Melody Extraction from Polyphonic Music Signals using Pitch Contour Characteristics

Justin Salamon* and Emilia Gómez

Abstract—We present a novel system for the automatic extraction of the main melody from polyphonic music recordings. Our approach is based on the creation and characterisation of pitch contours, time continuous sequences of pitch candidates grouped using auditory streaming cues. We define a set of contour characteristics and show that by studying their distributions we can devise rules to distinguish between melodic and non-melodic contours. This leads to the development of new voicing detection, octave error minimisation and melody selection techniques.

A comparative evaluation of the proposed approach shows that it outperforms current state-of-the-art melody extraction systems in terms of overall accuracy. Further evaluation of the algorithm is provided in the form of a qualitative error analysis and the study of the effect of key parameters and algorithmic components on system performance. Finally, we conduct a glass ceiling analysis to study the current limitations of the method, and possible directions for future work are proposed.

Index Terms—Predominant Melody Estimation, Multi-pitch Estimation, Music Information Retrieval, Audio Content Description, Pitch Contour.

I. INTRODUCTION

A. Definition and Motivation

GIVEN the audio recording of a piece of polyphonic music, the task of melody extraction involves automatically extracting a representation of the main melodic line. By polyphonic we refer to music in which two or more notes can sound simultaneously, be it 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). To define the extracted melody representation, we must first have a clear definition of what the *main melody* actually is. As stated in [1], the term *melody* is a musicological concept based on the judgement of human listeners, and we can expect to find different definitions for the melody in different contexts [2], [3]. In order to have a clear framework to work within, the Music Information Retrieval (MIR) community has adopted in recent years the definition proposed by [1], "...*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 recognise as being the 'essence' of that music when heard in comparison*". We use this definition for the purpose of this study and, as in previous studies, select the evaluation material such that given the above definition human listeners could easily agree on what the main

J. Salamon and E. Gómez are with the Music Technology Group (MTG) of the Department of Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, Spain, e-mail: {justin.salamon,emilia.gomez}@upf.edu

melody is, regardless of the musical genre of the piece. This is important as it allows us to generate an objective ground truth in order to quantitatively compare different approaches.

The melody representation used in this study is the one proposed by [4], namely a sequence of fundamental frequency (F0) values corresponding to the perceived pitch of the main melody. It is important to note that whilst pitch and F0 are different concepts (the former being perceptual and the latter a physical quantity), as common to the melody extraction literature we will use the term pitch to refer to the F0 of the melody. As argued in [4] such a mid-level description (avoiding transcription into, for example, Western score notation) has many potential applications such as Query by Humming [5], music de-soloing for the automatic generation of karaoke accompaniment [6] and singer identification [7], to name a few. Determining the melody of a song could also be used as an intermediate step towards the derivation of semantic labels from musical audio [8]. Note that we consider not only sung melodies but also those played by instruments, for example a jazz standard in which the melody is played by a saxophone.

B. Related Work

Many methods for melody extraction have been proposed. Of these perhaps the largest group are what could be referred to as *salience-based* methods, which derive an estimation of pitch salience over time and then apply tracking or transition rules to extract the melody line without separating it from the rest of the audio [3], [4], [9], [10]. Such systems follow a common structure – first a spectral representation of the signal is obtained. The spectral representation is used to compute a time-frequency representation of pitch salience, also known as a *salience function*. The peaks of the salience function are considered as potential F0 candidates for the melody. Different approaches exist for computing the salience function, [11] uses harmonic summation with weighting learned from instrument training data, whilst [4] lets different F0s compete for harmonics, using Expectation-Maximization (EM) to re-estimate a set of unknown harmonic-model weights. Finally, the melody F0s are selected using different methods of peak selection or tracking. In some systems a subsequent voicing detection step (determining whether the main melody is present or absent in each time frame) is also included. A detailed review of such systems is provided in [1].

Another set of approaches attempt to identify the melody by separating it from the rest of the audio using timbre-based source separation techniques [12], [13]. Such systems use two separate timbre models, one for the melody (sometimes specifically human singing voice) and the other for the accompaniment. Some systems incorporate grouping principles inspired

by auditory scene analysis (ASA), most often frequency proximity [4]. Other grouping principles have also been exploited – in [14] grouping principles based on frequency and amplitude proximity and harmonicity are incorporated into a separation framework based on spectral clustering, where a monophonic pitch tracker is later applied to the separated melody source.

Algorithms that exploit the spatial information in stereo recordings have also been proposed. In [15] stereophonic information is used to estimate the panning of each source, and a production model (source/filter) is used to identify and separate the melody. Melody extraction is used as an intermediate step to tune the separation parameters to the estimated melody. Finally, purely data-driven approaches have also been studied, such as [16] in which the entire short-time magnitude spectrum is used as training data for a support vector machine classifier.

Despite the variety of proposed approaches, melody extraction remains a challenging and unsolved task, with current state-of-the-art systems achieving overall accuracies[1] of around 70%[2]. The complexity of the task is twofold – firstly, the signal representation of polyphonic music contains the superposition of all instruments which play simultaneously. When considering the spectral content of the signal, the harmonics of different sources superimpose making it very hard to attribute specific frequency bands and energy levels to specific instrument notes. This is further complicated by mixing and mastering techniques such as adding reverberation (blurs note offsets) or applying dynamic range compression (reduces the difference between soft and loud sources, increasing interference). Secondly, even once we obtain a pitch-based representation of the signal, the task of determining which pitches constitute the main melody needs to be solved [17]. This in turn entails three main challenges – determining when the melody is present and when it is not (voicing detection), ensuring the estimated pitches are in the correct octave (avoiding octave errors), and selecting the correct melody pitch when there is more than one note sounding simultaneously.

C. Method Introduction, Contributions and Paper Outline

Though promising results have been achieved recently by separation-based methods [13], salience-based approaches are still amongst the best performing systems, as well as being conceptually simple and computationally efficient. In this paper a novel salience-based melody extraction method is presented. The method is centred on the creation and characterisation of *pitch contours* – time continuous sequences of F0 candidates generated and grouped using heuristics based on auditory streaming cues [18] such as harmonicity, pitch continuity and exclusive allocation. We define a set of musical features which are automatically computed for each contour. By studying the feature distributions of melodic and non-melodic contours we are able to define rules for distinguishing between the contours that form the melody and contours that should be filtered out. Combining these rules with voice

leading principles [19], novel techniques are developed for addressing the challenges mentioned earlier – voicing detection, avoiding octave errors and selecting the pitch contours that belong to the main melody.

The idea of F0 candidate grouping (or tracking) is not new to the literature [10], [20]. ASA inspired grouping principles have been employed in melody extraction systems based on source separation [14], as well as in [9] where pitch contours are first segmented into notes out of which the melody is selected. While the structure of our system is somewhat similar, the presented method differs in several important ways. To begin with, a wider set of contour characteristics beyond the basic pitch height, length and mean salience is considered. The method does not require segmentation into notes, and makes use of contour features that would be lost during pitch quantisation such as vibrato and pitch deviation. Furthermore, these features are exploited using new techniques following the study of contour feature distributions.

The main contribution of the paper is the contour characterisation and its application for melodic filtering. The contribution can be summarised as follows: a method for the generation and characterisation of pitch contours is described, which uses signal processing steps and a salience function specifically designed for the task of melody extraction. A set of pitch contour features is defined and their distributions are studied, leading to novel methods for voicing detection, octave error minimisation and melody selection.

In addition to the main contribution, a comparative evaluation with state-of-the-art systems is provided, including a statistical analysis of the significance of the results. We also study the effect of optimising individual stages of the system [21] on its overall performance, and assess the influence of different algorithmic components. These evaluations are complemented with a qualitative error analysis and glass ceiling analysis to determine the current limitations of the approach and propose directions for future work.

The outline of the remainder of the paper is as follows. In Section II the proposed melody extraction method is described. In Section III the evaluation process is described, including the test collections and metrics used for evaluation. In Section IV the results of the evaluation are presented, followed by a qualitative error analysis, component evaluation and a glass ceiling study. Finally in Section V we conclude with a discussion of the proposed method and the obtained results, providing some suggestions for future improvements as well as a discussion on the remaining challenges in melody extraction.

II. PROPOSED METHOD

Our approach is comprised of four main blocks, as depicted in Figure 1. In the following sections we describe each of the four blocks in detail. The motivation for choosing specific processing steps and parameter values for the first two blocks of the system is discussed in Section IV-C.

