A COMPARISON OF MELODY EXTRACTION METHODS BASED ON SOURCE-FILTER MODELLING

Juan J. Bosch[1] **Rachel M. Bittner**[2] **Justin Salamon**[2] **Emilia Gómez**[1]
[1] Music Technology Group, Universitat Pompeu Fabra, Spain
[2] Music and Audio Research Laboratory, New York University, USA
{juan.bosch,emilia.gomez}@upf.edu, {rachel.bittner,justin.salamon}@nyu.edu

ABSTRACT

This work explores the use of source-filter models for pitch salience estimation and their combination with different pitch tracking and voicing estimation methods for automatic melody extraction. Source-filter models are used to create a mid-level representation of pitch that implicitly incorporates timbre information. The spectrogram of a musical audio signal is modelled as the sum of the leading voice (produced by human voice or pitched musical instruments) and accompaniment. The leading voice is then modelled with a Smoothed Instantaneous Mixture Model (SIMM) based on a source-filter model. The main advantage of such a pitch salience function is that it enhances the leading voice even without explicitly separating it from the rest of the signal. We show that this is beneficial for melody extraction, increasing pitch estimation accuracy and reducing octave errors in comparison with simpler pitch salience functions. The adequate combination with voicing detection techniques based on pitch contour characterisation leads to significant improvements over state-of-the-art methods, for both vocal and instrumental music.

1. INTRODUCTION

Melody is regarded as one of the most relevant aspects of music, and melody extraction is an important task in Music Information Retrieval (MIR). Salamon et al. [21] define melody extraction as the estimation of the sequence of fundamental frequency (f_0) values representing the pitch of the lead voice or instrument, and this definition is the one employed by the Music Information Retrieval Evaluation eXchange (MIREX) [7]. While this definition provides an objective and clear task for researches and engineers, it is also very specific to certain types of music data. Recently proposed datasets consider broader definitions of melody, which are not restricted to a single instrument [2, 4, 6].

Composers and performers use several cues to make melodies perceptually salient, including loudness, timbre, frequency variation or note onset rate. Melody extraction methods commonly use cues such as pitch continuity and pitch salience, and some of them group pitches into higher level objects (such as tones or contours), using principles from Auditory Scene Analysis [8, 13, 16, 18, 20]. Some approaches have also considered timbre, either within a source separation framework [10, 17], with a machine learning approach [11], or in a salience based approach [14, 16].

One of the best performing methods so far in MIREX in terms of overall accuracy [20] (evaluated in 2011) is based on the creation and characterisation of pitch contours. This method uses a fairly simple salience function based on harmonic summation [15] and then creates and characterises pitch contours for melody tracking. Voicing detection (determining if a frame contains a melody pitch or not) is one of the strong aspects of this method, even though it might be improved further by incorporating timbre information. In contrast, alternative approaches employ more sophisticated salience functions, but the pitch tracking and voicing estimation components are less complex [10, 12]. Voicing detection has in fact been identified in the literature as a crucial task for improving melody extraction systems [10, 20].

While these approaches work especially well for vocal music, their performance decreases for instrumental pieces, as shown in [6] and [2], where a drop of 19 percentage points in overall accuracy was observed for instrumental pieces compared to vocal pieces. A main challenge for melody extraction methods is thus to cope with more complex and varied music material, with melodies played by different instruments, or with harmonised melodic lines [21]. A key step towards the development of more advanced algorithms and a more realistic evaluation is the availability of large and open annotated datasets. In [4, 6] the authors presented a dataset for melody extraction in orchestral music with such characteristics, and MedleyDB [2] also includes a variety of instrumentation and genres. Results on both datasets generally drop significantly in comparison to results on datasets used in MIREX [7].

Based on results obtained in previous work [5, 6], we hypothesise that a key ingredient for improving salience-based melody extraction in relatively complex music data is the salience function itself. In particular, we propose combining strong components of recently proposed algorithms: (1) a semantically rich salience function based on a

source-filter model, which proved to work especially well in pitch estimation [6, 9, 10], and (2) pitch-contour-based tracking [2, 20], which presents numerous benefits including high-performance voicing detection.

2. RELATED METHODS

This section describes the pitch salience functions and melody tracking methods used as building blocks for the proposed combinations.

2.1 Salience functions

Most melody extraction methods are based on the estimation of pitch salience - we focus here on the ones proposed by Salamon and Gómez [20], and Durrieu et al. [9].

