGRAMMING	SIGNAL/NOISE RATIO THRESHOLD	SOURCE SEPARATION
POLINER 2006 [21]	DOWNSAMPLE TO 8 kHz	STFT + LIMIT TO 2 kHz + NORMALIZE MAGNITUDE	N/A	SUPPORT VECTOR MACHINE CLASSIFIER	ENERGY THRESHOLD	DATA DRIVEN
SUTTON 2006 [35]	SEMITONE ATT. + BANDPASS	N/A	N/A	HMM COMBINATION OF MONOPHONIC PITCH TRACKERS	CONFIDENCE HMM	MONOPHONIC

submissions to MIREX since 2005. To do so, we have attempted to break down the extraction process into a series of steps that are common to most algorithms. Since some authors submitted several algorithms over the years, we have opted to include only their most recent (published) contribution, as in most cases it represents the latest version in the evolution of a single algorithm. If a certain step is not included in an algorithm (or otherwise not mentioned by the authors) a "—" is placed in the table. "N/A" means a step is not relevant to the method (e.g., Poliner and Ellis [21] determine the melody directly from the power spectrum and hence a multipitch representation of the audio signal is not relevant for this approach). Finally, we note that some algorithms (those by Durrieu [22] and Tachibana [23]) cannot be broken down into the same steps as the rest of the approaches. This is indicated by fusing the columns of some steps in the table for these algorithms.

The last column of the table, "Approach Type," attempts to classify the algorithms based on their underlying approach, with most falling into the categories of salience based and source separation introduced above. Some approaches, however, do not fit into either category, including the data-driven approach in which the power spectrum is fed directly into a machine-learning algorithm that attempts to classify the melody frequency based on the observed spectrum at each frame.



[FIG3] A block diagram of salience-based melody extraction algorithms.

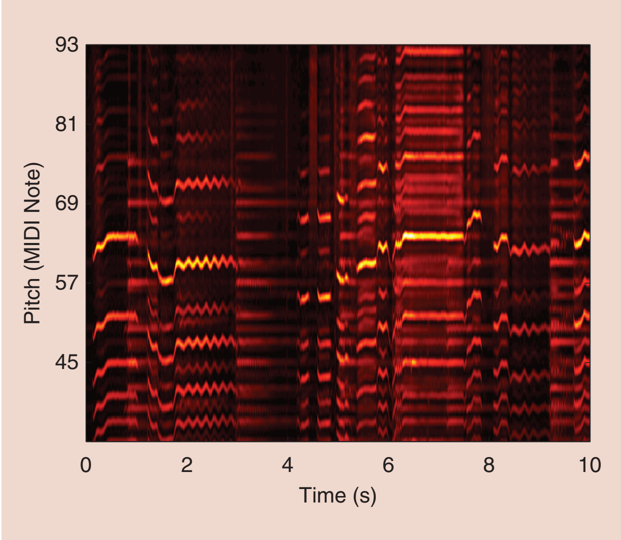
Note that while melody extraction includes detecting both sung melodies and melodies played by lead instruments, many algorithms are developed particularly for singing voice extraction. The reason for this is twofold: first, there is a large body of popular music with sung melodies, which makes vocal melody extraction commercially attractive. Second, the singing voice has unique characteristics that are different from most instruments [24], and algorithms can exploit these unique features to identify the melody more accurately.

SALIENCE-BASED APPROACHES

As evident in Table 1, the largest set of approaches are those based on time-frequency representations of pitch salience (a salience function). The general architecture of these approaches, with possible substeps, is depicted in Figure 3.

PREPROCESSING

As a first step, some approaches apply some type of preprocessing, normally a filter to enhance the frequency content where we expect to find the melody: Goto [27] applies a bandpass filter between 261.6 Hz and approximately 4 kHz, while Salamon and Gómez [16] apply a perceptually motivated equal loudness filter [7]. Some approaches use source separation to enhance the melody signal before it is further processed: Hsu [33] and Yeh [34] use a technique originally designed for harmonic-percussive sound separation (HPSS) adapted to perform melody-accompaniment separation (cf. the section "Source Separation-Based Approaches").



[FIG4] An example of the output of a salience function for an excerpt of vocal jazz [Figure 2(a) and (d)] computed using the algorithm proposed in [16].

SPECTRAL TRANSFORM AND PROCESSING

Next, the signal is chopped into time frames and a transform function is applied to obtain a spectral representation of each frame. The most straightforward approach is to apply the STFT, with a window length typically between 50 and 100 ms [5], [16], [26], [30], [32]. Such a window length usually provides sufficient frequency resolution to distinguish different notes while maintaining adequate time resolution to track pitch changes in the melody over short time periods. Still, some approaches attempt to overcome the time-frequency resolution limitation inherent to the Fourier transform by applying a multiresolution transform such as a multirate filterbank [27], the constant-Q transform [28], or the multiresolution FFT (MRFFT) [33], [34], [36]. In general, these transforms use larger windows at low frequencies (where we require greater frequency resolution to resolve close notes) and small windows at higher frequencies (where we need high-temporal resolution to track rapidly changing harmonics). In [16], a comparison between the STFT and MRFFT showed there was no statistically significant difference between using one transform over another for melody extraction. Nonetheless, since each step in a melody extraction system is highly sensitive to the output of the preceding step, it is possible that some algorithms do benefit from using multiresolution transforms. Finally, we note that some methods use transforms designed to emulate the human auditory system [7] such as the model used by Paiva [25].

After applying the transform, most approaches only use the spectral peaks for further processing. Apart from detecting the peaks themselves, different peak processing techniques may be applied: some methods filter peaks based on magnitude or sinusoidality criteria in an attempt to filter out peaks that do not represent harmonic content or the lead voice [26], [27], [30], [33], [34]. Other approaches apply spectral magnitude normalization in an attempt to reduce the influence of timbre on the analysis—Cancela [28] and Arora [32] take the log spectrum and Ryynänen and Klapuri (who use the whole spectrum, not just the peaks) apply spectral whitening [5]. Finally, Dressler [36] and Salamon and Gómez [16] obtain more accurate frequency and amplitude estimates for each spectral peak by computing its instantaneous frequency from the phase spectrum.

SALIENCE FUNCTION

At the core of salience-based algorithms lies the multipitch representation, i.e., the salience function. This function provides an estimate of the salience of each possible pitch value (within the range where we expect to find the melody) over time. An example of the output of a salience function (used by

Salamon and Gómez [16]) is depicted in Figure 4. 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 [5], [16], [28], [31], [33], [34]. Goto [27] and Marolt [26] use expectation maximisation to fit a set of tone models to the observed spectrum. The estimated maximum a posteriori probability (MAP) of the tone model whose f_0 corresponds to a certain pitch is considered to be the salience of that pitch. Other approaches include two-way mismatch computed by Rao [30], summary autocorrelation used by Paiva [25], and pairwise analysis of spectral peaks as done by Dressler [37].

As evident in Figure 4, the salience function approach has one main undesirable effect—the appearance of the "ghost" pitch values whose f_0 is an exact multiple (or submultiple) of the f_0 of the actual pitched sound. This effect can lead to what is commonly referred to as *octave errors*, in which an algorithm selects a pitch value that is exactly one octave above or below the correct pitch of the melody. [This type of error can be observed in Figure 2(f).] Different algorithms adopt different strategies to reduce the number of octave errors they commit. Some algorithms, such as the ones by Cancela [28] and Dressler [29], attempt to directly reduce the number of ghost pitch values present in the salience function. Dressler does this by examining pairs of spectral peaks that potentially belong to the same harmonic series and attenuating the result of their summation if there are many high amplitude spectral peaks whose frequencies lie between the pair being considered. Cancela attenuates the harmonic summation supporting a certain f_0 if the mean amplitude of spectral components at frequencies $2k \cdot f_0, 3k \cdot f_0/2$ and $3k \cdot f_0$ is above the mean of the components at frequencies $k \cdot f_0$ (this will attenuate ghost pitch values whose f_0 is 1/2, 2/3, or 1/3 of the real f_0). In [20], Klapuri proposes a method for reducing octave errors based on spectral smoothness. The amplitude of each peak in the salience function is recalculated after smoothing the spectral envelope of its corresponding harmonic frequencies. Peaks representing octave errors will have an irregular envelope (compared to a smoother envelope for real notes) and thus will be attenuated by this process. An alternative approach for coping with octave errors is proposed by Paiva [25] and Salamon [16], who first group the peaks of the salience function into pitch contours and then determine which contours are actually ghost contours and remove them. The underlying idea is that once salience peaks are grouped into contours, detecting duplicate contours becomes easier since they have identical trajectories one octave apart. Determining which of the two is the ghost contour is done using criteria based on contour salience and the overall pitch continuity of the melody. Finally, we note that practically all methods reduce octave errors nonexplicitly by penalizing large jumps in pitch during the tracking stage of the algorithm.