A. Sinusoid Extraction

The sinusoid extraction process is divided into three stages as depicted in Figure 1: filtering, spectral transform and sinusoid frequency/amplitude correction.

---

[1]Overall accuracy is defined in Section III-B.

[2]Music Information Retrieval Evaluation eXchange [Online]. Available: http://www.music-ir.org/mirex/wiki/Audio_Melody_Extraction (Dec. 2011).
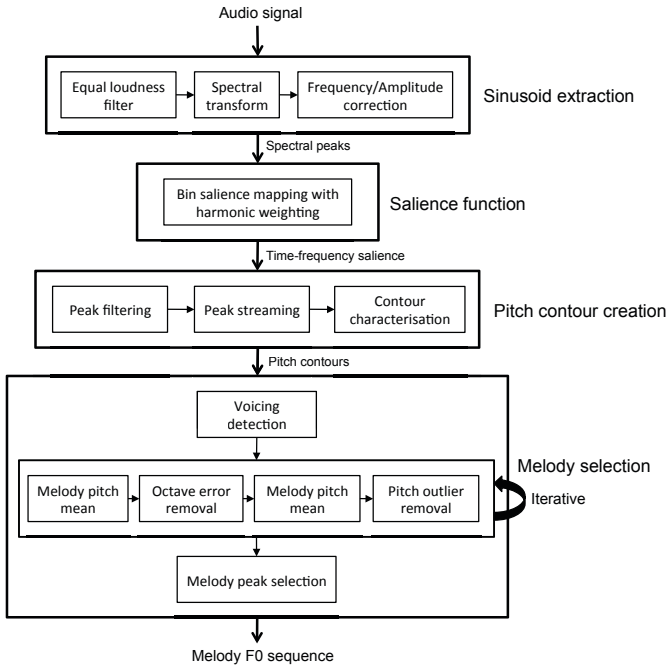
Fig. 1. Block diagram of the system's four main blocks: sinusoid extraction, salience function computation, pitch contour creation and melody selection.

1) Filtering: We apply an equal loudness filter [22], which enhances frequencies to which the human listener is more perceptually sensitive (and attenuates those to which they are not). This is done by taking a representative average of the equal loudness curves [23] and filtering the signal by its inverse. The filter is implemented as a 10th order IIR filter cascaded with a 2nd order Butterworth high pass filter (for further details see [22]). The filter is well suited for melody extraction, as it enhances mid-band frequencies where we can expect to find the melody, and attenuates low-band frequencies where low pitched instruments (e.g. the bass) can be found.

2) Spectral Transform: After filtering, we apply the Short-Time Fourier Transform (STFT) given by:

$$X_l(k) = \sum_{n=0}^{M-1} w(n) \cdot x(n + lH) e^{-j\frac{2\pi}{N}kn}, \quad (1)$$
$$l = 0, 1, \ldots \text{ and } k = 0, 1, \ldots, N-1$$

where $x(n)$ is the time signal, $w(n)$ the windowing function, l the frame number, M the window length, N the FFT length and H the hop size. We use the Hann windowing function with a window size of 46.4ms, a hop size of 2.9ms and a ×4 zero padding factor, which for data sampled at $f_S = 44.1$kHz gives $M = 2048$, $N = 8192$ and $H = 128$. The relatively small hop size (compared to other MIR tasks [24]) is selected to facilitate more accurate F0 tracking during the creation of pitch contours. The decision to use the STFT rather than some type of multi-resolution transform [4], [10], [20] is justified in Section IV-C. Given the FFT of a single frame $X_l(k)$, spectral peaks p_i are selected by finding all the local maxima k_i of the magnitude spectrum $|X_l(k)|$.

3) Frequency/Amplitude Correction: The location of the spectral peaks is limited to the bin frequencies of the FFT,

which for low frequencies can result in a relatively large error in the estimation of the peak frequency. To overcome this quantisation we use the approach described in [25], in which the phase spectrum $\phi_l(k)$ is used to calculate the peak's instantaneous frequency (IF) and amplitude, which provide a more accurate estimate of the peak's true frequency and amplitude. The choice of this correction method over alternative approaches is explained in Section IV-C.

The IF $\hat{f}_i$ of a peak p_i found at bin k_i is computed from the phase difference $\Delta(k)$ of successive phase spectra using the phase vocoder [26] method as follows:

$$\hat{f}_i = (k_i + \kappa(k_i)) \frac{f_S}{N}, \quad (2)$$

where the bin offset $\kappa(k_i)$ is calculated as:

$$\kappa(k_i) = \frac{N}{2\pi H} \Psi\left(\phi_l(k_i) - \phi_{l-1}(k_i) - \frac{2\pi H}{N}k_i\right), \quad (3)$$

where Ψ is the principal argument function which maps the phase to the $\pm\pi$ range. The instantaneous magnitude $\hat{a}_i$ is calculated using the peak's spectral magnitude $|X_l(k_i)|$ and the bin offset $\kappa(k_i)$ as follows:

$$\hat{a}_i = \frac{1}{2} \frac{|X_l(k_i)|}{W_{Hann}\left(\frac{M}{N}\kappa(k_i)\right)}, \quad (4)$$

where W_{Hann} is the Hann window kernel.

B. Salience Function Computation

The extracted spectral peaks are used to construct a salience function – a representation of pitch salience over time. The peaks of this function form the F0 candidates for the main melody. The salience computation in our system is based on harmonic summation similar to [11], where the salience of a given frequency is computed as the sum of the weighted energies found at integer multiples (harmonics) of that frequency. Unlike [11], only the spectral peaks are used in the summation, to discard spectral values which are less reliable due to masking or noise. Using the peaks also allows us to apply the aforementioned frequency correction which has been shown to improve the frequency accuracy of the salience function [21].

The important factors affecting the salience computation are the number of harmonics considered N_h and the weighting scheme used. The choice of these parameters is discussed in the results section of the paper (IV-C).

Our salience function covers a pitch range of nearly five octaves from 55Hz to 1.76kHz, quantized into $b = 1 \ldots 600$ bins on a cent scale (10 cents per bin). Given a frequency $\hat{f}$ in Hz, its corresponding bin $B(\hat{f})$ is calculated as:

$$B(\hat{f}) = \left\lfloor \frac{1200 \cdot \log_2\left(\frac{\hat{f}}{55}\right)}{10} + 1 \right\rfloor. \quad (5)$$

At each frame the salience function $S(b)$ is constructed using the spectral peaks p_i (with frequencies $\hat{f}_i$ and linear magnitudes $\hat{a}_i$) returned by the sinusoid extraction step ($i = $

$1 \ldots I$, where I is the number of peaks found). The salience function is defined as:

$$S(b) = \sum_{h=1}^{N_h} \sum_{i=1}^{I} e(\hat{a}_i) \cdot g(b,h,\hat{f}_i) \cdot (\hat{a}_i)^{\beta}, \qquad (6)$$

where β is a magnitude compression parameter, $e(\hat{a}_i)$ is a magnitude threshold function, and $g(b,h,\hat{f}_i)$ is the function that defines the weighting scheme. The magnitude threshold function is defined as:

$$e(\hat{a}_i) = \begin{cases} 1 & \text{if } 20\log_{10}(\hat{a}_M/\hat{a}_i) < \gamma, \\ 0 & \text{otherwise}, \end{cases} \qquad (7)$$

where $\hat{a}_M$ is the magnitude of the highest spectral peak in the frame and γ is the maximum allowed difference (in dB) between $\hat{a}_i$ and $\hat{a}_M$. The weighting function $g(b,h,\hat{f}_i)$ defines the weight given to peak p_i, when it is considered as the hth harmonic of bin b:

$$g(b,h,\hat{f}_i) = \begin{cases} \cos^2(\delta \cdot \frac{\pi}{2}) \cdot \alpha^{h-1} & \text{if } |\delta| \le 1, \\ 0 & \text{if } |\delta| > 1, \end{cases} \qquad (8)$$

where $\delta = |B(\hat{f}_i/h) - b|/10$ is the distance in semitones between the harmonic frequency $\hat{f}_i/h$ and the centre frequency of bin b, and α is the harmonic weighting parameter. The non-zero threshold for δ means that each peak contributes not just to a single bin of the salience function but also to the bins around it (with $\cos^2$ weighting). This avoids potential problems that could arise due to the quantization of the salience function into bins, and also accounts for inharmonicities. In the results section of this paper (IV-C) we discuss the optimisation of the aforementioned parameters for melody extraction, and examine the effect it has on the global performance of the system, comparing melody extraction results before and after parameter optimisation.