Durrieu et al. [9] propose a salience function within a separation-based approach using a Smoothed Instantaneous Mixture Model (SIMM). They model the spectrum X of the signal as the lead instrument plus accompaniment $\hat{X} = \hat{X}_v + \hat{X}_m$. The lead instrument is modelled as: $\hat{X}_v = X_\Phi \circ X_{f_0}$, where X_{f_0} corresponds to the source, X_Φ to the filter, and the symbol $\circ$ denotes the Hadamard product. Both source and filter are decomposed into basis and gains matrices as $X_{f_0} = W_{f_0} H_{f_0}$ and $X_\Phi = W_\Gamma H_\Gamma H_\Phi$ respectively. H_{f_0} corresponds to the pitch activations of the source, and can also be understood as a representation of pitch salience [9]. The accompaniment is modelled as a standard non negative matrix factorization (NMF): $\hat{X}_m = \hat{W}_m \hat{H}_m$. Parameter estimation is based on Maximum-Likelihood, with a multiplicative gradient method [10], updating parameters in the following order for each iteration: $H_{f_0}, H_\Phi, H_m, W_\Phi$ and W_m. Even though this model was designed for singing voice, it can be successfully used for music instruments, since the filter part is related to the timbre of the sound, and the source part represents a harmonic signal driven by the fundamental frequency.

Salamon and Gómez [20] proposed a salience function based on harmonic summation: a time-domain Equal-Loudness Filter (ELF) is applied to the signal, followed by the Short-Time Fourier Transform (STFT). Next, sinusoidal peaks are detected and their frequency/amplitude values are refined using an estimate of the peaks' instantaneous frequency. The salience function is obtained by mapping each peak's energy to all harmonically related f_0 candidates with exponentially decaying weights.

2.2 Tracking

The estimated pitch salience is then used to perform pitch tracking, commonly relying on the predominance of melody pitches in terms of loudness, and on the melody contour smoothness [1, 10, 12, 20].

Some methods have used pitch contour characteristics for melody tracking [1, 20, 22]. Salamon and Gómez [20] create pitch contours from the salience function by grouping sequences of salience peaks which are continuous in time and pitch. Several parameters need to be set in this process, which determine the amount of extracted contours. Created contours are then characterised by a set of features: pitch (mean and deviation), salience (mean, standard deviation), total salience, length and vibrato related features.

The last step deals with the selection of melody contours. Salamon [20] first proposed a pitch contour selection (PCS) stage using a set of heuristic rules based on the contour features. Salamon [22] and Bittner [1] later proposed a pitch contour classification (PCC) method based on contour features. The former uses a generative model based on multi-variate Gaussians to distinguish melody from non-melody contours, and the latter uses a discriminative classifier (a binary random forest) to perform melody contour selection. The latter also adds Viterbi decoding over the predicted melodic-contour probabilities for the final melody selection. However, these classification-based approaches did not outperform the rule-based approach on MedleyDB. One of the important conclusions of both papers was that the sub-optimal performance of the contour creation stage (which was the same in both approaches) was a significant limiting factor in their performance.

Durrieu et al. [10] similarly use an HMM in which each state corresponds to one of the bins of the salience function, and the probability of each state corresponds to the estimated salience of the source (H_{f_0}).

2.3 Voicing estimation

Melody extraction algorithms have to classify frames as voiced or unvoiced (containing a melody pitch or not, respectively). Most approaches use static or dynamic thresholds [8, 10, 12], while Salamon and Gómez exploit pitch contour salience distributions [20]. Bittner et al. [1] determine voicing by setting a threshold on the contour probabilities produced by the discriminative model. The threshold is selected by maximizing the F-measure of the predicted contour labels over a training set.

Durrieu et al. [10] estimate the energy of the melody signal frame by frame. Frames whose energy falls below the threshold are set as unvoiced and vice versa. The threshold is empirically chosen, such that voiced frames represent more than 99.95% of the leading instrument energy.

3. PROPOSED METHODS

We propose and compare three melody extraction methods which combine different pitch tracking and voicing estimation techniques with pitch salience computation based on source-filter modelling and harmonic summation. These approaches have been implemented in python and are available online [1]. We reuse parts of code from Durrieu's method [2], Bittner et al. [3], and Essentia [4] [3], an open source library for audio analysis, which includes an implementation of [20] which has relatively small deviations

[1] https://github.com/juanjobosch/SourceFilterContoursMelody
[2] https://github.com/wslihgt/separateLeadStereo
[3] https://github.com/rabitt/contour_classification
[4] https://github.com/MTG/essentia

in performance from the authors' original implementation MELODIA [5]. We refer to the original implementation of MELODIA as SAL, and to the implementation in the Essentia library as ESS.