TRACKING

Given the peaks of the salience function, the remaining task is to determine which peaks (i.e., pitch values) belong to the melody. This is one of the most crucial stages of each algorithm and, interestingly, it is also perhaps the most varied step where practically every algorithm uses a different approach. Most approaches attempt to directly track the melody from the salience peaks, though some (Paiva, Marolt, Cancela, and Salamon) include a preliminary grouping stage where peaks are grouped into continuous pitch contours (also referred to as *fragments* or *trajectories*) out of which the melody is later selected [16], [25], [26], [28]. This grouping is usually performed by tracking sequential peaks based on time, pitch, and salience continuity constraints. Given the pitch contours (or salience peaks if no grouping is applied), a variety of tracking techniques have been proposed to obtain the final melody sequence: Marolt [26] uses clustering, while Goto [27] and Dressler [29] use heuristic-based tracking agents. Ryynänen [5] and Yeh [34] use HMMs, while Rao [30] and Hsu [33] use dynamic programming. Finally, Paiva [25] and Salamon [16] take a different approach—rather than tracking the melody, they attempt to delete all pitch contours (or notes) that do not belong to the melody.

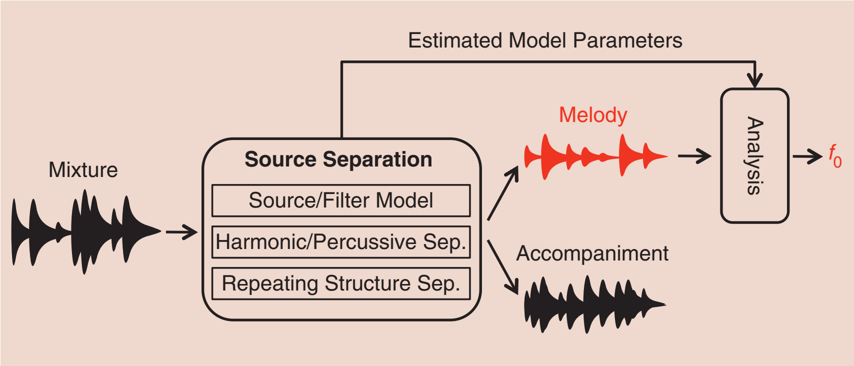
VOICING

An important part of melody extraction that is sometimes overlooked is voicing detection, i.e., determining when the melody is present and when it is not. The voicing detection step of an algorithm is usually applied at the very end, though exceptions do exist (e.g., Salamon uses a threshold based on the salience distribution of pitch contours in the entire piece to remove nonsalient contours before proceeding to filter out other nonmelody contours). A common approach is to use a fixed or dynamic per-frame salience-based threshold, as done by Paiva, Marolt, Cancela, Dressler, Rao, and Arora. Alternative strategies include Ryynänen's algorithm, which incorporates a silence model into the HMM tracking part of the algorithm, and Hsu's algorithm, which uses timbre-based classification to determine the presence (or absence) of human voice.

SOURCE SEPARATION-BASED APPROACHES

An alternative strategy to salience-based melody extraction is to use source separation algorithms to isolate the melody source from the mixture. A block diagram illustrating some of the strategies for melody extraction using source separation is provided in Figure 5. This type of approach is the most recent of the ones mentioned in Table 1 and has gained popularity in recent years following the advances in audio source separation research. While there is a large body of research on melody and lead voice source separation (cf. [22] and [38]–[43] and references therein), such algorithms are usually evaluated using measures based on signal to noise ratios, and only few have been evaluated in terms of estimating the frequency sequence of the melody, as is our goal here.

Two methods in Table 1 are source separation based—those of Durrieu et al. [22] and Tachibana et al. [23]. Durrieu models the power spectrogram of the signal as the instantaneous sum of two



[FIG5] A block diagram of source separation-based melody extraction algorithms.

contributions: the lead voice and the accompaniment. The contribution of the lead voice is represented with a source/filter model, and the contribution of the accompaniment as the sum of an arbitrary number of sources with distinct spectral shapes. Two different representations are proposed for the source/filter model: a smooth instantaneous mixture model (SIMM) and a smooth Gaussian scaled mixture model (SGSMM). The former represents the lead instrument (or voice) as the instantaneous mixture of all possible notes, while the latter is more realistic in that it only allows one source/filter couple to be active at any moment, albeit computationally heavier. In both cases, the model parameters are estimated using an expectation maximization framework. Once the model parameters are estimated, the final melody sequence is obtained using the Viterbi algorithm to find a smooth trajectory through the model parameters (which include the f_0 of the source). Voicing detection is done by first using Wiener filtering to separate the melody signal based on the estimated model parameters, and then computing the energy of this signal at every frame to determine an energy threshold for frames where the melody is present.

The approach proposed by Tachibana et al. is quite distinct. It is based on exploiting the temporal variability of the melody compared to more sustained chord notes. To do so, they make use of the HPSS algorithm [44]. The algorithm was originally designed to separate harmonic from percussive elements in a sound mixture by separating sources that are smooth in time (harmonic content) and sources smooth in frequency (percussive content). By changing the window length used for the analysis, the algorithm can be used to separate "sustained" (i.e., chord) sounds from "temporally variable" (melody plus percussive) sounds. Once the accompaniment is removed, the algorithm is run again, this time in its original form to remove percussive elements. After these two passes, the melody in the resulting signal should be significantly enhanced. The melody frequency sequence is obtained directly from the spectrogram of the enhanced signal using dynamic programming by finding the path which maximizes the MAP of the frequency sequence, where the probability of a frequency given the spectrum is proportional to the weighted sum of the energy at its harmonic multiples, and transition probabilities are a function of the distance between two subsequent frequency values. Voicing detection is done by setting a threshold on the (Mahalanobis) distance between the two signals produced by the second run of the HPSS algorithm (the melody signal and the percussive signal).

Finally, in Table 1 we see that some authors attempt to combine salience-based and source separation approaches. Here, source separation is used as a preprocessing step to attenuate the accompaniment signal, and then a salience function is computed from the processed signal. Both Hsu [33] and Yeh [34] use the HPSS method proposed by Tachibana, but rather than attempt to estimate the melody directly from the spectrum of the resulting signal, they continue to compute a salience function and further steps similar to other salience-based approaches.

For completeness, we briefly describe some singing voice source separation algorithms here. As mentioned earlier, while these methods have not been evaluated in terms of melody extraction, they could be used to build melody extraction systems by combining them with a monophonic pitch tracking algorithm that estimates the melody f_0 sequence from the separated voice signal, or by using them as a preprocessing step similar to the aforementioned approaches by Hsu and Yeh. We have already seen the source/filter model proposed by Durrieu et al. [22] and the HPSS method employed by Tachibana et al. [23]. A different strategy for separating the lead voice is to exploit the fact that the music accompaniment often has a repetitive structure, while the voice contains more variation. Huang et al. [41] exploit this by assuming that the spectrogram of the accompaniment can be modeled by a low-rank matrix, and the spectrogram of the voice by a sparse matrix. They use robust principal component analysis (RPCA) to factorize the spectrogram of the signal into the desired voice and accompaniment matrices. A different way of exploiting repetition is proposed by Rafii and Pardo [42]—they first compute the repetition period of the accompaniment using autocorrelation applied to the spectrogram of the mixture. By computing the median of the spectrograms of consecutive repetitions, they obtain a spectrogram that contains only the repeating signal (the accompaniment). This spectrogram is used to derive a time-frequency mask used to separate the voice from the accompaniment. This approach was extended by Liutkus et al. [43] to work on full songs (where the repetition period can change between verse and chorus) by searching for local periodicities in a song, and again by Rafii and Pardo by applying the algorithm to local windows of the signal and by computing a self-similarity matrix to better identify repeating segments in a song. In [42], the authors also present some experiments on combining their approach with existing pitch trackers to perform melody extraction, and we expect to see an increase in the number of source separation-based melody extraction algorithms participating in MIREX in the future.