C. Creating Pitch Contours (Peak Streaming)

Once the salience function is computed, its peaks at each frame are selected as potential melody F0 candidates. At this stage, some melody extraction methods attempt to track the melody directly from the set of available peaks [4], [27]. Our approach however is based on the idea that further information (which can be exploited to select the correct melody pitch) can be extracted from the data by first grouping the peaks into pitch contours – time and pitch continuous sequences of salience peaks. Each contour has a limited time span corresponding roughly to a single note in the shortest case or a short phrase in the longest. Though F0 grouping is not a new concept [9], [20], in this paper the characterisation of pitch contours is explored in new ways, resulting in original solutions to the challenges mentioned in Section I-B.

Before the streaming process is carried out, we first filter out non-salient peaks to minimise the creation of "noise" contours (non-melody contours). The filtering process is carried out in two stages: first, peaks are filtered on a per frame basis by comparing their salience to that of the highest peak in the current frame. Peaks below a threshold factor τ_+ of the salience of the highest peak are filtered out. In the second stage the salience mean μ_s and standard deviation σ_s of all remaining peaks (in all frames) are computed. Peaks with salience below $\mu_s - \tau_\sigma \cdot \sigma_s$ are then filtered out, where τ_σ determines the degree of deviation below the mean salience accepted by the filter. The first filter ensures we only focus on the most predominant pitch candidates at each frame, whilst the second, a precursor to our voicing detection method, removes peaks in segments of the song which are generally weaker (and more likely to be unvoiced). This filtering has an inherent trade-off – the more peaks we filter out the less noise contours will be created (thus improving the detection of non-voiced segments and the correct selection of melody contours), however the greater the risk of filtering out salience peaks which belong to the melody (henceforth "melody peaks"). The selection of optimal values for τ_+ and τ_σ is discussed at the end of this section.

The remaining peaks are stored in the set S^+, whilst the peaks that were filtered out are stored in S^-. The peaks are then grouped into contours in a simple process using heuristics based on auditory streaming cues [18]. We start by selecting the highest peak in S^+ and add it to a new pitch contour. We then track forward in time by searching S^+ for a salience peak located at the following time frame (time continuity cue) which is within 80 cents (pitch continuity cue) from the previously found peak. A matching peak is added to the pitch contour and removed from S^+ (exclusive allocation principle). This step is repeated until no further matching salience peaks are found. During the tracking we must ensure that short time gaps in the pitch trajectory do not split what should be a single contour into several contours. To do so, once no matching peak is found in S^+, we allow the tracking to continue for a limited amount of frames using peaks from S^-. The underlying assumption is that melody peaks whose salience is temporarily masked by other sources will be stored in S^-, and tracking them allows us to stay on the correct trajectory until we find a peak in S^+. If the gap length exceeds 100ms (see below for selection of threshold and parameter values) before a peak from S^+ is found the tracking is ceased. We then go back to the first peak of the contour and repeat the tracking process backwards in time. Once the tracking is complete we save the contour and the entire process is repeated until there are no peaks remaining in S^+.

To select the best parameters for the contour creation (τ_+, τ_σ, the maximum allowed pitch distance and gap length), we compared contours generated from different excerpts to the excerpts' melody ground truth and evaluated them in terms of pitch accuracy (distance in cents between the ground truth and the contours) and voicing (i.e. whether the contours exactly cover the ground truth or are otherwise too long or too short). This process was repeated in a grid search until the parameters which resulted in the most accurate tracking were found (0.9, 0.9, 80 cents and 100ms respectively). For τ_+ and τ_σ we also measured the amount of melody peaks (and non-melody peaks) before and after the filtering. This analysis revealed that as τ_+ is increased the number of non-melody salience peaks drops dramatically, whilst the number of melody peaks reduces very gradually. Using the selected parameter values

the number of non-melody peaks is reduced by 95% whilst melody peaks are reduced by less than 17% (and this loss can be recovered by the gap tracking). The result is that the percentage of melody peaks out of the total number of peaks goes up on average from 3% initially to 52% after filtering. The quality of contour formation is discussed in Section IV-G.

D. Pitch Contour Characterisation

Once the contours are created, the remaining challenge is that of determining which contours belong to the melody. To do so, a set of contour characteristics is defined which will be used to guide the system in selecting melody contours. Similarly to other systems, we define features based on contour pitch, length and salience. However, by avoiding the quantisation of contours into notes [9] we are able to extend this set by introducing features extracted from the pitch trajectory of the contour, namely its pitch deviation and the presence of vibrato. Note that whilst [9] also keeps a non-quantised version of each contour for use at a later stage of the algorithm, it does not exploit it to compute additional contour features. Furthermore, as shall be seen in the next section, we use not only the feature values directly but also their distributions. The characteristics computed for each contour are the following:

- **Pitch mean** $C_{\bar{p}}$**:** the mean pitch height of the contour.
- **Pitch deviation** C_{σ_p}**:** the standard deviation of the contour pitch.
- **Contour mean salience** $C_{\bar{s}}$**:** the mean salience of all peaks comprising the contour.
- **Contour total salience** $C_{\Sigma s}$**:** the sum of the salience of all peaks comprising the contour.
- **Contour salience deviation** C_{σ_s}**:** the standard deviation of the salience of all peaks comprising the contour.
- **Length** C_l**:** the length of the contour.
- **Vibrato presence** C_v**:** whether the contour has vibrato or not (true/false). Vibrato is automatically detected by the system using a method based on [28]: we apply the FFT to the contour's pitch trajectory (after subtracting the mean) and check for a prominent peak in the expected frequency range for human vibrato (5-8Hz).

In Figure 2 we provide examples of contours created for excerpts of different musical genres (the relative sparseness of non-melody contours can be attributed to the equal loudness filter and salience peak filtering described earlier). By observing these graphs we can propose contour characteristics that differentiate the melody from the rest of the contours: vibrato, greater pitch variance (in the case of human voice), longer contours, a mid-frequency pitch range and (though not directly visible in the graphs) greater salience. These observations concur with voice leading rules derived from perceptual principles [19]. To confirm our observations, we computed the feature distributions for melody and non-melody contours using the representative data-set described in Section III-A3. Note that in most (but not all) of the excerpts in this data-set the melody is sung by a human voice. The resulting distributions are provided in Figure 3, where for each feature we plot the distribution for melody contours (solid red line) and non-melody contours (dashed blue line). In

plots (c), (d) and (e) the feature values are normalised by the mean feature value for each excerpt. We see that the above observations are indeed evident in the feature distributions. Additionally, for vibrato presence we found that 95% of all contours in which vibrato was detected were melody contours. The consideration of various contour characteristics means accompanying instruments will not necessarily be selected as melody if they exhibit a certain melodic characteristic. For example, a contour produced by an accompanying violin with vibrato may still be discarded due to its pitch height. Finally, we note that basing our system on pitch contours gives us the possibility of introducing new contour features in the future, as well as using these features for other MIR tasks such as genre classification [29] or singing style characterisation.
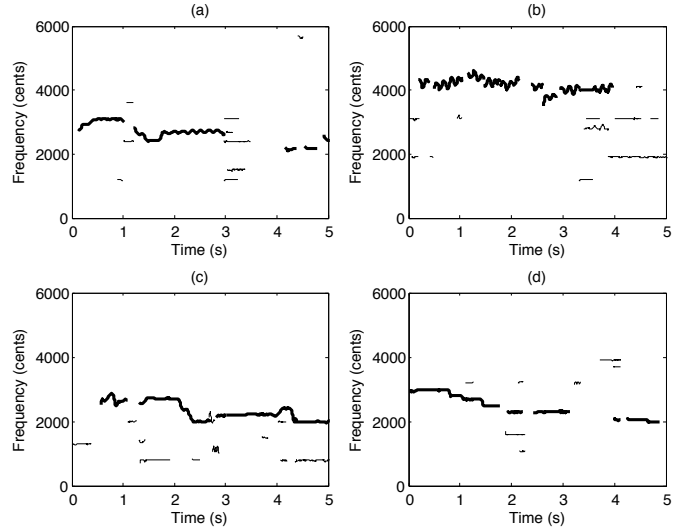
Fig. 2. Pitch contours generated from excerpts of (a) vocal jazz, (b) opera, (c) pop and (d) instrumental jazz. Melody contours are highlighted in bold.
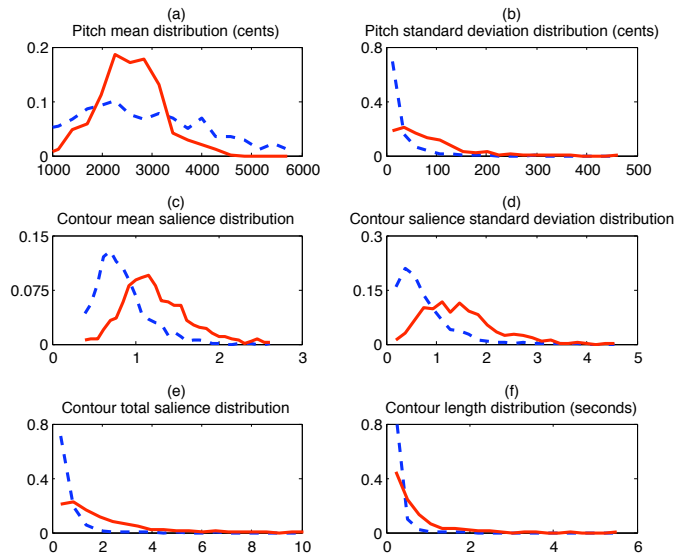
Fig. 3. Pitch contour feature distributions: (a) Pitch mean, (b) Pitch std. dev., (c) Mean salience, (d) Salience std. dev., (e) Total salience, (f) Length. The red solid line represents the distribution of melody contour features, the blue dashed line represents the distribution of non-melody contour features.