3.1 Pitch Salience Adaptation

There are important differences between the characteristics of salience functions obtained with SIMM (H_{f_0}) and harmonic summation (HS). For instance, H_{f_0} is considerably more sparse, and the range of salience values is much larger than in HS since the NMF-based method does not prevent values (weights) from being very high or very low. This is illustrated in Figure 1: (a) shows the pitch salience function obtained with the source filter model, H_{f_0}. Given the large dynamic range of H_{f_0} we display its energy on a logarithmic scale, whereas plots (b)–(d) use a linear scale. (b) corresponds to HS which is denser and results in complex patterns even for monophonic signals. Some benefits of this salience function with respect to H_{f_0} (SIMM) is that it is smoother, and the range of possible values is much smaller.

Given the characteristics of H_{f_0}, it is necessary to reduce the dynamic range of its salience values in order to use it as input to the pitch contour tracking framework, which is tuned for the characteristics of HS. To do so, we propose the combination of both salience functions $HS(k,i)$ and $H_{f_0}(k,i)$, where k indicates the frequency bin $k = 1 \dots K$ and i the frame index $i = 1 \dots I$. The combination process is illustrated in Figure 1: (1) **Global normalization** (Gn) of HS, dividing all elements by their maximum value $max_{k,i}(HS(k,i))$. (2) **Frame-wise normalization** (Fn) of H_{f_0}. For each frame *i*, divide $H_{f_0}(k,i)$ by $max_k(H_{f_0}(k,i))$. (3) **Convolution in the frequency axis** k of H_{f_0} with a Gaussian filter to smooth estimated activations. The filter has a standard deviation of .2 semitones. (4) **Global normalization** (Gn), whose output is $\widetilde{H_{f_0}}$ (see Figure 1 (c)). (5) **Combination** by means of an element-wise product: $S_c = \widetilde{H_{f_0}} \circ HS$ (see Figure 1 (d)).

3.2 Combinations

We propose three different combination methods. The first (C1) combines the output of two algorithms: estimated pitches from DUR and voicing estimation from SAL. The second (C2) is based on S_c, which combines harmonic summation HS computed with ESS with $\widetilde{H_{f_0}}$, and employs pitch contour creation and selection as the tracking method. The last method (C3) combines S_c with pitch contour creation from [20] and the contour classification strategy from [1]. C2 and C3 correspond to Figure 1, where the contour filtering stage is based on pitch contour selection or pitch contour classification, respectively.

4. EVALUATION

Evaluation was carried out using the MedleyDB and Orchset datasets, following the standard MIREX evaluation

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

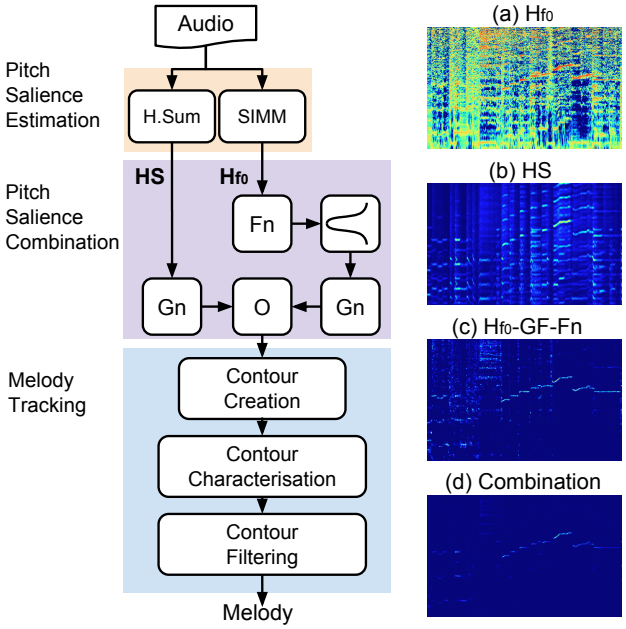

Figure 1. **Left**: Schema for C2 and C3. H.Sum: Harmonic Summation (outputs HS); SIMM: Smoothed Instantaneous Mixture Model (outputs H_{f_0}); Fn: Frame-wise normalisation; Gn: Global normalisation; o: Hadamard product; Gaussian symbol: Gaussian filtering. **Right**: Time-frequency pitch salience representation of an excerpt from "MusicDelta_FunkJazz.wav" (MedleyDB) with (a) SIMM: $log_{10}(H_{f_0})$ is represented for visualisation purposes) (b) Harmonic Summation: HS (c) H_{f_0} normalised per frame, Gaussian filtered and globally normalized ($\widetilde{H_{f_0}}$) (d) Combination (S_c).