ALTERNATIVE APPROACHES

While most melody extraction approaches are either salience or source separation based, some very different strategies have been proposed as well. The first to appear in Table 1 is the data-driven approach by Poliner and Ellis [21]. Rather than handcraft knowledge about musical acoustics into the system (e.g., in the form of a salience function based on harmonic summation), they propose to use machine learning to train a classifier to estimate the melody note directly from the power spectrum. As a preprocessing step they downsample the audio to 8 kHz, and use the STFT to obtain a spectral representation. Bins corresponding to frequencies above 2 kHz are discarded and the magnitude of the remaining bins is normalized over a short time period to reduce the influence of different instrument timbres. The resulting 256 feature vector is used to train a support vector machine classifier using training data where each frame is labeled with one of 60 possible output classes corresponding to 60 MIDI notes spanning five octaves. Voicing detection is done by means of a global threshold based on the magnitude squared energy found between 200 and 1,800 Hz.

> MELODY EXTRACTION ALGORITHMS ARE EXPECTED TO ACCOMPLISH TWO GOALS: ESTIMATE THE CORRECT PITCH OF THE MELODY (PITCH ESTIMATION), AND ESTIMATE WHEN THE MELODY IS PRESENT AND WHEN IT IS NOT (VOICING DETECTION).

Another completely different strategy is the one proposed by Sutton et al. [35]. Rather than design an algorithm to handle polyphonic audio signals, they compute the pitch sequences returned by two different monophonic pitch estimators and then combine them using an HMM. The underlying assumption is that while monophonic pitch estimators are not designed to handle audio where there is more than one pitch present at a time (normally leading to a large degree of estimation errors), by combining the output of different estimators a more reliable result could be obtained.

EVALUATION: MEASURES AND MUSIC COLLECTIONS

As explained earlier, melody extraction algorithms are expected to accomplish two goals: estimate the correct pitch of the melody (pitch estimation), and estimate when the melody is present and when it is not (voicing detection). The output of a melody extraction algorithm typically includes two columns, the first with timestamps at a fixed interval (e.g., for MIREX a 10-ms interval is used), and the second with f_0 values representing the algorithm's pitch estimate for the melody at each timestamp (i.e., at each analysis frame). Algorithms can report a pitch even for frames where they estimate the melody to be absent (nonmelody frames), in this way allowing us to evaluate pitch estimation and voicing detection independently.

To evaluate the performance of an algorithm for a given audio excerpt, we compare the algorithm's output with the excerpt's ground truth. The ground truth file has the same format as the output file, and contains the correct series of f_0 values representing the melody of the excerpt. The ground truth is produced by running a monophonic pitch tracker on the solo melody track of the excerpt (meaning we require access to the multitrack recording of every song we use for evaluation). Using a graphical user interface such as SMSTools or WaveSurfer (see "For More Information"), the output of the monophonic pitch tracker is manually inspected and corrected if necessary. Given the ground truth file, an algorithm is evaluated by comparing its output on a per-frame basis to the ground truth. For nonmelody frames in the ground truth, the algorithm is expected to indicate that it has detected the absence of melody. For melody frames, the algorithm is expected to return a frequency value matching the one in the ground truth. An algorithm's frequency estimate is considered correct if it is within 50 cents (i.e., half a semitone) of the ground truth.

MEASURES

Based on this per-frame comparison, we compute five global measures that assess different aspects of the algorithm's performance for the audio excerpt in question. These measures were first used in MIREX 2005 [2], and have since become the de facto set of measures for evaluating melody extraction algorithms. If the system's estimated melody pitch frequency vector is $\mathbf{f}$ and the true sequence is $\mathbf{f}^*$, let us also define a voicing indicator vector $\mathbf{v}$, whose τth element $v_\tau = 1$ when a melody pitch is detected, with corresponding ground truth $\mathbf{v}^*$. We also define an "unvoicing" indicator $\bar{v}_\tau = 1 - v_\tau$. Recall that an algorithm may report an estimated melody pitch $(f_\tau > 0)$ even for times where it reports no voicing $(v_\tau = 0)$. Then the measures are as follows:

- **Voicing recall rate:** The proportion of frames labeled as melody frames in the ground truth that are estimated as melody frames by the algorithm

$$\mathrm{Rec}_{\mathrm{vx}} = \frac{\sum_\tau v_\tau v_\tau^*}{\sum_\tau v_\tau^*}. \tag{8}$$

- **Voicing false alarm rate:** The proportion of frames labeled as nonmelody in the ground truth that are mistakenly estimated as melody frames by the algorithm

$$\mathrm{FA}_{\mathrm{vx}} = \frac{\sum_\tau v_\tau \bar{v}_\tau^*}{\sum_\tau \bar{v}_\tau^*}. \tag{9}$$

- **Raw pitch accuracy:** The proportion of melody frames in the ground truth for which f_τ is considered correct (i.e., within half a semitone of the ground truth f_τ^*)

$$\mathrm{Acc}_{\mathrm{pitch}} = \frac{\sum_\tau v_\tau^* \mathcal{T}[\mathcal{M}(f_\tau) - \mathcal{M}(f_\tau^*)]}{\sum_\tau v_\tau^*}, \tag{10}$$

where $\mathcal{T}$ is a threshold function defined by

$$\mathcal{T}[a] = \begin{cases} 1 & \text{if } |a| < 0.5 \\ 0 & \text{if } |a| \geq 0.5 \end{cases} \tag{11}$$

and $\mathcal{M}$ maps a frequency in Hertz to a melodic axis as a real-valued number of semitones above an arbitrary reference frequency f_{ref} (55 Hz, or note pitch A1, in this work):

[TABLE 2] TEST COLLECTIONS FOR MELODY EXTRACTION EVALUATION IN MIREX.

COLLECTION	DESCRIPTION
ADC2004	20 EXCERPTS OF ROUGHLY 20 s IN THE GENRES OF POP, JAZZ, AND OPERA. INCLUDES REAL RECORDINGS, SYNTHESIZED SINGING, AND AUDIO GENERATED FROM MIDI FILES. TOTAL PLAY TIME: 369 s.
MIREX05	25 EXCERPTS OF 10–40 s DURATION IN THE GENRES OF ROCK, R&B, POP, JAZZ, AND SOLO CLASSICAL PIANO. INCLUDES REAL RECORDINGS AND AUDIO GENERATED FROM MIDI FILES. TOTAL PLAY TIME: 686 s.
INDIAN08	FOUR 1-MIN-LONG EXCERPTS FROM NORTH INDIAN CLASSICAL VOCAL PERFORMANCES. THERE ARE TWO MIXES PER EXCERPT WITH DIFFERENT AMOUNTS OF ACCOMPANIMENT, RESULTING IN A TOTAL OF EIGHT AUDIO CLIPS. TOTAL PLAY TIME: 501 s.
MIREX09 (0 dB)	374 KARAOKE RECORDINGS OF CHINESE SONGS (i.e., RECORDED SINGING WITH KARAOKE ACCOMPANIMENT). THE MELODY AND ACCOMPANIMENT ARE MIXED AT A 0-dB SIGNAL-TO-ACCOMPANIMENT RATIO. TOTAL PLAY TIME: 10,020 s.
MIREX09 (–5 dB)	SAME 374 EXCERPTS AS MIREX09 (0 dB), BUT HERE THE MELODY AND ACCOMPANIMENT ARE MIXED AT A –5-dB SIGNAL-TO-ACCOMPANIMENT RATIO. TOTAL PLAY TIME: 10,020 s.
MIREX09 (+5 dB)	SAME 374 EXCERPTS AS MIREX09 (0 dB), BUT HERE THE MELODY AND ACCOMPANIMENT ARE MIXED AT A +5-dB SIGNAL-TO-ACCOMPANIMENT RATIO. TOTAL PLAY TIME: 10,020 s.