E. Melody Selection

We now turn to describe how the melody is chosen out of all the contours created in the previous step of our method. Rather than selecting melody contours, we pose this task as a contour filtering problem, where our goal is to filter out all non-melody contours. As seen in the block diagram in Figure 1, this process is comprised of three steps: voicing detection, octave error minimisation/pitch outlier removal, and final melody selection. In the first two steps contour characteristics are used to filter out non-melody contours, and in the final step the melody frequency at each frame is selected out of the remaining contours.

1) Voicing Detection: Voicing detection is the task of determining when the melody is present and when it is not. For example in plot (a) of Figure 2 the melody is present between seconds 0-3 and 4-5, but not between 3-4 where non-melody contours are found. To filter out these contours we take advantage of the contour mean salience distribution given in plot (c) of Figure 3. Though the distributions are not perfectly separated, we see that by setting a threshold slightly below the average contour mean salience of all contours in the excerpt $\overline{C_{\overline{s}}}$, we can filter out a considerable amount of non-melody contours with little effect on melody contours. We define the following voicing threshold τ_v based on the distribution mean $\overline{C_{\overline{s}}}$ and its standard deviation $\sigma_{C_{\overline{s}}}$:

$$\tau_v = \overline{C_{\overline{s}}} - \nu \cdot \sigma_{C_{\overline{s}}} \tag{9}$$

The parameter ν determines the lenience of the filtering – a high ν value will give more false positives (i.e. false melody contours) and low value more false negatives (i.e. filter out melody contours). The sensitivity of the system to the value of ν is discussed in Section IV-E. We also compared using the contour total salience $C_{\Sigma s}$ instead of the mean salience in the equation above, but the latter was found to give better results. This is likely due to the bias of the contour total salience towards longer contours, which is not beneficial at this stage as we risk removing short melody contours. At a later stage length will be exploited to guide the system when a choice must be made between alternative concurrent contours.

In the previous section we also noted that if the system detected vibrato in a contour, it was almost certainly a melody contour. Furthermore, in plot (b) of Figure 3 we see that there is a sudden drop in non-melody contours once the pitch deviation goes above 20 cents, and once the deviation is greater than 40 cents the probability of a contour being a non-melody contour is less than 5%. We use this information to tune our voicing filter, by giving "immunity" to contours where vibrato was detected ($C_v = true$) or whose pitch deviation is above 40 cents ($C_{\sigma_p} > 40$). In this way we ensure that contours which have relatively low salience but strong melodic characteristics are not filtered out at this stage.

2) Octave Errors and Pitch Outliers: One of the main sources of errors in melody extraction systems is the selection of a harmonic multiple/sub-multiple of the correct melody F0 instead of the correct F0, commonly referred to as octave errors. Various approaches have been proposed for the minimisation of octave errors, usually performed directly after the calculation of the salience function and on a per-frame basis [20], [30]. When we consider a single frame in isolation, determining whether two salience peaks with a distance of one octave between them were caused by two separate sources or whether they are both the result of the same source (one peak being a multiple of the other) can prove a difficult task. On the other hand, once we have created the pitch contours, detecting the presence of octave duplicates becomes a relatively straight forward task, as these manifest themselves as contours with practically identical trajectories at a distance of one octave from each other. In practice, to compare contour trajectories we compute the distance between their pitch values on a per-frame basis for the region in which they overlap, and compute the mean over this region. If the mean distance is within 1200 ± 50 cents, the contours are considered octave duplicates. An example of octave duplicates can be observed in Figure 2 plot (b) between seconds 3-4s where the correct contour is at about 4000 cents and the duplicate at about 2800 cents.

In this paper we propose a method for octave error minimisation that takes advantage of this type of temporal information in two ways. Firstly, as mentioned above, we use the creation of pitch contours to detect octave duplicates by comparing contour trajectories. Secondly, we use the relationship between neighbouring contours (in time) to decide which of the duplicates is the correct one. Our approach is based on two assumptions: firstly, that most (though not all) of the time the correct contour will have greater salience than its duplicate (the salience function parameters were optimised to this end). Secondly, that melodies tend to have a continuous pitch trajectory avoiding large jumps, in accordance with voice leading principles [19].

To implement these principles, we iteratively calculate a "melody pitch mean" $\overline{P(t)}$, i.e. a pitch trajectory that represents the large scale time evolution of the melody's pitch. When octave duplicates are encountered, the assumption is that the contours directly before and after the duplicates will pull $\overline{P(t)}$ towards the duplicate at the correct octave. Thus, the duplicate closest to $\overline{P(t)}$ is selected as the correct contour and the other is discarded. Similarly, we use $\overline{P(t)}$ to remove "pitch outliers" – contours more than one octave above or below the pitch mean. Filtering outliers ensures there are no large jumps in the melody (continuity assumption), and may also filter out non-voiced contours that were not captured by the voicing detection algorithm. The distance between a contour and $\overline{P(t)}$ is computed as before, by averaging the per-frame distances between them. The complete process can be summarised as:

1) Calculate $\overline{P(t)}$ at each frame as the weighted mean of the pitch of all contours present in the frame.
2) Smooth $\overline{P(t)}$ using a 5-second sliding mean filter (length determined empirically) with a hop size of 1 frame. This limits the rate at which the melody pitch trajectory can change, ensuring continuity and avoiding large jumps.
3) Detect pairs of octave duplicates and, for each pair, remove the contour furthest from $\overline{P(t)}$.
4) Recompute $\overline{P(t)}$ using the remaining contours, following Steps 1-2.
5) Remove pitch outliers by deleting contours at a distance of more than one octave from $\overline{P(t)}$.

6) Recompute $\overline{P(t)}$ using the remaining contours, following Steps 1-2.

7) Repeat Steps 3-6 twice more, each time starting with all contours that passed the voicing detection stage, but using the most recently computed melody pitch mean $\overline{P(t)}$. The number of iterations was chosen following experimentation suggesting this was sufficient for obtaining a good approximation of the true trajectory of the melody. In the future we intend to replace the fixed iteration number by a stabilisation criterion.

8) Pass the contours remaining after the last iteration to the final melody selection stage.

It was found that the pitch mean $\overline{P(t)}$ computed in Step 1 most closely approximates the true trajectory of the melody when each contour's contribution is weighted by its total salience $C_{\Sigma s}$. This biases the mean towards contours which are salient for a longer period of time, which is desirable since such contours are more likely to belong to the melody, as evident from the distributions in Figure 3 (e) and (f).