methodology. We evaluate the proposed methods (C1–C3) and the original algorithms by Durrieu (DUR), Bittner (BIT) and Salamon. Table 1 provides and overview of their main building blocks. In the case of Salamon's approach, we include the original implementation MELODIA (SAL), and the implementation in the Essentia library (ESS). The latter can be viewed as a baseline for the proposed combination methods (C2, C3), since all three share the same contour creation implementation.

For the evaluation of classification-based methods, we followed [1], and created train/test splits using an "artist-conditional" random partition on MedleyDB. For Orchset we created a "movement-conditional" random partition, meaning all excerpts from the same movement must be used in the same subset: either for training or for testing. Datasets are split randomly into a training, validation and test sets with roughly 63%, 12%, and 25% of the songs/excerpts in the dataset, respectively. This partitioning was chosen so as to have a training set that is as large as possible while retaining enough data in the validation and test sets for results to be meaningful. In order to account for the variance of the results, we repeat each experiment with four different randomized splits.

We set the same frequency limit for all algorithms: $f_{min} = 55$ Hz and $f_{max} = 1760$ Hz. The number of

	(Pre Proc.)+Transform	Salience/Multif_0 Estim.	Tracking	Voicing
DUR [10]	STFT	SIMM	Vit(S)	Energy thd.
SAL [20]	(ELF)+STFT+IF	H.Sum.	PCS	Salience-based
BIT [1]	(ELF)+STFT+IF	H.Sum.	PCC+Vit(C)	Probability-based
C1	(ELF)+STFT+IF	H.Sum + SIMM	PCS+Vit(S)	Salience-based
C2	(ELF)+STFT+IF	H.Sum + SIMM	PCS	Salience-based
C3	(ELF)+STFT+IF	H.Sum + SIMM	PCC+Vit(C)	Probability-based

Table 1. Overview of the methods. STFT: Short Time Fourier Transform, IF: Instantaneous Frequency estimation, ELF: Equal-Loudness Filters, SIMM: Smoothed Instantaneous Mixture Model, using a Source-Filter model, H.Sum: Harmonic Summation, HMM: Hidden Markov Model, Vit(S): Viterbi on salience function, Vit(C): Viterbi on contours, PCS: Pitch Contour Selection, PCC: Pitch Contour Classification.

bins per semitone was set to 10, and the hop size was 256 samples (5.8 ms), except for SAL which is fixed to 128 samples (2.9 ms) given a sampling rate of 44100 Hz.

4.1 Datasets

The evaluation is conducted on two different datasets: MedleyDB and Orchset, converted to mono using (left+right)/2. MedleyDB contains 108 melody annotated files (most between 3 and 5 minutes long), with a variety of instrumentation and genres. We consider two different definitions of melody, **MEL1**: the f_0 curve of the predominant melodic line drawn from a single source (MIREX definition), and **MEL2**: the f_0 curve of the predominant melodic line drawn from multiple sources. We did not use the third type of melody annotation included in the dataset, since it requires algorithms to estimate more than one melody line (i.e. multiple concurrent lines). Orchset contains 64 excerpts from symphonies, ballet suites and other musical forms interpreted by symphonic orchestras. The definition of melody in this dataset is not restricted to a single instrument, with all (four) annotators agreeing on the melody notes [4, 6]. The focus is pitch estimation, while voicing detection is less important: the proportion of voiced and unvoiced frames is 93.7/6.3%.

Following MIREX methodology [6], the output of each algorithm is compared against a ground truth sequence of melody pitches. Five standard melody extraction metrics are computed using `mir_eval` [19]: Voicing Recall Rate (VR), Voicing False Alarm Rate (VFA), Raw Pitch Accuracy (RPA), Raw Chroma Accuracy (RCA) and Overall Accuracy (OA). See [21] for a definition of each metric.

