MAXIMUM FILTER VIBRATO SUPPRESSION FOR ONSET DETECTION

Sebastian Böck and Gerhard Widmer

Department of Computational Perception
Johannes Kepler University
Linz, Austria
sebastian.boeck@jku.at

ABSTRACT

We present *SuperFlux* - a new onset detection algorithm with vibrato suppression. It is an enhanced version of the universal spectral flux onset detection algorithm, and reduces the number of false positive detections considerably by tracking spectral trajectories with a maximum filter. Especially for music with heavy use of vibrato (e.g., sung operas or string performances), the number of false positive detections can be reduced by up to 60% without missing any additional events. Algorithm performance was evaluated and compared to state-of-the-art methods on the basis of three different datasets comprising mixed audio material (25,927 onsets), violin recordings (7,677 onsets) and operatic solo voice recordings (1,448 onsets). Due to its causal nature, the algorithm is applicable in both offline and online real-time scenarios.

1. INTRODUCTION AND RELATED WORK

Onset detection is the process of finding the starting points of all musically relevant events in an audio performance. While the detection of percussive onsets can be considered a solved problem,[1] softer onsets, vibrato and tremolo still constitute major challenges for existing algorithms.

Since soft onsets (e.g., of woodwind or bowed string instruments) have a long attack phase with a slow rise in energy, energy- and magnitude-based approaches are not the best choice for detecting them. To overcome the shortcomings of these approaches, specific algorithms that solve the soft onset problem by additionally incorporating phase [2, 3, 4] or pitch information [5, 6, 7] or a combination thereof [8] have been proposed. However, magnitude-based methods [9] have advanced and perform on par with the above methods and outperform them on all kinds of percussive audio material.

The current state-of-the-art methods for online [10] and offline [11] onset detection are based on a probabilistic model and incorporate a recurrent neural network with the spectral magnitude and its first time derivative as input features. In particular the offline variant *OnsetDetector* shows superior performance on all sorts of signals [1]. Because of its bidirectional architecture, it is able to model the context of an onset in order to both detect hard do discover onsets in complex mixes (e.g., a soft note onset of relatively low volume) and suppress events which are erroneously considered onsets by other algorithms, such as the sound of stopped strings.

Vibrato is an artistic effect commonly used in classical music and can be sung or played by instruments. It reflects a quasi-periodic change in the frequency of a played or sung note. Vibrato is characterized technically by the amount of pitch variation (e.g., $\pm$ a semitone for string instruments and up to a complete tone in operas) and the frequency with which the pitch changes over time (e.g., 6 Hz). It is sometimes used synonymously as a combination with another effect: the tremolo, which describes changes in the volume of a note. As it is technically difficult for a human musician to play pure vibrato or tremolo, both effects are usually performed simultaneously. Because of the resulting fluctuations in loudness and frequency, it is very hard for onset detection algorithms to correctly distinguish between new note onsets and an intended variation of the note.

So far only very few publications have addressed the problem of spuriously detected onsets in vibrato music. Collins [5] uses a vibrato suppression stage in his pitch-based onset detection method that first identifies vibrato regions which fluctuate by at most one semitone around the center frequency and collects the extrema in a list. The region is then expanded gradually in time to cover the whole duration of the vibrato. After having identified the complete extent of the vibrato, all values within this window are replaced by the mean of the extrema list. The onset detection function is based on the concept of stable pitches and uses the changes in pitches as cues for new onsets.

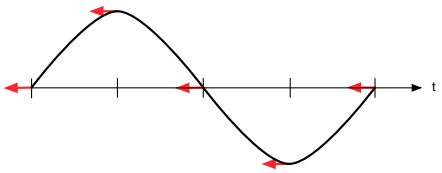
Schleusing et al. [7] deploy a system based on the inverse correlation of N consecutive spectral frames centered around the current location. Regions of stable pitch lead to low inverse correlation values, and pitch changes result in peaks in the detection function. To suppress vibrato, they use a warp compensation which cancels out small pitch changes within the window under consideration, leaving the changes due to onsets mostly untouched.