An example of running steps 1-6 is provided in Figure 4. In plot (a) we start with a set of contours, together with the smoothed melody pitch mean $\overline{P(t)}$ (Steps 1-2) represented by the dashed red line. In the next plot (b), octave duplicates are detected, and the duplicate farther from the melody pitch mean is removed (Step 3). Next (c) the mean $\overline{P(t)}$ is recomputed (Step 4), and pitch outliers are detected and removed (Step 5). Finally $\overline{P(t)}$ is recomputed once more (Step 6), displayed in plot (d) together with the remaining contours.
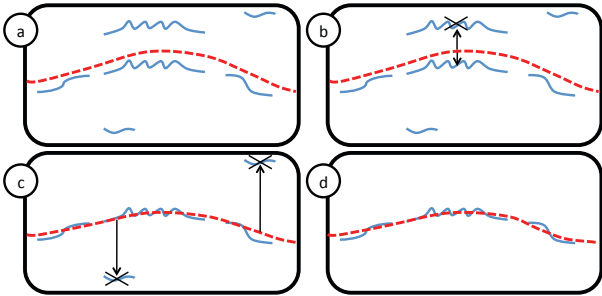

Fig. 4. Removing octave duplicates and pitch outliers. (a) Steps 1-2: the initial smoothed melody pitch mean $\overline{P(t)}$ is computed (dashed red line). (b) Step 3: an octave duplicate is detected and removed. (c) Steps 4-5: $\overline{P(t)}$ is recomputed and two pitch outliers are removed. (d) Step 6: $\overline{P(t)}$ is recomputed.

3) Final Melody Selection: In this final step we need to select from the remaining contours the peaks which belong to the main melody (recall that each peak represents an F0 candidate). Whilst in other systems this step often involves fairly complicated peak tracking using streaming rules or note transition models, in our system these considerations have already been taken into account by the contour creation, characterisation and filtering process. This means that often there will only be one peak to choose. When there is still more than one contour present in a frame, the melody is selected as the peak belonging to the contour with the highest total salience $C_{\Sigma s}$. If no contour is present the frame is regarded as unvoiced. In order to evaluate raw pitch and chroma accuracy (see III-B) we also provide an F0 estimate for unvoiced frames

by selecting the peak of the most salient contour that was present in these frames prior to contour filtering. In Figure 5 we provide an example of the complete melody extraction process for the excerpt previously featured in plot (a) of Figure 2. In Figure 5 plot (a) we show all created contours, in plot (b) the remaining contours after filtering with the final melody pitch mean $\overline{P(t)}$ indicated by the thick red line, and in plot (c) the final melody estimation (black) and the ground truth (thick red, shifted down one octave for clarity).
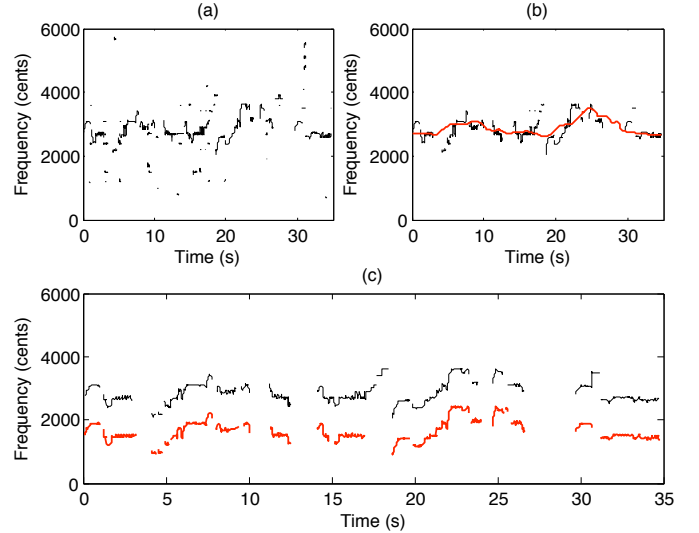

Fig. 5. Vocal jazz excerpt: (a) all pitch contours created, (b) contours after filtering and melody pitch mean (thick red line), (c) final extracted melody (black) and ground truth (thick red, shifted down one octave for clarity).

III. EVALUATION METHODOLOGY

The described system was submitted to the 2010 and 2011 Music Information Retrieval Evaluation eXchange (MIREX), an annual campaign in which different state-of-the-art MIR algorithms are evaluated against the same data-sets in order to compare their performance [31]. This allowed us not only to evaluate our proposed method on an extensive and varied set of testing material, but also to compare it with alternative approaches. The difference between our submission in 2010 and 2011 is the analysis and parameter optimisation described in Section IV-C. By comparing our results before optimisation (2010) and after (2011) we can evaluate the effect of the optimisation on the overall performance of the system.

In addition to the MIREX results, we carried out several complementary evaluation experiments, providing further insight into the nature of the remaining challenges. These include: a qualitative error analysis focusing on octave errors, a study of the effect of the key parameter in our voicing detection method ν on performance, an evaluation of the influence of each algorithmic component of the system on overall performance, and a glass ceiling study in which we examine the current limitations of the system, including the quality of contour formation (peak streaming). The results of these experiments, in particular the glass ceiling study, allow us to identify which parts of the algorithm could be further improved, and provide future directions for our research.

A. Test Collections

1) MIREX: Four music collections were used in the MIREX evaluations (2010 and 2011), as detailed in Table I. Note that each clip from the MIREX09 collection was mixed at three different levels of signal-to-accompaniment ratio resulting in three different test collections, which together with the other collections makes a total of six test collections.

TABLE I
MUSIC COLLECTIONS USED FOR EVALUATION IN MIREX 2010/2011.

Collection	Description
ADC2004	20 excerpts of roughly 20s in the genres of pop, jazz and opera. Includes real recordings, synthesized singing and audio generated from MIDI files. Total play time: 369s.
MIREX05	25 excerpts of a 10-40s 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: 686s.
MIREX08	Four 1 minute long excerpts from north Indian classical vocal performances. There are two mixes per excerpt with differing amounts of accompaniment for a total of 8 audio clips. Total play time: 501s.
MIREX09	374 Karaoke recordings of Chinese songs (i.e. recorded singing with karaoke accompaniment). Each recording is mixed at three different levels of signal-to-accompaniment ratio {-5dB, 0dB, +5dB} for a total of 1,122 audio clips. Total play time: 10,022s.

2) Parameter Optimisation: For optimising system parameters, a separate collection of 14 excerpts of various genres was used. Further details about this collection and the optimisation procedure are provided in [21]. A summary of the results obtained in [21] is provided in section IV-C.

3) Additional experiments: For the voicing detection (IV-E), component evaluation (IV-F) and glass ceiling study (IV-G), we used a representative test set freely available to researchers. This set includes 16 of the ADC2004 excerpts, 13 excerpts similar to those used in the MIREX05 collection, and 40 excerpts similar to those used in the MIREX09 collection.

B. Evaluation Metrics

The algorithms in MIREX were evaluated in terms of five metrics, as detailed in [1]:

- **Voicing Recall Rate**: the proportion of frames labeled voiced in the ground truth that are estimated as voiced by the algorithm.
- **Voicing False Alarm Rate**: the proportion of frames labeled unvoiced in the ground truth that are estimated as voiced by the algorithm.
- **Raw Pitch Accuracy**: the proportion of voiced frames in the ground truth for which the F0 estimated by the algorithm is within $\pm\frac{1}{4}$ tone (50 cents) of the ground truth F0. Algorithms may also report F0 values for frames they estimated as unvoiced so that the raw pitch accuracy is not affected by incorrect voicing detection.
- **Raw Chroma Accuracy**: same as the raw pitch accuracy except that both the estimated and ground truth F0s are mapped into a single octave. This gives a measure of pitch accuracy ignoring octave errors which are common in melody extraction systems.

- **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 frames (out of the entire piece) correctly estimated by the algorithm, where for non-voiced frames this means the algorithm labeled them as non-voiced, and for voiced frames the algorithm both labeled them as voiced and provided a correct F0 estimate for the melody (i.e. within $\pm\frac{1}{4}$ tone of the ground truth).

IV. RESULTS

The results obtained by our optimised algorithm are presented in Table III (Section IV-D). For completeness, we start by presenting the results of MIREX 2010, followed by a qualitative error analysis of our submission. Next, we provide a summary of the optimisation process carried out in [21], and then we present the results obtained by our optimised algorithm in MIREX 2011. Then, we describe the additional evaluation experiments carried out to assess the influence of specific parameters and algorithmic components. Finally, we present the results of a glass ceiling analysis of our algorithm.

A. Comparative Evaluation: MIREX 2010

Five algorithms participated in the 2010 audio melody extraction task of the MIREX campaign. In Table II we present the overall accuracy results obtained by each system for each of the test collections. Systems are denoted by the initials of their authors – HJ [32], TOOS [33], JJY (who submitted two variants) [34] and SG (our submission). For completeness, we also include the results obtained by the best performing system from the previous year's campaign [10], denoted KD. In the last column we provide the mean overall accuracy computed over all six collections[3].