4.2 Contour creation results

Before evaluating complete melody extraction systems, we examine the initial step, by computing the recall of the pitch contour extraction stage as performed in [1]. We measure the amount of the reference melody that is covered by the extracted contours, by selecting the best possible f_0 curve from them. For the MEL1 definition in MedleyDB the oracle output yielded an average RPA of .66 (σ = .22) for HS and .64 (σ = .20) for S_c. In the case of MEL2: .64 (σ = .20) for HS and .62 (σ = .18) for S_c. For Orchset we obtain .45 (σ = .21) for HS and .58 (σ = .18) for S_c. These results represent the highest raw pitch accuracy that could be obtained by any of the melody extraction methods using contours created from HS and S_c. Note however that these values are dependent on the parametrization of the contour creation stage, as described in [20].

[6] http://www.music-ir.org/mirex/wiki/2014:Audio_Melody_Extraction

4.3 Melody extraction results

Results for all evaluated algorithms and proposed combinations are presented in Table 2 for MedleyDB (MEL1 and MEL2) and in Table 3 for Orchset. The first remark is that the three proposed combination methods yield a statistically significantly (t-test, significance level α = .01) higher overall accuracy (OA) than the baseline (ESS) for both datasets and both melody definitions. The OA of C2 and C3 is also significantly higher than the OA of all other evaluated approaches on MedleyDB (MEL1), with the exception of SAL* (SAL with a voicing threshold optimized for MedleyDB/MEL1): C2-SAL* (p = .10), C3-SAL* (p = .27). For the MEL2 definition C2 and C3 yield an OA that is significantly higher than all compared approaches. In the case of Orchset, C3 is significantly better than C1 and C2 except when increasing the voicing threshold on C2* (p = .78), and outperforms all compared approaches but DUR. As expected, pitch related metrics (RPA, RCA) are the same for C1 and DUR (they output the same pitches), and voicing detection metrics (VR, VFA) are the same for C1 and SAL. This simple combination is already able to significantly improve overall accuracy results on MedleyDB in comparison to all evaluated state-of-the-art approaches except SAL, thanks to the highest pitch estimation accuracy obtained by DUR, and the lowest VFA yielded by SAL. However, OA results are not as high as with DUR on Orchset, due to the lower recall of SAL. An important remark is that DUR always obtains almost perfect recall, since this method outputs almost all frames as voiced. This has a huge influence on the overall accuracy on Orchset, since this dataset contains 93.7% of voiced frames. However, the false alarm rate is also very high, which lowers OA results on MedleyDB, since it contains full songs with large unvoiced portions.

SAL and BIT perform similarly on MedleyDB, but the usefulness of Bittner's method becomes evident on Orch-

		MedleyDB-MEL1					MedleyDB-MEL2				
Method	ν	VR	VFA	RPA	RCA	OA	VR	VFA	RPA	RCA	OA
DUR	-	1.0 (.01)	.96 (.05)	.66 (.21)	.73 (.16)	.36 (.16)	1.0 (.01)	.95 (.06)	.65 (.18)	.73 (.14)	.42 (.14)
SAL	.2	.78 (.13)	.38 (.14)	.54 (.27)	.68 (.19)	.54 (.17)	.76 (.12)	.33 (.12)	.52 (.24)	.66 (.17)	.53 (.17)
SAL*	-1	.57 (.21)	.20 (.12)	.52 (.26)	.68 (.19)	.57 (.18)	.54 (.19)	.17 (.09)	.49 (.23)	.66 (.17)	.53 (.18)
BIT	-	.80 (.13)	.48 (.13)	.51 (.23)	.61 (.19)	.50(.15)	.79 (.10)	.44 (.13)	.50 (.20)	.60 (.16)	.50 (.14)
ESS	.2	.79 (.13)	.44 (.15)	.55 (.26)	.68 (.19)	.50 (.17)	.77 (.12)	.39 (.14)	.53 (.23)	.66 (.17)	.50 (.17)
C1	.2	.78 (.13)	.38 (.14)	.66 (.21)	.73 (.16)	.56 (.14)	.76 (.12)	.33 (.12)	.65 (.18)	.73 (.14)	.57 (.13)
C2	.2	.65 (.15)	.26 (.11)	.63 (.21)	.70 (.16)	.61 (.15)	.62 (.14)	.21 (.08)	.61 (.19)	.69 (.14)	.60 (.15)
C3	-	.75 (.15)	.38 (.16)	.58 (.23)	.64 (.19)	.59 (.16)	.74 (.13)	.34 (.13)	.58 (.19)	.64 (.17)	.60 (.14)