Both systems work only in offline mode because they require future information to reliably detect the vibrato and apply their counter-measures. Furthermore, they can be used only for pitched non-percussive music and are unsuitable for all other kinds of audio material. Glover et al. [12] described a linear prediction post-processing technique that can be applied to existing online onset detection algorithms and is not limited to pitched instruments. Although not designed specifically for vibratos, it is related because it improves mostly the recall performance of the investigated onset detection algorithms.

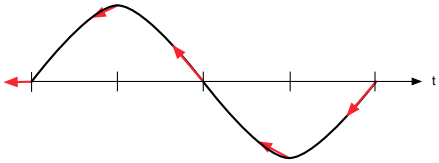
In this paper, we concentrate on vibrato suppression methods which can be applied both to online (i.e., real-time processing of a continuous audio stream with minimal latency) and offline onset detection scenarios. As a basis for our research we chose the *LogFiltSpecFlux* method proposed in [9], which is the current non-probabilistic state-of-the-art onset detection method [1]. It operates in the spectral domain; more specifically, it only considers the magnitude spectrogram without incorporating any phase informa-

[1] State-of-the art detection algorithms achieve F-measure values greater than 0.95 on percussive sounds [1].

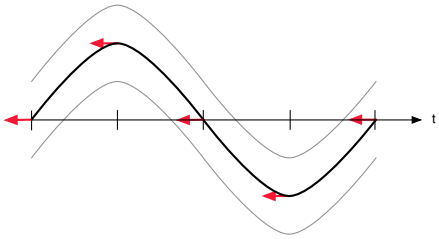
tion. Like the common spectral flux algorithm [13] it relies on the detection of positive changes in the energy over time, but instead of calculating the difference from the same bin of a previous frame (see Figure 1a), it includes a special trajectory-tracking stage. A general approach to trajectory tracking is shown in Figure 1b and illustrates the ability of this method to suppress spurious positive energy fragments (which are falsely detected as new onsets by the spectral flux algorithm) because it calculates the difference along the trajectory path. The new method incorporates a maximum filter (Figure 1c) to track the trajectory in a computationally efficient and simple but effective way.



(a) *classical bin-wise difference calculation*



(b) *trajectory tracking-based difference calculation*



(c) *maximum filter-based difference calculation*

Figure 1: *(a) the problem of difference calculation in vibrato signals inherent in spectral flux-based methods, (b) a general trajectory tracking-based solution, and (c) the proposed maximum filter-based method. Arrows indicate the positions used for difference calculation, with the tails indicating the positions of the minuends and the heads those of the subtrahends. The grey lines in (c) mark the frequency bounds of the regions which are assigned the same magnitude value via the maximum filter.*

2. PROPOSED METHOD

Our method adds a spectral trajectory-tracking stage to the common spectral flux (SF) [13] algorithm. The system processes the signal in a frame-wise manner. Thus the signal is divided into overlapping chunks of length $N = 2048$ samples, and each frame is weighted with a Hann window of the same length before being transformed to the spectral domain via the Discrete Fourier Transform (DFT).

The original spectral flux implementation uses the temporal evolution of the magnitude spectrogram $|X(n,k)|$ by calculating the bin-wise difference between two consecutive short-time spectra and then sums all positive deviations [13]:

$$SF(n) = \sum_{k=1}^{k=\frac{N}{2}} H\left(|X(n,k)| - |X(n-1,k)|\right) \qquad (1)$$

with $H(x) = \frac{x+|x|}{2}$ being the half-wave rectifier function, n the frame number and k the frequency bin index.

The problem of the difference calculation in signals containing vibrato can be seen in Figure 2b. Many spectral peaks appear if the difference between a frequency bin and the same frequency bin k of the previous frame $n-1$ is calculated. The result of our maximum filter-based trajectory-tracking approach is illustrated in Figure 1c and described below.