TABLE II
OVERALL ACCURACY RESULTS: MIREX 2010.

Algorithm	2004	2005	2008	2009 0dB	2009 -5dB	2009 +5dB	Mean
HJ	0.61	0.54	0.77	**0.76**	**0.63**	**0.83**	0.69
TOOS	0.54	0.61	0.72	0.72	**0.63**	0.79	0.67
JJY2	**0.72**	0.61	**0.80**	0.63	0.47	0.79	0.67
JJY1	0.70	**0.62**	**0.80**	0.63	0.47	0.79	0.67
SG	0.70	**0.62**	0.78	0.74	0.58	0.81	**0.70**
KD	0.86	0.75	0.81	0.68	0.52	0.78	0.73

We see that of the systems participating in 2010, our system achieved the highest mean overall accuracy, surpassed only by the best performing system from the previous year. Nonetheless, the performance of all systems is very similar (with the exception of KD for the 2004 and 2005 data-sets[4]). We performed an analysis of variance (ANOVA) of the results obtained by the algorithms participating in 2010, revealing that for the 2004, 2005 and 2008 data-sets there

---

[3]The mean is not weighted by the size of the data-sets due to the order of magnitude difference in size between the 2009 data-sets and the other collections which, though smaller, are more representative of the type of material one would encounter in a real world scenario.

[4]A possible explanation for this is KD's better ability at extracting non-vocal melodies, which constitute a larger proportion of these collections.

was in fact no statistically significant difference between any of the algorithms (for a p-value < 0.05). This is probably in part due to the small size of these collections. For the three 2009 collections, a statistically significant difference was found between most algorithms, though the artificial nature of these collections (karaoke accompaniment, amateur singing and no studio mixing or post production) makes them less representative of a real-world scenario. In conclusion, the comparable performance of most systems suggests that further error analysis would be of much value. Only through analysing the types of errors made by each algorithm can we get a better understanding of their advantages and pitfalls.

B. Qualitative Error Analysis

Following the conclusions of the previous section, we performed a qualitative error analysis of our submission, focusing on octave errors. We noted that for the MIREX05 collection there was a significant difference between the raw pitch accuracy and the raw chroma accuracy. This disparity is caused due to the selection of contours at the wrong octave. In Figure 6 we display the raw pitch accuracy versus the raw chroma accuracy obtained by our algorithm in each of the collections (a), and the per-song results for the MIREX05 collection (b).
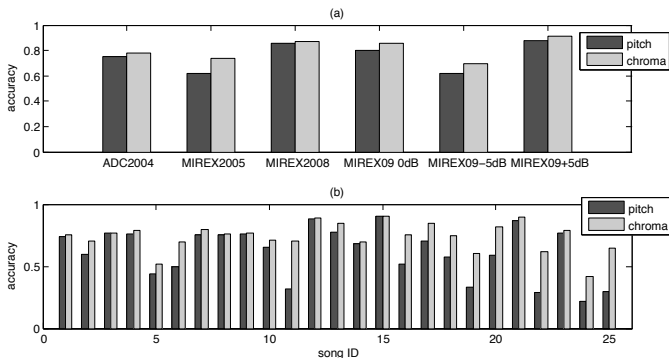

Fig. 6. (a) Mean pitch and chroma accuracies for each test collection. (b) Per-song pitch and chroma accuracies for the MIREX05 collection.

Examining the per-song results we discovered that the largest differences between pitch and chroma accuracy occur mainly in non-vocal excerpts, especially solo piano pieces. This suggests that whilst our octave selection method works well for vocal music, further work would be required to adapt it for instrumental music, especially that performed by a single (polyphonic) instrument.

C. Process Analysis and Parameter Optimisation

In [21] the first two blocks of the system, sinusoid extraction and salience function computation, were studied with the goal of identifying the processing steps and parameter values most suitable for melody extraction. In this section we provide a brief summary of the conclusions reached in that study, which were used to select the processing steps and parameter values for the first two blocks of the system presented in this paper. The effect of the optimisation is shown in the following section were the MIREX 2011 results are presented.

In the first part of the study carried out in [21] alternative signal processing methods were compared for each of the three stages in the sinusoid extraction process (filtering, spectral transform and frequency/amplitude correction). For filtering, it was shown that the equal loudness filter (c.f. II-A1) considerably reduces the energy of non-melody spectral peaks whilst maintaining almost all energy of melody peaks.

Next we evaluated the spectral transform. Some melody extraction systems use a multi-resolution transform instead of the STFT which has a fixed time-frequency resolution [4], [10], [20]. The motivation for using a multi-resolution transform is that it might be beneficial to have greater frequency resolution in the low frequency range where peaks are bunched closer together and are relatively stationary over time, and higher time resolution for the high frequency range where we can expect peaks to modulate rapidly over time (e.g. the harmonics of singing voice with a deep vibrato). In the study we compared the STFT to the multi-resolution FFT (MRFFT) proposed in [25]. Interestingly, it was shown that the MRFFT did not provide any statistically significant improvement to spectral peak frequency accuracy and only a marginal improvement to the final melody F0 accuracy (less than 0.5 cents). Following these observations we opted for using the STFT in the proposed system.

For frequency/amplitude correction two methods were compared: parabolic interpolation [35] and instantaneous frequency using the phase vocoder method [26]. It was shown that both methods provide a significant improvement in frequency accuracy compared to simply using the bin locations of the FFT, and that the phase-based method (used in this paper) performs slightly better (no significant difference though).

In the second part of the study, an evaluation was carried out to study the effect of the weighting parameters α and β, the magnitude threshold γ and the number of harmonics N_h on the resulting salience function. The salience function was computed with different parameter value combinations using a grid search and the resulting salience peaks were evaluated using metrics specifically designed to estimate the predominance of the melody compared to other pitched elements present in the salience function. This led to the determination of optimal values for the salience function parameters: $\alpha = 0.8$, $\beta = 1$, $\gamma = 40$ and $N_h = 20$. For comparison, the values used in MIREX 2010 were 0.8, 2, 40 and 8 respectively, empirically assigned based on initial experiments carried out before the more comprehensive parameter optimisation study in [21].

D. Comparative Evaluation: MIREX 2011

Eight participants took part in the MIREX 2011 melody extraction campaign, including our optimised system (SG)[5]. The overall accuracy results are provided in Table III. For easy comparison, our result from 2010 is repeated in the last row of the table. We see that our optimised system achieves the highest overall accuracy in four of the six test-sets. Consequently, our method also achieves the highest mean overall accuracy (surpassing KD), making it the best performing

TABLE III
OVERALL ACCURACY RESULTS: MIREX 2011.

Algorithm	2004	2005	2008	2009 0dB	2009 -5dB	2009 +5dB	Mean
TY	0.47	0.51	0.70	0.52	0.41	0.56	0.53
TOS	0.59	0.57	0.72	0.74	**0.62**	0.82	0.68
LYRS	0.73	0.59	0.72	0.47	0.36	0.54	0.57
HCCPH	0.44	0.45	0.64	0.50	0.39	0.59	0.50
CWJ	0.73	0.57	0.69	0.53	0.40	0.62	0.59
YSLP	**0.85**	0.65	0.73	0.52	0.39	0.66	0.63
PJY	0.81	0.65	0.71	0.74	0.54	0.83	0.71
SG	0.74	**0.66**	**0.83**	**0.78**	0.61	**0.85**	**0.75**
SG (2010)	0.70	0.62	0.78	0.74	0.58	0.81	0.70

melody extraction algorithm to be evaluated on the current MIREX test-sets (2009 to date). When comparing our results before optimisation (2010) and after (2011), we see that for all collections there is a notable improvement in accuracy. The increase can be attributed to better voicing detection (resulting in lower voicing false alarm rates), better contour generation (higher pitch and chroma accuracies) and less octave errors (smaller difference between pitch and chroma accuracies). We note that whilst the system's parameters have been optimised, it could still be improved through the introduction of new contour characteristics or additional signal processing steps. These options are discussed further in Section IV-G.