Table 2. Mean results (and standard deviation) over all excerpts for the five considered metrics, on MedleyDB with MEL1 and MEL2 definition. Parameter ν refers to the voicing threshold used in the methods based on pitch-contour selection [20]. In the case of classification-based methods (BIT and C3), this parameter is learnt from data. SAL* refers to the results obtained with the best ν for MedleyDB/MEL1.

set: with the same candidate contours, the RPA increases with respect to SAL. This classification-based method is thus partially able to learn the characteristics of melody contours in orchestral music. Orchset is characterized by a higher melodic pitch range compared to most melody extraction datasets which often focus on sung melodies [4].

5. DISCUSSION

5.1 Salience function and contour creation

By comparing the results obtained by SAL and C2 we can assess the influence of the salience function on methods based on pitch contour selection [20]. SAL obtains lower pitch related accuracies (RPA, RCA) than C2, especially for orchestral music. The difference between RPA and RCA is also greater in SAL than compared to C2, indicating SAL makes a larger amount of octave errors, especially for Orchset. This indicates that the signal representation yielded by the proposed pitch salience function S_c is effective at reducing octave errors, in concurrence with the observations made in [9]. C3 also provides a significantly higher accuracy in comparison to BIT, showing that the proposed salience function helps to improve melody extraction results also when combined with a pitch contour classification based method. Once again, this is particularly evident in orchestral music.

Note that even if the performance ceiling when creating the pitch contours from HS on MedleyDB is 2 percentage points higher than with S_c (see section 4.2), melody extraction results are better with S_c. This is due to the fact that the precision of the contour creation process with the proposed salience function is higher than with HS.

5.2 Pitch tracking method

By comparing the results of C2 and C3 we can assess the influence of the pitch tracking strategy, as both methods use the same contours as input. In MedleyDB, there is no significant difference between both methods in terms of overall accuracy, but the contour classification based method (C3) has a higher voicing recall for both melody definitions, while the contour selection method (C2) has a better RPA, RCA and VFA. This agrees with the findings from Bittner et al. [1] who also compared between both pitch tracking strategies using HS as the salience function. In the case of Orchset, the difference in OA is evident between C2-C3 (p = .004), since the classification based approach tends to classify most frames as voiced, which is beneficial when evaluating on this dataset. However, increasing the tolerance in C2 (C2*, $\nu = 1.4$) provides similar OA results: C2*-C3 (p = .78).

An analysis of feature importance for pitch contour classification (using S_c) revealed that salience features are the most discriminative in both datasets, especially mean salience. This suggests that the proposed salience function S_c is successful at assigning melody contours a higher salience compared to non-melody contours.

The most important difference between C2 and C3 is that C3 allows the model to be trained to fit the characteristics of a dataset, avoiding the parameter tuning necessary in rule-based approaches like [20]. The set of rules from [20] used in C2 are not tuned to orchestral music, which also explains why C2 obtains a lower OA on Orchset with the default parameters. Careful tuning could considerably improve the results.

5.3 Influence of parameters

We ran some additional experiments with C2 in order to investigate the influence of the parameters used to compute the pitch salience function and contour creation step. Several parameters affect the creation of the salience function [9], here we focus on the number of iterations used for the source-filter decomposition and how it affects the results obtained with the proposed salience function S_c. We found that on Orchset the drop in OA when reducing the number of iterations from 50 to 10 is less than 4%. On MedleyDB the change in OA is less than 1% when varying from 50 to 10 iterations. We also found that DUR is generally more sensitive to the decrease in number of iterations, which is a positive aspect of our proposed approach, given the high computational cost of the pitch salience estimation algorithm. For instance, DUR experiments a relative decrease in OA of around 7% when going from 50 to 10 iterations (on MedleyDB with MEL1 definition). The relative decrease in the case of C2 is less than 3%. The results reported in this study are based on 30 iterations.