2.1. Pre-processing

To facilitate trajectory tracking, some pre-processing measures are taken. In general, it is desirable to have a much finer temporal resolution than the standard frame-rate of $f_r = 100$ fps used for onset detection. We thus chose to double the frame-rate so that we can report onsets with 5 ms accuracy. An increased frame rate has the advantage that the quantized magnitude spectrogram features much smoother trajectories, which simplifies tracking. However (in addition to the higher computational cost) is has the disadvantage that the individual differences (on which the onset detection function is based) are much smaller due to greater overlapping of the windows. Thus, instead of calculating the difference between consecutive frames, we use frames that are further apart, the offset determined by the parameter μ:

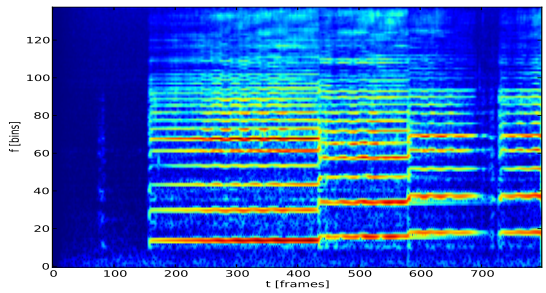
$$\mu = \max\left(1, \lfloor (N/2 - \min\{n | w(n) > r\})/h + 0.5 \rfloor\right) \qquad (2)$$

with r being a parameter which defines the height ratio of the window function $w(n)$ with length N, and the hop-size h between two frames. The hop-size can be calculated by dividing the sample-rate of the audio signal f_s by the frame-rate f_r. The spectral flux onset detection function with the improved difference calculation is given by:

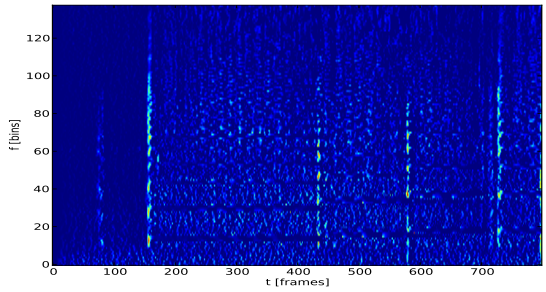
$$SF'(n) = \sum_{k=1}^{k=\frac{N}{2}} H\left(|X(n,k)| - |X(n-\mu,k)|\right) \qquad (3)$$

with $\mu \geq 1$. The main advantage of this measure is that the difference values are greater since the overlap of the two windows considered is smaller (because they are located further apart). Values of $r = 0.5$, resulting in $\mu = 2$, were found to yield the best performance for a frame-rate $f_r = 200$ fps and the standard sample-rate $f_s = 44.1$ kHz of the audio signal. Additionally, the peaks of the resulting onset detection function are much closer to the actual onset positions, which renders lag compensation in the final peak-picking unnecessary.

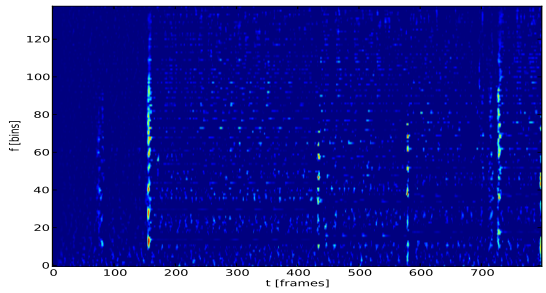
To simplify trajectory tracking, the linear magnitude spectrogram is filtered with a filterbank $F(k,m)$ with $M = 138$ triangular filters with center frequencies aligned on the western music scale and separated by a quarter-tone from each other, covering a frequency range of $27.5 \ldots 16{,}000$ Hz. Operating on a logarithmic frequency scale has the advantage that a constant frequency



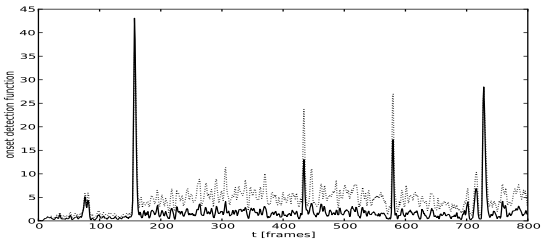
(a) *magnitude spectrogram*



(b) *classical bin-wise positive difference*



(c) *positive difference with maximum-filtering trajectory tracking*



(d) *sum of differences*

Figure 2: *(a) logarithmic magnitude spectrogram of a 4 s violin recording featuring vibrato, and filtered with a quarter-tone filterbank, (b) the positive differences calculated by taking the bin-wise difference between two consecutive spectrogram frames, and (c) with the proposed maximum-filtering trajectory tracking. (d) shows the sum of all positive differences, the dotted line representing the sum of the spectrogram given in (b) [9] and the solid line the sum with the maximum filter applied shown in (c).*

shift (e.g., by a semi-tone) always results in a shift by the same number of frequency bins (2 if quarter-tone filters are used) independent of the fundamental frequency of a sounding note. Thus, the search range for trajectory tracking is constant, independently of the starting frequency bin m.