E. Voicing

In Section II-E1 we proposed a new voicing detection method in which the determination of voiced sections is based on the study of contour feature distributions. The method was in part responsible for the successful results in MIREX, where our system achieved the best trade-off between voicing recall and voicing false alarm rates. In this section we study the sensitivity of our system to the method's key parameter ν (Equation 9). Recall that ν determines the lenience of the filtering: increasing ν makes it more lenient (less contours are filtered out), whilst decreasing ν makes it stricter (more contours are filtered out). In Figure 7 we plot the overall accuracy, voicing recall and voicing false alarm rates for each collection in our representative test set, as a function of ν.
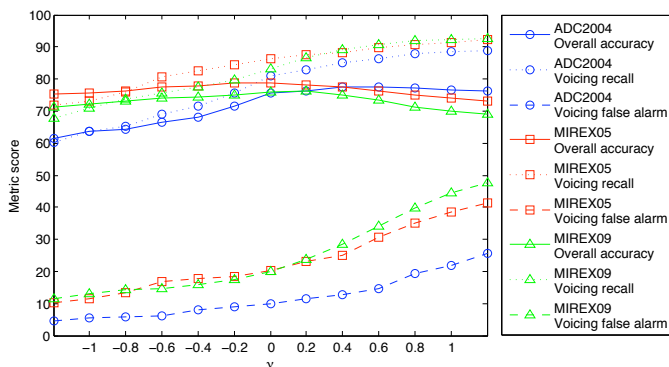

Fig. 7. Overall accuracy, voicing recall and voicing false alarm rates versus the voicing parameter ν.

As expected, the trade-off between the voicing recall and voicing false alarm rates is clearly visible. As ν is increased

(reducing the filtering threshold τ_v) the recall rate goes up for all collections, but so does the false alarm rate. The optimal value for ν is the one which gives the best balance between the two, and can be inferred from the overall accuracy. We see that this optimal value is slightly different for each of the three collections. This is because the relationship between the salience distribution of melody contours and the salience distribution of non-melody contours (c.f. Figure 3 plot (c)) is affected by the type of musical accompaniment used, which varies between the collections. Nonetheless, the optimal ν values for the three collections lie within a sufficiently limited range (0.0-0.4) such that a satisfactory compromise can be made (e.g. for the collections under investigation, $\nu = 0.2$). Finally, this (albeit small) difference between the optimal ν values suggests that whilst the proposed approach already provides good results, further contour characteristics would have to be considered in order to improve voicing detection rates across a wide range of musical styles and genres. As future work we propose the development of a voiced contour classifier trained using a wider set of contour features.

F. Component Evaluation

As with the voicing filter, each algorithmic component of the system influences its overall performance. In Table IV we evaluate the complete system on the representative test set (III-A3) each time removing one component, in this way assessing its effect on overall performance. The components removed are: equal loudness filter (EQ), peak frequency correction (FC), voicing filter (VF), octave duplicate and outlier removal (OO). We also tested replacing the optimised salience function parameters with the MIREX 2010 configuration (SF), as well as removing different combinations of components.

TABLE IV
SYSTEM PERFORMANCE WITH DIFFERENT COMPONENTS REMOVED.

Component Removed	Voicing Recall	Voicing False Alarm	Raw Pitch	Raw Chroma	Overall Accuracy
None	0.86	0.19	0.81	0.83	0.77
EQ	0.83	0.19	0.79	0.81	0.75
FC	0.85	0.19	0.79	0.82	0.76
EQ & FC	0.83	0.18	0.77	0.80	0.75
SF	0.85	0.24	0.77	0.81	0.74
VF	0.92	0.42	0.81	0.83	0.72
OO	0.87	0.24	0.79	0.83	0.75
VF & OO	0.95	0.56	0.79	0.83	0.67
All	0.95	0.64	0.71	0.78	0.60

We see that each component has a direct effect on the overall accuracy. Importantly, we note that there is a strong interaction between components. For example, without the voicing filter (VF) accuracy goes down by 5% and without the octave duplicate and outlier removal (OO) it goes down by 2%, but if both were removed the accuracy would drop by 10%. This reveals that the latter step (OO), in addition to its primary role, also improves voicing detection by removing non-voiced contours that were missed by the voicing filter. If all components were removed the combined effect would cause a drop of 17% in overall accuracy, which is 4% more than the sum of all individual accuracy decreases combined.

G. Glass Ceiling Analysis

As a final evaluation step, we test to see what would be the best result our algorithm could possibly achieve, assuming we had a perfect contour filtering approach. To do this, we compare all contours generated for an excerpt with its ground truth, and keep only those which overlap with the reference melody. These contours are then passed to the final melody selection stage as before, and the resulting melody is evaluated against the ground truth. In Table V we present for each collection the best result obtained by our algorithm, followed by the result obtained using the perfect filtering simulation.

TABLE V
RESULTS ACHIEVED BY SYSTEM AND GLASS CEILING RESULTS.

Collection	Voicing Recall	Voicing False Alarm	Raw Pitch	Raw Chroma	Overall Accuracy
ADC2004	0.83	0.11	0.79	0.81	0.76
	0.84	0.05	0.84	0.84	0.84
MIREX05	0.88	0.23	0.83	0.84	0.78
	0.86	0.07	0.84	0.85	0.86
MIREX09	0.87	0.24	0.81	0.84	0.76
	0.86	0.14	0.85	0.85	0.83

Comparing the results obtained by our system to the results using the perfect filtering simulation, we can make several important observations. First of all, we see that the overall accuracy using the perfect contour filtering simulation is below 100%. As suggested by the title of this section, this reveals a glass ceiling, i.e. a top limit on the overall accuracy that could be obtained by the system in its current configuration. We begin by discussing the differences between our system's results and the glass ceiling results, and then analyse the limitations of the system that result in this glass ceiling limit.

We start by drawing the reader's attention to the raw chroma metric. We see that the chroma accuracy of our system is practically equal to the glass ceiling result. This suggests that the system can almost perfectly select the correct contour when faced with two or more simultaneous contours (that are not octave duplicates). Turning to the raw pitch accuracy, the results obtained by the system are on average only 3.5% below the glass ceiling result. Again, this implies that whilst there is still room for improvement, the octave error minimisation method proposed in the paper is certainly promising. The main difference between our system and the glass ceiling results is the voicing false alarm rate. Though already one of the best voicing detection methods in MIREX, we see that further improvements to the method would provide the most significant increase in the overall accuracy of our system.

Finally, we consider the possible cause of the identified glass ceiling limit. Assuming the system can perform perfect contour filtering, the overall accuracy is determined entirely by the accuracy of the contour formation. If all melody contours were perfectly tracked, the raw pitch and chroma scores of the glass ceiling should reach 100%. This implies that to increase the potential performance of our system, we would have to improve the accuracy of the contour formation. Currently, our tracking procedure takes advantage of temporal, pitch and salience information. We believe that an important part of the puzzle that is still missing is timbre information.

Timbre attributes have been shown to provide important cues for auditory stream segregation [36], suggesting they could similarly be of use for pitch contour tracking. Furthermore, the extraction of pitch specific timbre attributes could lead to the development of a contour timbre feature C_t, that could be used in the melody selection process by introducing rules based on timbre similarity between contours. Another possibility for improving contour formation would be the suppression of noise elements in the signal before the salience function is computed. For instance, we could apply harmonic/percussive source separation such as in [33], [37] to minimise the disruptions in the salience function caused by percussive instruments.

V. CONCLUSION

In this paper we presented a system for automatically extracting the main melody of a polyphonic piece of music from its audio signal. The signal processing steps involved in the extraction of melody pitch candidates were described, as well as the process of grouping them into pitch contours. It was shown that through the characterisation of these pitch contours and the study of their distributions, we can identify characteristics that distinguish melody contours from non-melody contours. It was then explained how these features are used for filtering out non-melody contours, resulting in novel voicing detection and octave error minimisation methods.

The proposed system was evaluated in two MIREX campaigns, where the latest version of our algorithm (2011) was shown to outperform all other participating state-of-the-art melody extraction systems. The results were complemented with a qualitative error analysis, revealing that the different characteristics of instrumental music complicate the task of octave error minimisation, requiring further adjustments to the proposed method for this type of musical content. The MIREX 2011 results confirmed the expected increase in performance following the optimisation of system parameters [21]. We evaluated the influence of individual algorithmic components on system performance, and noted that the interaction between different components can be important for maintaining high accuracies. Finally, a glass ceiling analysis confirmed that in most cases the proposed contour filtering process is successful at filtering out non-melody contours, though a further increase in accuracy could still be achieved by reducing the voicing false alarm rate of the system. In addition, it was determined that to increase the potential performance of the system we would have to improve its contour formation stage, and possible methods for achieving this were proposed.