$$\mathcal{M}(f) = 12\log_2\left(\frac{f}{f_{\text{ref}}}\right). \tag{12}$$

- **Raw chroma accuracy:** As raw pitch accuracy, except that both the estimated and ground truth f_0 sequences are mapped onto a single octave. This gives a measure of pitch accuracy that ignores octave errors, a common error made by melody extraction systems

$$\text{Acc}_{\text{chroma}} = \frac{\sum_\tau v_\tau^* \mathcal{T}[\langle \mathcal{M}(f_\tau) - \mathcal{M}(f_\tau^*)\rangle_{12}]}{\sum_\tau v_\tau^*}. \tag{13}$$

Octave equivalence is achieved by taking the difference between the semitone-scale pitch values modulo 12 (one octave), where

$$\langle a\rangle_{12} = a - 12\left\lfloor \frac{a}{12} + 0.5\right\rfloor. \tag{14}$$

- **Overall accuracy:** This measure combines the performance of the pitch estimation and voicing detection tasks to give an overall performance score for the system. It is defined as the proportion of all frames correctly estimated by the algorithm, where for nonmelody frames this means the algorithm labeled them as nonmelody, and for melody frames the algorithm both labeled them as melody frames and provided a correct f_0 estimate for the melody (i.e., within half a semitone of the ground truth)

$$\text{Acc}_{\text{ov}} = \frac{1}{L}\sum_\tau v_\tau^* \mathcal{T}[\mathcal{M}(f_\tau) - \mathcal{M}(f_\tau^*)] + \bar{v}_\tau^* \bar{v}_\tau, \tag{15}$$

where L is the total number of frames.

The performance of an algorithm on an entire music collection for a given measure is obtained by averaging the per-excerpt scores for that measure over all excerpts in the collection.

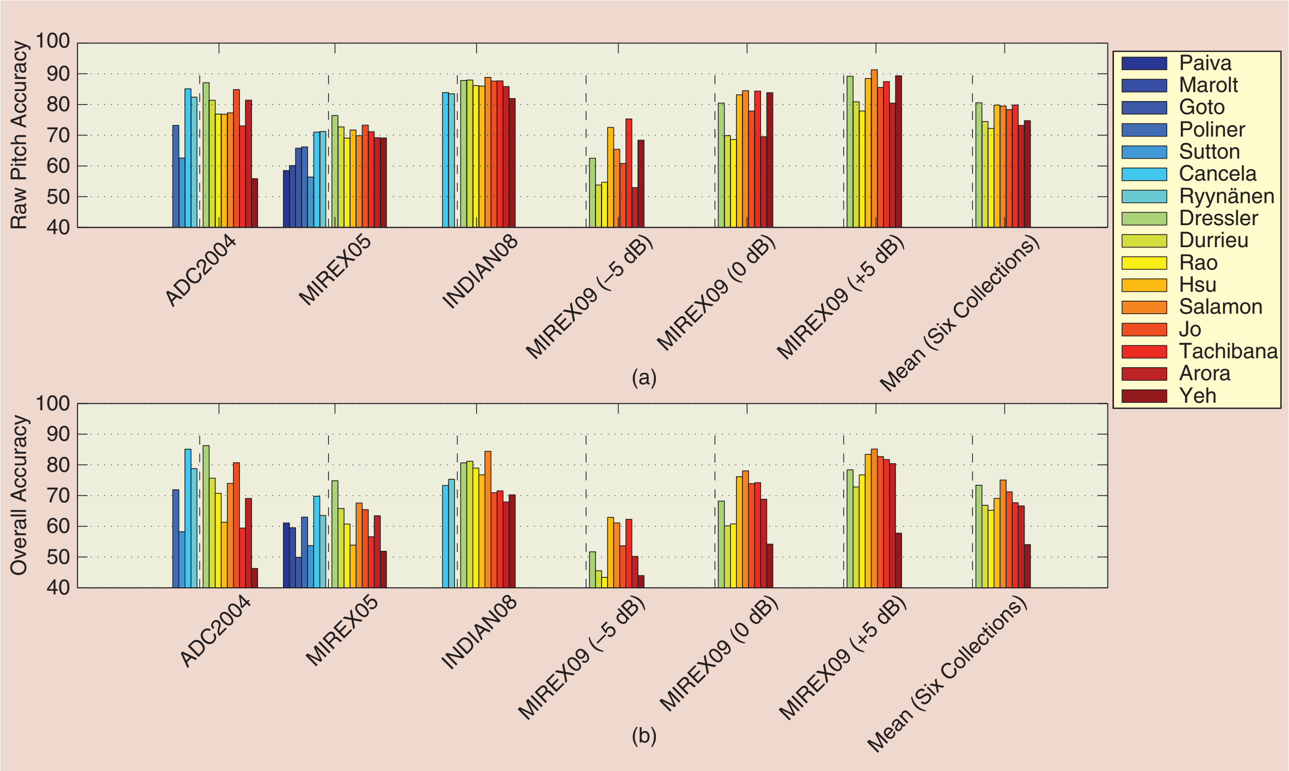
MUSIC COLLECTIONS

Over the years, different research groups have contributed annotated music collections for evaluating melody extraction in MIREX. The limited amount of multitrack recordings freely available, and the time-consuming annotation process, mean most of these collections are relatively small compared to those used in other MIR tasks. The collections currently used for evaluation in MIREX, which have remained fixed since 2009, are described in Table 2.

PERFORMANCE: 2005 TO DATE

EXTRACTION ACCURACY

In Figure 6, we present the results obtained by the 16 algorithms in Table 1 for the MIREX evaluation collections. Note that some algorithms only participated in MIREX before all the collections were added, meaning we only have partial results for these algorithms. This is indicated in the graph with vertical dashed lines that separate the algorithms that were only evaluated on some of the collections (to the left of the line) from those evaluated on all collections (to the right of the line). We only compute the mean for algorithms evaluated on all six collections. To get a general idea of the performance of the algorithms, it is sufficient to focus on two evaluation measures—the raw pitch accuracy [Figure 6(a)] and the overall accuracy [Figure 6(b)]. The former tells us how well the algorithm tracks the pitch of the melody, and the latter combines this measure with the efficiency of the algorithm's voicing detection, meaning the voicing-related measures are (to an extent) also reflected in this measure. Starting with the raw pitch, the first thing we note is that the accuracy of all algorithms varies depending on the collection being analyzed. While some collections are generally harder for all approaches (e.g., MIREX09 (– 5 dB) where the accompaniment is louder and masks the melody), in general the variability in performance is not homogeneous. This highlights the advantages and disadvantages of different approaches with respect to the music material being analyzed. For instance, we see that Dressler's method outperforms all others for the ADC2004 and MIREX05 collections, which contain a mixture of vocal and instrumental pieces, but does not for the other collections where the melody is always vocal. On the one hand this means that her approach is generalizable to a wider range of musical material, but on the other hand we see that approaches that take advantage of specific features of the human voice (e.g., Tachibana or Salamon) can do better on vocal melodies. We also see that the HPSS melody enhancement applied by Hsu, Tachibana, and Yeh is particularly advantageous when the melody source is relatively weak compared to the accompaniment [MIREX09 (– 6 dB)]. Finally, examining the raw pitch accuracy results for the MIREX05 collection, we see that results have improved gradually from 2005 to 2009, after which raw



[FIG6] (a) Raw pitch accuracy and (b) overall accuracy obtained in MIREX by the 16 melody extraction algorithms in Table 1. The vertical dashed line separates the algorithms that were only evaluated on some of the collections (left of the line) from those evaluated on all six collections (right of the line).

pitch accuracies have remained relatively unchanged (more on the evolution of performance in the section "Are We Improving?"). Overall, we see that the average pitch accuracy over all collections lies between 70 and 80%.