It has been found to be advantageous (i) to filter the spectrogram first and then take the logarithm of the summed (filtered) magnitude as in [9] (using the same trick of adding 1 before taking the logarithm) and (ii) not to normalize the filters of the filterbank to have equal areas. The logarithmic filtered spectrogram is given by:

$$X_{log,filt}(n,m) = \log_{10}\left(|X(n,k)| \cdot F(k,m) + 1\right) \qquad (4)$$

with m being the frequency bin index on the quarter-tone frequency scale used.

2.2. Trajectory tracking

The frequency deviation of vibrato in string music is usually ± 1 semitone with an alternation frequency of up to 10 Hz. In operatic singing, the frequency deviation can be much greater, but the alternation frequency is lower, which results in a very similar search space for the tracking of the magnitude trajectories. For the given setting of $f_r = 200$ fps and a quarter-tone filtered spectrogram, a search space of $m = \pm 1$ frequency bins over $\mu = 2$ consecutive time frames covers the expected fluctuations.

Below we present two methods we investigated. They are superseded by our maximum filter-based approach, described subsequently, which performs as well or better but has a much lower computational complexity.

First we investigated an approach which uses the cross-correlation of two frames to determine the shift in frequency needed to achieve the highest similarity between the two frames. Based on this frequency shift, we calculated the bin-wise differences from the μ-th preceding frame shifted by exactly this lag. The method is similar to that used in [7]. There, the correlation between two consecutive frames of the linearly scaled spectrogram is used to formulate a detection function, but a special warping method is needed to compensate for the greater frequency spreading at higher frequencies. Using a logarithmic frequency scale as described in the previous section and incorporated in the approach in [14] renders warping for proper cross-correlation calculation between two frames unnecessary. Because both methods use the level of correlation directly as a feature (in [7] weighted by the energy of the signal), they only use frequencies up to 8000 Hz [7] and 3000 Hz [14], respectively, to achieve higher correlation values. Since we use the cross-correlation values only to choose the shift needed for maximum correlation, the frequency range need not be limited.

While this method works perfectly for monophonic pitched non-percussive music, it shows inferior performance when used for mixed audio signals where high energy portions of the signal can impede the vibrato detection based on correlation. Thus, we implemented a more universal approach which works on all kinds of musical signals. A simple trajectory tracking approach was chosen which follows the magnitude trajectory of each frequency bin m backwards in time in μ individual time-steps. The difference for each bin is then calculated with respect to the magnitude along the trajectory path, as can be seen in Figure 1b. Since this approach is computationally expensive, methods with lower complexity were investigated.

A very simple method which performs as well as (or better than) the two aforementioned approaches – independently of the audio material used – incorporates a maximum filter. Maximum filters are commonly used in computer vision and set the value at a given position to the maximum value in its neighborhood, which is defined by the shape of the filter. We chose the filter shape such that the current frequency bin and its direct neighbors on the logarithmically scaled filtered spectrogram $X_{log,filt}(n,m)$ are covered, but limited to the current time frame. The effect of this maximum filter is a widened trajectory (on the frequency axis), and is shown in Figure 1c. The maximum filtered spectrogram is then given by:

$$X^{max}_{log,filt}(n,m) = \max\left(X_{log,filt}(n,m-1:m+1)\right) \quad (5)$$

In the final *SuperFlux* detection function, the difference is then calculated with respect to this maximum-filtered spectrogram:

$$SF^*(n) = \sum_{m=1}^{m=M} H\left(X_{log,filt}(n,m) - X^{max}_{log,filt}(n-\mu,m)\right) \quad (6)$$