Method	ν	VR	VFA	RPA	RCA	OA
DUR	-	1.0 (.00)	.99 (.09)	.68 (.20)	.80 (.12)	.63 (.20)
SAL	.2	.60 (.09)	.40 (.23)	.28 (.25)	.57 (.21)	.23 (.19)
SAL*	1.4	.81 (.07)	.57 (.25)	.30 (.26)	.57 (.21)	.29 (.23)
BIT	-	.69 (.14)	.45 (.25)	.35 (.17)	.53 (.15)	.37 (.16)
ESS	.2	.59 (.10)	.38 (.22)	.29 (.24)	.55 (.20)	.22 (.19)
C1	.2	.60 (.09)	.40 (.22)	.68 (.20)	.80 (.12)	.42 (.14)
C2	.2	.49 (.11)	.28 (.16)	.57 (.20)	.69 (.14)	.39 (.16)
C2*	1.4	.70 (.11)	.44 (.21)	.57 (.20)	.70 (.14)	.52 (.19)
C3	-	.73 (.12)	.46 (.23)	.53 (.19)	.65 (.14)	.53 (.18)

Table 3. Mean results (and standard deviation) over all excerpts for the five considered metrics, on Orchset. Parameter ν refers to the voicing threshold used in the methods based on pitch-contour selection. In the case of the classification-based methods (BIT and C3), this parameter is learnt from data. The sign * refers to the results obtained with the best v.

We also analysed the influence of Gaussian filtering (see Figure 1), by suppressing it from the salience function creation process. The effect is quite small on MedleyDB, but is more noticeable on Orchset where it results in a 4% point drop in OA. A possible explanation is that in symphonic music many instruments contribute to the melody but are not perfectly in tune. By smoothing the salience function we are able to increase the pitch salience of notes played by orchestral sections in unison. Pitch contour extraction and voicing detection parameters are more relevant, however. Overall accuracy generally increases on MedleyDB when the maximum allowed gap between pitches in a contour is decreased from 100 ms to 50 ms (50 ms is used in the reported experiments). Since SIMM can add noise to unvoiced frames, using the stricter threshold of 50 ms in the contour creation step can help filter some of this noise by preventing it from being tracked as part of a contour.

We also conducted a study of the effect of the voicing parameter (ν) on both C2 and SAL. A higher value results in less contours being filtered as unvoiced, which is beneficial on Orchset. A lower value (heavier filtering) is beneficial when evaluating against the MEL1 definition, since the melody is restricted to a single instrument. Varying ν from -1.4 to 1.0, the OA results with SAL range from .46 to .57 on MedleyDB MEL1, while with C2 they only range from .56 to .61. In the case of MEL2, the OA of SAL ranges from .46 to .54, while in the case of C2 the range is also smaller, from .57 to .60. This shows that the proposed method is more robust to the selection of the voicing parameter. While default contour creation parameters in ESS already provided satisfying results for C2 on MedleyDB, further tests on Orchset showed that they could be tuned to go up to 0.60 overall accuracy. In fact, just modifying the voicing parameter to $\nu = 1.4$ already increases the OA of C2 to 0.52. The highest overall accuracy obtained by SAL with the best parameter configuration on Orchset is 0.29 (see Table 3). This again shows that the same pitch contour selection based method can be improved with the proposed salience function, especially on orchestral music.

5.4 Pitch salience integration in contour creation

The benefits of combining a source-filter model and a pitch contour based tracking method have become evident by now, and each of the proposed combination approaches has its advantages and disadvantages. The main advantage of C1 is its simplicity, and that it always yields the same RPA as DUR, which is always the best in all datasets. The main disadvantage is that the contour creation process from SAL does not take advantage of the benefits of the pitch salience from DUR. This is the reason why it becomes important to integrate the source-filter model into the pitch contour creation process, as performed in C2 and C3. One difficulty of the integration is that the salience function from DUR needs to be adapted to the pitch contour creation framework. However, this improves overall accuracy in both MedleyDB and Orchset.

6. CONCLUSIONS AND FUTURE WORK

This paper presents a comparison of melody extraction methods based on source-filter models within a pitch contour based melody extraction framework. We propose three different combination methods, based on a melody oriented pitch salience function which adapts a source-filter model to the characteristics of the tracking algorithm. The adaptation is based on the combination with a salience function based on harmonic summation. We have shown that the proposed salience function helps improve pitch estimation accuracy and reduce octave errors in comparison to harmonic summation. This salience function consistently improves the mean overall accuracy results when it substitutes harmonic summation in pitch contour based tracking methods. This is true for both heuristic and machine-learning-based approaches, when evaluated on a large and varied set of data. Future work deals with improving the proposed salience function, in order to further reduce the amount of noise in unvoiced parts.