Turning over to the overall accuracy, we see that performance goes down compared to the raw pitch accuracy for all algorithms, since voicing detection is now factored into the results. Note that the results for Goto and Yeh are artificially low since these methods do not include a voicing detection step. The importance of this step depends on the intended use of the algorithm. For example, if we intend to use it as a first step in a transcription system, it is very important that we do not include notes that do not belong to the melody in our output. On the other hand, similarity-based applications which rely on matching algorithms that can handle gaps in the alignment of melodic sequences may be less sensitive to voicing mistakes. If we look at the average results over all six collections, we see that the algorithms obtaining the best overall accuracy are those that obtain good raw pitch accuracy combined with an effective voicing detection method. Generally, we see that overall accuracy results lie between 65 and 75% for the best performing algorithms. While this clearly indicates that there are still many challenges remaining (see the section "Challenges"), this degree of accuracy is in fact good enough for new applications to be built on top of melody extraction algorithms (cf. the section "Software and Applications").

Finally, we note that one important aspect of performance that is not reflected in Figure 6 is the computational cost of each approach. Depending on the intended application, we may have limited resources (e.g., time, computing power) and this can influence our decision when choosing which algorithm to use. While deriving O-notation complexity estimates is too complicated for some of the algorithms, generally we observe that algorithms involving source separation techniques (which are often implemented as iterative matrix operations) tend to be significantly more computationally complex than salience-based approaches. In this respect Dressler's algorithm is of particular interest, obtaining both the lowest runtime and the highest mean overall accuracy among the algorithms participating in 2009 (only Salamon and Gómez obtain a higher mean accuracy, but there is no runtime information for 2011).

ARE WE IMPROVING?

In the previous section we noted that, for some collections, performance has not improved much over the last three to four years. In Figure 7, we present the evolution of the overall accuracy obtained for the six MIREX collections over the years. For each collection, we plot the best overall accuracy result obtained up to a given year (e.g., for 2008 we plot the best result obtained up to 2008, for 2009 the best result obtained up to 2009, etc.). Indeed, our previous observation seems to be confirmed—for the two earliest collections (ADC2004 and MIREX05), we observe a steady improvement in results from 2005 to 2009, after which performance does not improve. For the more recent collections (INDIAN08 and the three MIREX09 collections), we see a gradual improvement up to 2011; in 2012 no algorithm outperformed its predecessors for any of the collections. This highlights an important limitation of the MIREX evaluation campaign—since the collections are kept secret, it is very hard for researchers to learn from the results to improve their algorithms. This limitation is discussed further in the section "Challenges."

SOFTWARE AND APPLICATIONS

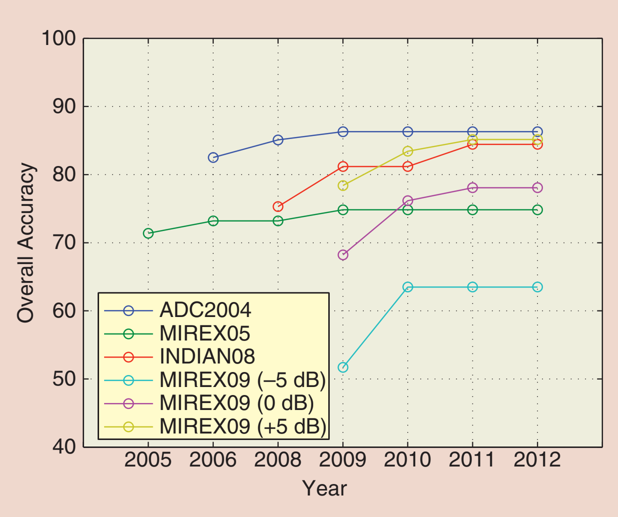
SOFTWARE

While various melody extraction algorithms have been proposed, relatively few implementations are freely available for people to download and use. Such tools are important for facilitating comparative evaluations, increasing the reproducibility of research and facilitating the development of new applications that make use of melody extraction technology (cf. the section "Applications Based on Melody Extraction"). Below we provide a list of known melody extraction related tools that are freely available (for links to all tools mentioned below see "For More Information")

- LabROSAmelodyextract2005 includes the code for the melody extraction system submitted by Poliner and Ellis to MIREX 2005 [21]. Runs on Linux and OSX systems and requires both MATLAB and Java.
- FChT is an open source MATLAB/C++ implementation of the Fan Chirp Transform (FChT) and f_0 gram (salience function) proposed by Cancela et al. in [45].
- separateLeadStereo is an open-source python implementation of the algorithm by Durrieu et al. reported in [40]. The code includes functionality for melody extraction, as well as lead instrument/accompaniment source separation.
- IMMF0salience is an open-source vamp plug-in for visualizing a salience function derived from the intermediate steps of the algorithm by Durrieu et al. [22], [40].
- MELODIA is a vamp plug-in available as a compiled library for Windows, OSX, and Linux. The plug-in implements the melody extraction algorithm by Salamon and Gómez [16], and in addition to its final output (i.e., the f_0 sequence of the melody) it provides visualizations of intermediate steps of the algorithm such as the salience function and pitch contours computed before selecting the final melody.

For completeness, we also briefly mention some commercially available software: Dressler's algorithm is incorporated in Fraunhofer's "Melody Extraction for Music Games" library, and certain melody extraction functionality is also incorporated in Adobe Audition and Melodyne, though the details of the algorithms used in these products are not published.

THE ADVANCES IN ALGORITHMIC PERFORMANCE OF MELODY EXTRACTION ALGORITHMS OVER THE PAST DECADE MEAN THEY NOW PROVIDE SUFFICIENTLY GOOD RESULTS FOR MORE COMPLEX APPLICATIONS TO BE BUILT ON TOP OF THEM.



[FIG7] The evolution of the best overall accuracy result over the years for the six MIREX collections.

APPLICATIONS BASED ON MELODY EXTRACTION

The advances in algorithmic performance of melody extraction algorithms over the past decade mean they now provide sufficiently good results for more complex applications to be built on top of them. Below we provide a summary of some of these applications, whose wide range evidences the importance of melody extraction algorithms for MIR and computational music analysis.

RETRIEVAL

One of the most commercially attractive applications for melody extraction is music retrieval. That is, helping users find the music they are interested in or discover new music by means of automatically analyzing and comparing songs. Within this large application area we highlight two different yet related retrieval applications: version identification (version ID) and QBH. Version ID (also known as cover song ID) is the task of automatically retrieving different versions of a musical recording provided to the system by the user. Use cases range from the detection of copyright violations on Web sites such as YouTube, to automating the analysis of how musicians influence each other's compositions. Since the melody is often one of the few musical facets that remain unchanged across different renditions, various studies have explored the use of melody extraction for version ID, either by attempting to fully transcribe it [46], by using it as a midlevel representation for computing similarity [47], or by combining it with other tonal features (e.g., harmony, bass line, or the accompaniment as a whole) [8], [9].

The second retrieval task, QBH, is designed to help in the scenario where the user remembers the melody of a song but does not have any of its editorial information (e.g., title, album, or artist). QBH systems help the user retrieve this information by allowing them to sing or hum the melody as a search query. One important problem in the creation of QBH systems is the generation of a melody database (song index) against which the sung queries are to be compared. While it is possible to find MIDI versions of many songs on the Internet, such an approach will always be limited since it is not feasible to generate (i.e., transcribe) MIDI files manually for the very large music collections in existence today. Another solution is to match queries against other queries (i.e., user-recorded melodies), as performed by SoundHound (see "For More Information"). While this avoids the need for manual transcription, the approach still suffers from the same "cold start" problem—a song "does not exist" until a user records it. This problem can be alleviated by using melody extraction to automatically create a melody index for QBH systems. While no commercial QBH system based on melody extraction has been launched yet, research prototypes have shown promising results [9], [48], [49].

CLASSIFICATION

Automatic music classification attempts to help individual users as well as managers of large music corpora to organize their collections by automatically assigning descriptive labels to the songs in these collections. One of the most commonly used labels for organizing music is musical genre. The characteristics of the melody are often related to the musical genre (e.g., use of vibrato, pitch range), and could help in its identification. In [10], the authors present a genre classification system based on melody-related features obtained using melody extraction and demonstrate how combining these features with more commonly used timbre-related features such as Mel-frequency cepstral coefficients (MFCCs) can help to improve classification accuracy.