The effect of the measures described above is clearly visible in Figure 2c, which plots the positive difference (calculated to the next to last frame) with maximum-filtering trajectory tracking of the 4-second recording of a violin played with vibrato shown in Figure 2a. Compared to the standard spectral flux difference calculation approach (Figure 2b), it clearly shows fewer positive energy components in the regions played with vibrato. Figure 2d plots the sums of the two difference calculation approaches shown above. The solid line represents the *SuperFlux* detection function according to Equation 6, the dotted line the standard spectral flux algorithm (applied to the filtered logarithmic spectrogram given in Equation 4). This approach is described in [9] and serves as a state-of-the-art spectral flux implementation for evaluation in Section 3. Although the peaks of the *SuperFlux* detection function are sometimes a bit lower if the new notes are played slurred (e.g., onsets around frame numbers 430 and 580), the overall detection function has a much lower noise floor caused by vibrato. The remaining ripple is mostly due to variations in loudness, for instance effects intended by the player (e.g., tremolo) or a natural loudness fluctuation while playing vibrato.

2.3. Peak picking

We use the peak-picking method described in [9] to select the final onsets of the *SuperFlux* detection function. This method is simple and suitable for both offline and online settings. In online mode (i.e., when reading an incoming audio stream) no future information is available, and thus only past information can be used. A frame n of the *SuperFlux* onset detection function $SF^*(n)$ is selected as an onset if it fulfills the following three conditions:

1. $SF^*(n) = \max\left(SF^*(n - pre\_max : n + post\_max)\right)$,
2. $SF^*(n) \geq \text{m}ean(SF^*(n - pre\_avg : n + post\_avg)) + \delta$,
3. $n - n_{previous\,onset} > combination\_width$,

where δ is the tunable threshold. The other parameters were chosen to yield the best performance on the complete dataset. Specifically, $pre\_max = 30$ ms, $post\_max = 30$ ms, $pre\_avg = 100$ ms, $post\_avg = 70$ ms, and $combination\_width = 30$ ms achieved good overall results. Parameter values must first be converted into frames depending on the frame-rate f_r used. For peak picking in online mode, $post\_max$ and $post\_avg$ are set to 0.

3. EVALUATION

We used a variety of datasets and settings in our evaluation to maximize comparability with published methods.

3.1. Datasets

The biggest dataset used for evaluation is that described in [9], which consists mostly of mixed audio material covering different types of musical genres, performed on various instruments. It includes the sets used in [3], [8], and [11]. The 321 files have a total length of approximately 102 minutes and have 27,774 annotated onsets (25,927 if all onsets within 30 ms are combined). The main purpose of this set is to show how the new *SuperFlux* algorithm performs on a general-purpose dataset. This dataset is hereafter referred to as *Böck*. Based on this set, we built a subset covering only the violin and cello recordings played with vibrato. These 16 files have 849 onsets.

For comparison with the current state-of-the-art algorithm for pitched non-percussive music presented in [7], we use the authors' dataset. However, not all sound files and annotations could be used for evaluation, since the authors could provide only part of this set. As the available dataset contains 75% of the original dataset (7,677 instead of 9,717 onsets) and an identical distribution of the different playing styles (50% contain vibrato, some staccato etc.), we are confident that the results obtained are nonetheless comparable. We call this the *Wang* dataset.

To investigate our algorithm's ability to suppress the vibrato in operatic singing, a third dataset (called the *Opera* dataset) consisting of solo singing rehearsal recordings of a Haydn opera was used. The recordings were made at the *Ars Electronica Future Lab* in Linz, Austria. The set covers both male and female singers and has a total length of 10 minutes, containing 1,448 onsets.

3.2. Performance measures and evaluation settings

The performance of onset detection methods is commonly evaluated by means of Precision, Recall, and F-measure. If a detected onset is within the evaluation window around an annotated ground truth onset location, it is considered to be correct. However, every detected onset can only match once, and thus any detected onset within the evaluation window of two different annotated onsets counts as one true positive and one false negative (a missed onset). The same applies to annotations: all additionally reported onsets within the evaluation window of an annotation are counted as false positive detections. For better comparability with other results, we match the evaluation parameters as follows:

Our standard setting is that used in [9], which combines all annotated onsets within 30 ms to a single onset and uses an evaluation window of $\pm$ 25 ms to identify correctly detected onsets. Thus, the $combination\_width$ parameter of our peak-picking is also set to 30 ms.