7. ACKNOWLEDGEMENTS

This work is partially supported by the European Union under the PHENICX project (FP7-ICT-601166) and the Spanish Ministry of Economy and Competitiveness under CASAS project (TIN2015-70816-R) and Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).

8. REFERENCES

[1] R. Bittner, J. Salamon, S. Essid, and J. Bello. Melody extraction by contour classification. In *Proc. ISMIR*, pages 500–506, Málaga, Spain, Oct. 2015.

[2] R. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. Bello. Medleydb: a multitrack dataset for annotation-intensive mir research. In *Proc. ISMIR*, pages 155–160, Taipei, Taiwan, Oct. 2014.

[3] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X Serra. Essentia: an open source library for audio analysis. *ACM SIGMM Records*, 6, 2014.

[4] J. Bosch and E. Gómez. Melody extraction in symphonic classical music: a comparative study of mutual agreement between humans and algorithms. In *Proc. 9th Conference on Interdisciplinary Musicology – CIM14*, Berlin, Germany, Dec. 2014.

[5] J. Bosch and E. Gómez. Melody extraction by means of a source-filter model and pitch contour characterization (MIREX 2015). In *11th Music Information Retrieval Evaluation eXchange (MIREX), extended abstract*, Málaga, Spain, Oct. 2015.

[6] J. Bosch, R Marxer, and E. Gómez. Evaluation and combination of pitch estimation methods for melody extraction in symphonic classical music. *Journal of New Music Research*, DOI: 10.1080/09298215.2016.1182191, 2016.

[7] J Stephen Downie. The music information retrieval evaluation exchange (2005-2007): A window into music information retrieval research. *Acoustical Science and Technology*, 29(4):247–255, 2008.

[8] K. Dressler. Towards Computational Auditory Scene Analysis: Melody Extraction from Polyphonic Music. In *Proc. CMMR*, pages 319–334, London, UK, 2012.

[9] J. Durrieu, B. David, and G. Richard. A musically motivated mid-level representation for pitch estimation and musical audio source separation. *Sel. Top. Signal Process. IEEE J.*, 5(6):1180–1191, 2011.

[10] J. Durrieu, G. Richard, B. David, and C. Févotte. Source/filter model for unsupervised main melody extraction from polyphonic audio signals. *Audio, Speech, Lang. Process. IEEE Trans.*, 18(3):564–575, 2010.

[11] D. Ellis and G. Poliner. Classification-based melody transcription. *Machine Learning*, 65(2-3):439–456, 2006.

[12] B. Fuentes, A. Liutkus, R. Badeau, and G. Richard. Probabilistic model for main melody extraction using constant-Q transform. In *Proc. IEEE ICASSP*, pages 5357–5360, Kyoto, Japan, Mar. 2012. IEEE.

[13] 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*, 43(4):311–329, September 2004.

[14] C. Hsu and J. Jang. Singing pitch extraction by voice vibrato/tremolo estimation and instrument partial deletion. In *Proc. ISMIR*, pages 525–530, Utrecht, Netherlands, Aug. 2010.

[15] A. Klapuri. Multiple fundamental frequency estimation by summing harmonic amplitudes. In *Proc. ISMIR*, pages 216–221, Victoria, Canada, Oct. 2006.

[16] M. Marolt. Audio melody extraction based on timbral similarity of melodic fragments. In *EUROCON 2005*, volume 2, pages 1288–1291. IEEE, 2005.

[17] 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. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(5):1564–1578, 2007.

[18] R. Paiva, T. Mendes, and A. Cardoso. Melody detection in polyphonic musical signals: Exploiting perceptual rules, note salience, and melodic smoothness. *Computer Music Journal*, 30(4):80–98, 2006.

[19] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis. mir_eval: A transparent implementation of common MIR metrics. In *Proc. ISMIR*, pages 367–372, Taipei, Taiwan, Oct. 2014.

[20] J. Salamon and E. Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Trans. Audio. Speech. Lang. Processing*, 20(6):1759–1770, 2012.

[21] J. Salamon, E. Gómez, D. Ellis, and G. Richard. Melody Extraction from Polyphonic Music Signals: Approaches, applications, and challenges. *IEEE Signal Process. Mag.*, 31:118–134, 2014.

[22] J. Salamon, G. Peeters, and A. Röbel. Statistical characterisation of melodic pitch contours and its application for melody extraction. In *13th Int. Soc. for Music Info. Retrieval Conf.*, pages 187–192, Porto, Portugal, Oct. 2012.