DE-SOLOING

Music de-soloing involves "removing" the lead instrument from a polyphonic music mixture. Doing this automatically is a highly attractive application for karaoke bars and fans—any song could automatically be converted into a karaoke accompaniment. Melody extraction can be used as a first step for de-soloing by providing a "score" of the melody that can be used to guide source separation algorithms in eliminating the melody from the audio mix [11].

TRANSCRIPTION

As we have already shown, a midlevel frequency-based representation of the melody is already very useful for various applications. However, sometimes transcribing all the way to symbolic notation (e.g., Western score notation) is desirable. For starters, music transcription is an attractive end goal in its own right, helping users learn music from automatically generated scores [5]. Automatic transcription can also help formalize the symbolic representation of orally transmitted music traditions, such as Flamenco [13]. Finally, by obtaining a symbolic representation of the melody we can apply the wide range of techniques that have been developed for symbolic melodic similarity and retrieval [4]. In all cases, the first step for obtaining a symbolic transcription of the melody from a polyphonic recording is by applying a melody extraction algorithm, whose output is then quantized in time and pitch to produce musical notes.

COMPUTATIONAL MUSIC ANALYSIS

As a final application, we discuss a couple of examples where melody extraction is useful for computational music analysis. Unlike the previous applications, whose goal was to enhance the way we find, represent, and interact with music, here our goal is to learn about the musical content itself by means of automated analysis. In [15], the authors combine melody extraction with a pattern recognition algorithm to detect the presence (or absence) of musical patterns that were predefined by musicologists. This type of analysis allows musicologists to study important aspects of the given musical style, e.g., to confirm existing musical hypotheses.

In [14], melody extraction is used for a different type of analysis. Here, melodies are extracted from excerpts of Indian classical music and summarized as pitch histograms with a high-frequency resolution. The resulting histograms are used for intonation analysis—an important aspect in Carnatic music (a type of Indian classical music). The intonation of a singer can be used to identify the raga of the piece, as well as characterize the musical expression of the performer.

CHALLENGES

While melody extraction algorithms have improved considerably since 2005, many challenges still remain. In the following sections we discuss some of the important issues, in terms of both algorithmic design and evaluation, that future research on melody extraction will have to address.

INSTRUMENTAL MUSIC AND HIGH DEGREES OF POLYPHONY

Earlier in our review, we mentioned that while most approaches can process instrumental music, many of them are particularly tailored for vocal music. We noted that this stems both from the popularity of vocal music, and from the uniqueness of the human voice which can be exploited by algorithms. However, if we wish to develop algorithms which generalize to a broader range of music material, melody extraction for instrumental music must be properly addressed. This presents two challenges compared with vocal melody extraction: first, instrumental music is not as constrained as vocal music. Instruments have a wider pitch range, can produce rapidly changing pitch sequences and include large jumps in pitch. Second, an instrument playing the melody may be closer, both in timbre and in the pitch contour of individual notes, to other accompanying instruments, which makes the task of distinguishing the melody from the accompaniment more complicated.

Regardless of the instrument playing the melody, the task becomes harder as we increase the number of instruments in the mixture. This causes greater overlap of spectral content, making it harder to determine individual pitched sources correctly. Even when we manage to correctly distinguish the pitch values of different notes, determining which of these belong to the melody is now harder. Currently, algorithms are designed to handle material that is primarily homophonic, i.e., a single dominant lead instrument (or voice) with some harmonic accompaniment (strictly speaking, homophonic implies that the accompaniment shares the same rhythm as the melody, here we use the term more generally to refer to all music which has a lead melody with some form of harmonic accompaniment). Accurately extracting a specific melody from (for example) a fugue with a high degree of polyphony and several competing melodic lines is something current melody extraction algorithms can not do yet. Even in the simpler homophonic case we can think of challenging examples for melody extraction, for instance, songs that have backing vocals or even just a second voice. A second voice will usually move very similarly to the melody, reside in the same pitch range, and often be equally loud. This makes the task of determining which of the two voices is the actual melody highly challenging.

MELODY IS WITHOUT DOUBT A VERY IMPORTANT AND DISTINCT ASPECT OF MUSIC INFORMATION, AND SYSTEMS FOR AUTOMATICALLY EXTRACTING IT FROM MUSIC AUDIO ARE SURE TO BE CENTRAL TO FUTURE MUSIC INFORMATION TECHNOLOGIES.

VOICING DETECTION

When considering algorithmic performance, we saw that the key to obtaining high overall accuracy is the combination of high raw pitch accuracy with a good voicing detection method. To date, most approaches focus primarily on the former aspect of melody extraction, and less so on the latter (in Table 1 we see that some algorithms do not even include a voicing detection step). Often, voicing detection is only considered at the very end of the processing chain by applying a simple global energy threshold. Currently, even the algorithms with the most effective voicing detection methods obtain an average voicing false alarm rate (i.e., detecting melody where there isn't any) of more than 20%. In [16], the authors note that the most significant potential improvement in the performance of their algorithm would come from reducing the voicing false alarm rate, even though it is already one of the lowest in MIREX.

DEVELOPMENT CYCLE AND EVALUATION

In the section "Are We Improving?" we saw that for some MIREX collections performance has not improved significantly in recent years, and it was noted that this highlights a problem in the research and development cycle of melody extraction algorithms. Since the MIREX collections (with the exception of ADC2004) are kept secret for use in future evaluations, researchers have no way of analyzing the data to understand *why* their algorithms fail. Without listening to the audio content and examining the output of intermediate steps of the algorithm, the final results obtained, even if broken into several metrics, only tell you where and how you fail, but not why.

For algorithmic research and development, researchers use open data sets that are freely available. Since preparing a data set usually requires access to multitrack recordings and a considerable amount of manual annotation, there are very few such collections: the ADC2004 data set, the MIREX05 train data set, the MIR-1K data set, and the RWC pop data set (see "For More Information"). But the problem does not end here—the former two collections, while varied in terms of music material, are very small in size (20 and 13 excerpts, respectively), and the latter two, which are larger, are limited to a single musical genre (Chinese and Japanese pop, respectively). This means the collections are either too small to give statistically stable results, or too homogeneous to represent the universe of musical styles on which we would like our algorithms to work.

The current challenges in melody extraction evaluation are studied in detail in [50]. The authors focus on three aspects of evaluation in the MIREX campaign: ground truth generation, the duration of the excerpts used in test collections, and the size and content of the collections themselves. They first show how the lack of a common protocol for generating ground truth annotations could potentially lead to systematic errors in evaluation. By comparing algorithms' performance on excerpts with their performance on shorter subclips taken from the same excerpts, they also show that often short excerpts are not representative of the full song, implying that test collections should use complete songs rather than excerpts. Finally, they discuss the stability and representativeness of the results based on the size of the data sets, as we have already commented above. As the authors note, these findings do not invalidate the MIREX results, but rather emphasize the fact that we can not generalize them with confidence to significantly larger data sets of full songs. In an attempt to answer these problems, the Audio Melody Extraction Annotation Initiative (AMEAI) was launched in late 2012 (see "For More Information"). The goal of the initiative is to establish a common annotation protocol and compile a new, open data set for evaluation. The data set is planned to comprise full songs, large enough to provide statistically stable results and varied enough to represent a larger set of musical genres than those currently represented by existing evaluation collections.