REFERENCES

[1] G. E. Poliner, D. P. W. Ellis, F. Ehmann, E. Gómez, S. Steich, and B. Ong, "Melody transcription from music audio: Approaches and evaluation," *IEEE Trans. on Audio, Speech and Language Process.*, vol. 15, no. 4, pp. 1247–1256, 2007.

[2] R. Typke, "Music retrieval based on melodic similarity," Ph.D. dissertation, Utrecht University, Netherlands, 2007.

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

[4] M. Goto, "A real-time music-scene-description system: predominant-f0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication*, vol. 43, pp. 311–329, 2004.

[5] R. B. Dannenberg, W. P. Birmingham, B. Pardo, N. Hu, C. Meek, and G. Tzanetakis, "A comparative evaluation of search techniques for query-by-humming usig the MUSART testbed," *J. of the American Soc. for Inform. Science and Technology*, vol. 58, no. 5, pp. 687–701, Feb. 2007.

[6] J.-L. Durrieu, G. Richard, and B. David, "An iterative approach to monaural musical mixture de-soloing," in *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Apr. 2009, pp. 105–108.

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

[8] X. Serra, R. Bresin, and A. Camurri, "Sound and music computing: Challenges and strategies," *J. of New Music Research*, vol. 36, no. 3, pp. 185–190, 2007.

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

[10] K. Dressler, "Audio melody extraction for mirex 2009," in *5th Music Inform. Retrieval Evaluation eXchange (MIREX)*, 2009.

[11] A. Klapuri, "Multiple fundamental frequency estimation by summing harmonic amplitudes," in *Proc. 7th Int. Conf. on Music Inform. Retrieval*, Victoria, Canada, Oct. 2006, pp. 216–221.

[12] 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. on Audio, Speech, and Language Process.*, vol. 15, no. 5, pp. 1564–1578, Jul. 2007.

[13] J.-L. Durrieu, "Automatic transcription and separation of the main melody in polyphonic music signals," Ph.D. dissertation, Télécom ParisTech, 2010.

[14] M. Lagrange, L. G. Martins, J. Murdoch, and G. Tzanetakis, "Normalized cuts for predominant melodic source separation processing," *IEEE Trans. on Audio, Speech and Language Process.*, vol. 16, no. 2, pp. 278–290, Feb. 2008.

[15] J.-L. Durrieu, A. Ozerov, C. Févotte, G. Richard, and B. D. David, "Main instrument separation from stereophonic audio signals using a source/filter model," in *Proc. 17th European Signal Process. Conf. (EUSIPCO)*, Glasgow, Aug. 2009, pp. 15–19.

[16] G. Poliner and D. Ellis, "A classification approach to melody transcription," in *Proc. 6th Int. Conf. on Music Inform. Retrieval*, London, Sep. 2005, pp. 161–166.

[17] A. Cheveigné, "Pitch perception models," in *Pitch*, ser. Springer Handbook of Auditory Research, R. R. Fay, A. N. Popper, C. Plack, R. Fay, A. Oxenham, and A. Popper, Eds. Springer New York, 2005, vol. 24, pp. 169–233.

[18] A. Bregman, *Auditory scene analysis*. Cambridge, Massachussetts: MIT Press, 1990.

[19] D. Huron, "Tone and voice: A derivation of the rules of voice-leading from perceptual principles," *Music Perception*, vol. 19, no. 1, pp. 1–64, 2001.

[20] P. Cancela, "Tracking melody in polyphonic audio," in *4th Music Inform. Retrieval Evaluation eXchange (MIREX)*, 2008.

[21] J. Salamon, E. Gómez, and J. Bonada, "Sinusoid extraction and salience function design for predominant melody estimation," in *Proc. 14th Int. Conf. on Digital Audio Effects (DAFx-11)*, Paris, France, Sep. 2011, pp. 73–80.

[22] (2011, Dec.) Equal loudness filter. [Online]. Available: http://replaygain.org/

[23] D. W. Robinson and R. S. Dadson, "A re-determination of the equal-loudness relations for pure tones," *British J. of Applied Physics*, vol. 7, pp. 166–181, 1956.

[24] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the cuidado project," Institut de Recherche et Coordination Acoustique/Musique (IRCAM), Tech. Rep., 2004.

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

[26] J. L. Flanagan and R. M. Golden, "Phase vocoder," *Bell Systems Technical Journal*, vol. 45, pp. 1493–1509, 1966.

[27] V. Rao and P. Rao, "Vocal melody extraction in the presence of pitched accompaniment in polyphonic music," *IEEE Trans. on Audio Speech and Language Process.*, vol. 18, no. 8, pp. 2145–2154, Nov. 2010.

[28] P. Herrera and J. Bonada, "Vibrato extraction and parameterization in the spectral modeling synthesis framework," in *Proc. Workshop on Digital Audio Effects (DAFx-98)*, 1998, pp. 107–110.

[29] J. Salamon, B. Rocha, and E. Gómez, "Musical genre classification using melody features extracted from polyphonic music signals," in *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Kyoto, Japan, Mar. 2012.

[30] A. Klapuri, "A method for visualizing the pitch content of polyphonic music signals," in *Proc. 10th Int. Soc. for Music Inform. Retrieval Conf.*, Kobe, Japan, 2009, pp. 615–620.

[31] J. S. Downie, "The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research," *Acoustical Science and Technology*, vol. 29, no. 4, pp. 247–255, 2008.

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

[33] H. Tachibana, T. Ono, N. Ono, and S. Sagayama, "Extended abstract for audio melody extraction in mirex 2010," in *6th Music Inform. Retrieval Evaluation eXchange (MIREX)*, Aug. 2010.

[34] S. Joo, S. Jo, and C. D. Yoo, "Melody extraction from polyphonic audio signal mirex2010," in *6th Music Inform. Retrieval Evaluation eXchange (MIREX)*, Aug. 2010.

[35] X. Serra, *Musical Sound Modeling with Sinusoids Plus Noise*. Swets & Zeitlinger, 1997, pp. 91–122.

[36] P. Iverson, "Auditory stream segregation by musical timbre: Effects of static and dynamic acoustic attributes," *J. of Experimental Psychology: Human Perception and Performance*, vol. 21, no. 4, pp. 751–763, Aug. 1995.

[37] 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*, ser. Studies in Computational Intelligence, Z. Ras and A. Wieczorkowska, Eds. Springer Berlin / Heidelberg, 2010, vol. 274, pp. 213–236.

Justin Salamon obtained a B.A. (Hons.) in Computer Science from the University of Cambridge, UK, in 2007. In 2008 he obtained his M.Sc. in Cognitive Systems and Interactive Media from Universitat Pompeu Fabra (UPF), Barcelona, Spain. Currently he is a researcher and PhD 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 Music Information Retrieval (MIR), with a focus on content-based MIR and audio and music processing, including musical stream estimation, melody and bass line extraction and characterisation, query-by-humming/example, classification, music and melodic similarity and indexing.

Emilia Gómez is a post-doctoral researcher and assistant professor (Professor Lector) at the Music Technology Group (MTG), Department of Information and Communication Technologies (DTIC), Universitat Pompeu Fabra (UPF). She graduated as a Telecommunication Engineer specialized in Signal Processing at Universidad de Sevilla. Then, she received a DEA in Acoustics, Signal Processing and Computer Science applied to Music (ATIAM) at IRCAM, Paris. In July 2006, she completed her PhD in Computer Science and Digital Communication at the 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.

正弦提取
sinusoid
extraction
{
等响度滤波器（Equal loudness filter）

频谱变换（spectral transform）—— DFT，得到每个频率强度
　　　　　　　　　　　　　　　　　保留能量峰值的
　　　　　　　　　　　　　　　　　频率

频/幅修正（Frequency/Amplitude correction）

用瞬时频率（IF）修正 DFT 的频率估计
}

得到 Time-Frequency 图

显著化
Salience function
{
谐波求和
harmonic summation
}

谐音列（harmonic series）
能量加权，得到显著性，
每个 block 都重复，得到一个代表性音高
　　　　　　　　　　　　　　↓
　　　　　　　　　　　　　pitch

得到每个时刻的 显著音高（representation of pitch salience）

轮廓创建
contour creation
{
轮廓：T-F 上连续音调值

每个时刻的 显著 pitch ——auditory streaming 听觉流——→ 轮廓
}

得到各条 contour

选择旋律
melody selection
{
计算一组特征
contour
{
pitch mean

pitch deviation

contour mean salience

" total salience

length

vibrato presence
}
}

挑选出主旋律

Melody Extraction by Justin Salamon