SUMMARY AND CONCLUSIONS

In this article, we provided a review of melody extraction algorithms, considering not only aspects of algorithmic design and performance, but also the very definition of the task, its potential applications, and the challenges that still need to be solved. We started by considering the definition of melody and noted that to develop and evaluate melody extraction algorithms, we require a simplified and pragmatic definition. This was achieved by limiting the task to "single source predominant fundamental frequency estimation from musical content with a lead voice or instrument." We described the challenges melody extraction entails from a signal processing point of view, and noted the differences between melody extraction, monophonic pitch estimation and multipitch estimation. By means of a case study, we highlighted some of the most common errors made by melody extraction algorithms and identified their possible causes. Next, we provided a comprehensive review of algorithmic design by considering 16 of the most relevant algorithms submitted to the MIREX evaluation campaign since 2005. We noted the great diversity of approaches and signal processing techniques applied, and identified two main algorithmic categories: salience-based methods and source separation-based methods. The evaluation measures most commonly used to assess melody extraction algorithms were described, and algorithmic performance was considered in terms of these measures. We saw that the best performing algorithms obtain a raw pitch accuracy between 70 and 80% and an overall accuracy of between 65 and 75%. We also saw that while performance has not improved much for some of the earlier collections, overall performance has improved gradually over the years.

Next, we provided a list of freely available melody extraction software, and considered some of the applications that have already been built on top of melody extraction algorithms, including: retrieval (version ID and QBH), genre classification, automatic de-soloing, music transcription, and computational music analysis. Finally, we considered some of the challenges that still need to be addressed by the research community. We noted that current algorithms are primarily designed to handle homophonic vocal music, and that in the future they will have to be extended to handle instrumental and highly polyphonic material. We highlighted the importance of voicing detection and noted the problem in the development cycle caused by the lack of open evaluation collections. We finally considered the evaluation process itself and noted that to be able to generalize the results obtained by melody extraction algorithms to larger music collections, we require new, larger and more heterogeneous test collections.

After nearly a decade of formal evaluations and many dozens of complete systems, it is fair to ask what we have learned about the best approaches to this problem. In our distinction between salience-based and source separation approaches, we find representatives of both among the best-performing systems according to the evaluations. One might argue that further progress in source separation (and full polyphonic transcription) will ultimately subsume this problem, but even despite the issue of greater computational expense, it remains an open question how best to model the perception and cognitive processing of the full music signal that goes on in the heads of listeners, who are not, we assume, performing a full analysis of the sound into individual sources when they listen to music. Notwithstanding the difficulties in obtaining a precise definition, melody is without doubt a very important and distinct aspect of music information, and systems for automatically extracting it from music audio are sure to be central to future music information technologies.

AUTHORS

Justin Salamon (justin.salamon@upf.edu) obtained a B.A. degree (Hons.) in computer science from the University of Cambridge, United Kingdom, in 2007. In 2008 he obtained his M.S. degree in cognitive systems and interactive media from Universitat Pompeu Fabra (UPF), Barcelona, Spain. Currently he is a researcher and Ph.D. student at the Music Technology Group (MTG), UPF. As part of his doctoral studies, he was a visiting researcher at the Sound Analysis-Synthesis research team of the Institut de Recherche et Coordination Acoustique/Musique (IRCAM), Paris, France. His main field of interest is MIR, with a focus on content-based MIR and audio and music processing, including musical stream estimation, melody and bass line extraction and characterization, QBH/example, classification, music, and melodic similarity and indexing. He is a Student Member of the IEEE.

Emilia Gómez (emilia.gomez@upf.edu) is a postdoctoral researcher and assistant professor (professor lector) at the MTG, Department of Information and Communication Technologies, UPF. She graduated as a telecommunication engineer specialized in signal processing at the Universidad de Sevilla. She received a DEA degree in acoustics, signal processing, and computer science applied to music at IRCAM, Paris. In 2006, she completed her Ph.D. degree in computer science and digital communication at UPF on the topic of tonal description of music audio signals. Her main research interests are related to melodic and tonal description of music audio signals, computer-assisted music analysis, and computational ethnomusicology. She is a Member of the IEEE.

Daniel P.W. Ellis (dpwe@ee.columbia.edu) received the Ph.D. degree in electrical engineering from the Massachusetts Institute of Technology (MIT). He is an associate professor in the Electrical Engineering Department, Columbia University, New York. His Laboratory for Recognition and Organization of Speech and Audio is concerned with all aspects of extracting high-level information from audio, including speech recognition, music description, and environmental sound processing. At MIT, he was a research assistant at the Media Lab, and he spent several years as a research scientist at the International Computer Science Institute, Berkeley, California, where he remains an external fellow. He is a Senior Member of the IEEE.

Gaël Richard (gael.richard@telecom-paristech.fr) received the State Engineering degree from Telecom ParisTech, France, (formerly ENST) in 1990, the Ph.D. degree from LIMSI-CNRS, University of Paris-XI, in 1994 in speech synthesis, and the Habilitation à Diriger des Recherches degree from the University of Paris XI in 2001. He spent two years at the Center for Advanced Information Processing, Rutgers University, Piscataway, New Jersey, and from 1997 to 2001, he successively worked for Matra, Bois d'Arcy, France, and Philips, Montrouge, France. He joined the Department of Signal and Image Processing, Telecom ParisTech, in 2001, where he is now a full professor in audio signal processing and head of the Audio, Acoustics, and Waves research group. He is a coauthor of over 100 papers and patents and is also one of the experts of the European commission in the field of audio signal processing and man/machine interfaces. He is a Senior Member of the IEEE.

REFERENCES

[1] M. Goto and S. Hayamizu, “A real-time music scene description system: Detecting melody and bass lines in audio signals,” in *Working Notes of the IJCAI-99 Workshop on Computational Auditory Scene Analysis*, 1999, pp. 31–40.

[2] G. E. Poliner, D. P. W. Ellis, F. Ehmann, E. Gómez, S. Streich, and B. Ong, “Melody transcription from music audio: Approaches and evaluation,” *IEEE Trans. Audio, Speech, Lang. Processing*, vol. 15, no. 4, pp. 1247–1256, 2007.

[3] J. S. Downie, “The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research,” *Acoust. Sci. Technol.*, vol. 29, no. 4, pp. 247–255, 2008.

[4] R. Typke, “Music retrieval based on melodic similarity,” Ph.D. dissertation, Dept. Inform. Computing Sci., Utrecht University, The Netherlands, 2007.

[5] M. Ryynänen and A. Klapuri, “Automatic transcription of melody, bass line, and chords in polyphonic music,” *Comput. Music J.*, vol. 32, no. 3, pp. 72–86, 2008.

[6] A. L. Ringer. (2013, Jan.). Melody. Grove Music Online, Oxford Music Online. [Online]. Available: www.oxfordmusiconline.com/subscriber/article/grove/music/18357

[7] B. C. J. Moore, *An Introduction to the Psychology of Hearing*, 5th ed. San Diego, CA: Academic Press, 2003.

[8] R. Foucard, J.-L. Durrieu, M. Lagrange, and G. Richard, “Multimodal similarity between musical streams for cover version detection,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2010, pp. 5514–5517.

[9] J. Salamon, J. Serrà, and E. Gómez, “Tonal representations for music retrieval: From version identification to query-by-humming,” *Int. J. Multimedia Inform. Retrieval* (*Special Issue on Hybrid Music Information Retrieval*), vol. 2, no. 1, pp. 45–58, Mar. 2013.

[10] J. Salamon, B. Rocha, and E. Gómez, “Musical genre classification using melody features extracted from polyphonic music signals,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 81–84.

[11] J.-L. Durrieu, G. Richard, and B. David, “An iterative approach to monaural musical mixture de-soloing,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Apr. 2009, pp. 105–108.

[12] A. Mesaros, T. Virtanen, and A. Klapuri, “Singer identification in polyphonic music using vocal separation and pattern recognition methods,” in *Proc. 8th Int. Conf. Music Information Retrieval*, 2007, pp. 375–378.

[13] E. Gómez, F. Cañadas, J. Salamon, J. Bonada, P. Vera, and P. Cabañas, “Predominant fundamental frequency estimation vs. singing voice separation for the automatic transcription of accompanied flamenco singing,” in *Proc. 13th Int. Society for Music Information Retrieval Conf.*, Porto, Portugal, Oct. 2012, pp. 601–606.

[14] G. K. Koduri, J. Serrà, and X. Serra, “Characterization of intonation in carnatic music by parametrizing pitch histograms,” in *Proc. 13th Int. Society for Music Information Retrieval Conf.*, Porto, Portugal, Oct. 2012, pp. 199–204.

[15] A. Pikrakis, F. Gómez, S. Oramas, J. M. D. Báñez, J. Mora, F. Escobar, E. Gómez, and J. Salamon, “Tracking melodic patterns in flamenco singing by analyzing polyphonic music recordings,” in *Proc. 13th Int. Society for Music Information Retrieval Conf.*, Porto, Portugal, Oct. 2012, pp. 421–426.

[16] J. Salamon and E. Gómez, “Melody extraction from polyphonic music signals using pitch contour characteristics,” *IEEE Trans. Audio, Speech, Lang. Processing*, vol. 20, no. 6, pp. 1759–1770, Aug. 2012.

[17] J. Kreiman and D. V. L. Sidtis, *Foundations of Voice Studies: An Interdisciplinary Approach to Voice Production and Perception*. Hoboken, NJ: Wiley-Blackwell, 2011, Ch. 10.

[18] W. Hess, *Pitch Determination of Speech Signals: Algorithms and Devices*. Berlin, Germany: Springer-Verlag, 1983.

[19] 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.

[20] A. Klapuri, “Signal processing methods for the automatic transcription of music,” Ph.D. dissertation, Tampere University of Technology, Finland, Apr. 2004.

[21] G. Poliner and D. Ellis, “A classification approach to melody transcription,” in *Proc. 6th Int. Conf. Music Information Retrieval*, London, Sept. 2005, pp. 161–166.

[22] J.-L. Durrieu, G. Richard, B. David, and C. Févotte, “Source/filter model for unsupervised main melody extraction from polyphonic audio signals,” *IEEE Trans. Audio, Speech, Lang. Processing*, vol. 18, no. 3, pp. 564–575, Mar. 2010.

[23] H. Tachibana, T. Ono, N. Ono, and S. Sagayama, “Melody line estimation in homophonic music audio signals based on temporal-variability of melodic source,” in *IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2010, pp. 425–428.

[24] J. Sundberg, *The Science of the Singing Voice*. DeKalb, IL: Northern Illinois Univ. Press, 1987.

[25] R. P. Paiva, T. Mendes, and A. Cardoso, “Melody detection in polyphonic musical signals: Exploiting perceptual rules, note salience, and melodic smoothness,” *Comput. Music J.*, vol. 30, pp. 80–98, Dec. 2006.

[26] M. Marolt, “On finding melodic lines in audio recordings,” in *Proc. 7th Int. Conf. Digial Audio Effects (DAFx’04)*, Naples, Italy, Oct. 2004, pp. 217–221.

[27] M. Goto, “A real-time music-scene-description system: predominant-f0 estimation for detecting melody and bass lines in real-world audio signals,” *Speech Commun.*, vol. 43, no. 4, pp. 311–329, Sept. 2004.

[28] P. Cancela, “Tracking melody in polyphonic audio,” in *Proc. 4th Music Information Retrieval Evaluation eXchange (MIREX)*, 2008.

[29] K. Dressler, “An auditory streaming approach for melody extraction from polyphonic music,” in *Proc. 12th Int. Society for Music Information Retrieval Conf.*, Miami, FL, Oct. 2011, pp. 19–24.

[30] V. Rao and P. Rao, “Vocal melody extraction in the presence of pitched accompaniment in polyphonic music,” *IEEE Trans. Audio, Speech, Lang. Processing*, vol. 18, no. 8, pp. 2145–2154, Nov. 2010.

[31] S. Jo, S. Joo, and C. D. Yoo, “Melody pitch estimation based on range estimation and candidate extraction using harmonic structure model,” in *Proc. InterSpeech*, Makuhari, Japan, Sept. 2010, pp. 2902–2905.

[32] V. Arora and L. Behera, “On-line melody extraction from polyphonic audio using harmonic cluster tracking,” *IEEE Trans. Audio, Speech, Lang. Processing*, vol. 21, no. 3, pp. 520–530, Mar. 2013.

[33] C. Hsu and J. R. Jang, “Singing pitch extraction by voice vibrato/tremolo estimation and instrument partial deletion,” in *Proc. 11th Int. Society for Music Information Retrieval Conf.*, Utrecht, The Netherlands, Aug. 2010, pp. 525–530.

[34] T.-C. Yeh, M.-J. Wu, J.-S. Jang, W.-L. Chang, and I.-B. Liao, “A hybrid approach to singing pitch extraction based on trend estimation and hidden Markov models,” in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 457–460.

[35] C. Sutton, “Transcription of vocal melodies in popular music,” Master’s thesis, Queen Mary, School of Electron. Eng. Comput. Sci., University of London, United Kingdom, 2006.

[36] K. Dressler, “Sinusoidal extraction using an efficient implementation of a multi-resolution FFT,” in *Proc. 9th Int. Conf. Digital Audio Effects (DAFx-06)*, Montreal, Canada, Sept. 2006, pp. 247–252.

[37] K. Dressler, “Pitch estimation by the pair-wise evaluation of spectral peaks,” in *Proc. AES 42nd Int. Conf.*, Ilmenau, Germany, July 2011, pp. 278–290.

[38] A. Ozerov, P. Philippe, F. Bimbot, and R. Gribonval, “Adaptation of Bayesian models for single-channel source separation and its application to voice/music separation in popular songs,” *IEEE Trans. Audio, Speech, and Lang. Processing*, vol. 15, no. 5, pp. 1564–1578, July 2007.

[39] Y. Li and D. Wang, “Separation of singing voice from music accompaniment for monaural recordings,” *IEEE Trans. Audio, Speech, Lang. Processing*, vol. 15, no. 4, pp. 1475–1487, May 2007.

[40] J.-L. Durrieu, B. David, and G. Richard, “A musically motivated mid-level representation for pitch estimation and musical audio source separation,” *IEEE J. Select. Topics Signal Processing*, vol. 5, no. 6, pp. 1180–1191, Oct. 2011.

[41] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson, “Singing voice separation from monaural recordings using robust principal component analysis,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 57–60.

[42] Z. Rafii and B. Pardo, “Repeating pattern extraction technique (REPET): A simple method for music/voice separation,” *IEEE Trans. Audio, Speech, Lang. Processing*, vol. 21, no. 1, pp. 71–82, Jan. 2013.

[43] A. Liutkus, Z. Rafii, R. Badeau, B. Pardo, and G. Richard, “Adaptive filtering for music/voice separation exploiting the repeating musical structure,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 53–56.

[44] N. Ono, K. Miyamoto, H. Kameoka, J. Le Roux, Y. Uchiyama, E. Tsunoo, T. Nishimoto, and S. Sagayama, “Harmonic and percussive sound separation and its application to MIR-related tasks,” in *Advances in Music Information Retrieval* (Studies in Computational Intelligence, vol. 274), Z. Ras and A. Wieczorkowska, Eds. Berlin, Germany: Springer, 2010, pp. 213–236.

[45] P. Cancela, E. López, and M. Rocamora, “Fan chirp transform for music representation,” in *Proc. 13th Int. Conf. Digital Audio Effects (DAFx)*, Graz, Austria, Sept. 2010, pp. 54–61.

[46] W.-H. Tsai, H.-M. Yu, and H.-M. Wang, “Using the similarity of main melodies to identify cover versions of popular songs for music document retrieval,” *J. Inform. Sci. Eng.*, vol. 24, no. 6, pp. 1669–1687, 2008.

[47] M. Marolt, “A mid-level representation for melody-based retrieval in audio collections,” *IEEE Trans. Multimedia,* vol. 10, no. 8, pp. 1617–1625, Dec. 2008.

[48] J. Song, S. Y. Bae, and K. Yoon, “Mid-level music melody representation of polyphonic audio for query-by-humming system,” in *Proc. 3rd Int. Conf. Music Information Retrieval*, Paris, France, Oct. 2002, pp. 133–139.

[49] M. Ryynänen and A. Klapuri, “Query by humming of MIDI and audio using locality sensitive hashing,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Apr. 2008, pp. 2249–2252.

[50] J. Salamon and J. Urbano, “Current challenges in the evaluation of predominant melody extraction algorithms,” in *Proc. 13th Int. Society for Music Information Retrieval Conf.*, Porto, Portugal, Oct. 2012, pp. 289–294.