The second set of parameters (used for the evaluation of the *Wang* dataset) uses the same settings as in [7], where all onsets within 50 ms are combined (i.e., $combination\_width = 50$ ms) and an evaluation window of $\pm$ 70 ms is used.

Unless otherwise noted, all results were obtained by swiping the threshold parameter δ of the peak-picking stage and choosing the value that maximizes the F-measure on the respective dataset.

3.3. Results & Discussion

In order to demonstrate that the *SuperFlux* algorithm is a good all-round performer which not only suppresses false positive onsets in music with vibrato, but also performs on the same level as state-of-the-art methods, we tested it against various other onset detection algorithms.

3.3.1. Competitors

For comparison, we chose the four best-performing online and offline onset detection methods among those submitted to the 2012 MIREX evaluation [1]. We consider these submissions the state of the art, since they achieved the highest ever F-measures in the MIREX evaluation. *OnsetDetector.2012* is an improved version of the method originally proposed in [11], which shows superior performance in offline scenarios. Together with its online variant, *OnsetDetectorLL* [10], it belongs to the group of probabilistic approaches. Since both were trained on the complete *Böck* dataset (cf. Section 3.1), results given for these algorithms were obtained with 8-fold cross-validation and parameter tuning on the training subset. The *LogFiltSpecFlux* [9] algorithm uses no probabilistic information and thus is much less computationally demanding. It can be used both in online and offline scenarios and marks the upper bound of performance "simple" algorithms are able to achieve to date.

3.3.2. Böck set

The results for the full *Böck* dataset are given in Table 1. In online mode, the new *SuperFlux* algorithm clearly outperforms the *LogFiltSpecFlux* method [9] on which it is based, and it closes the gap to the reference *OnsetDetectorLL* neural network-based approach [10].

	Precision	Recall	F-measure
online			
OnsetDetectorLL [10]	0.863	0.783	0.821
LogFiltSpecFlux [9]	0.854	0.753	0.801
SuperFlux	0.855	0.787	0.820
offline			
OnsetDetector.2012 [11]	0.892	0.855	0.873
LogFiltSpecFlux [9]	0.877	0.756	0.812
SuperFlux	0.883	0.793	0.836

Table 1: *Precision, Recall and F-measure of different onset detection algorithms using online (upper half) and offline (lower half) settings on the Böck dataset. Results for the OnsetDetectorLL [10] and OnsetDetector.2012 [11] algorithms were obtained with 8-fold cross-validation and parameters selected solely on the training set.*

An important aspect of the results is the shift of the new method towards higher recall values (and thus a more balanced ratio with respect to precision). Although the algorithm does not detect more onsets per se, suppressing spurious onsets has the very favorable side effect of allowing a lower overall threshold to be chosen for the peak-picking stage, which leads, in turn, to a higher recall rate without too many additional false positives.

In offline mode, the overall picture is very similar: all methods performed slightly better than in online mode with the exception of the *OnsetDetector.2012* algorithm, which exhibited superior performance. This is mainly due to the algorithm's ability to model the context of an onset and thus to detect "more difficult" onsets that cannot be found by other methods. Detailed investigations of the remaining false positive detections revealed that *OnsetDetector.2012* recognizes the sound of a stopped string and thus does not report an onset in such situations, which results in a higher precision rate. However, this is only possible if future information is available (i.e., only in offline mode) and exploited by the algorithm – which is not the case for the *SuperFlux* since its trajectory tracking is strictly causal, and the offline mode only differs in the peak-picking settings.

Table 2 compares *SuperFlux* and *LogFiltSpecFlux* on the basis of the string pieces of the dataset, and highlights the ability of our *SuperFlux* algorithm to successfully suppress false positive detections originating mostly from vibrato. Especially in online mode, the number of false detections decreases from 185 to 118, which is a reduction by 36%. At the same time *SuperFlux* misses fewer notes (263 compared to 294) because of the lower threshold chosen. In offline mode, the number of false positive detections cannot be reduced any further, but a few additional correctly identified onsets lead to slightly improved results compared to the online mode.

	Precision	Recall	F-measure
online			
OnsetDetectorLL [10]	0.822	0.676	0.742
LogFiltSpecFlux [9]	0.750	0.654	0.699
SuperFlux	0.832	0.690	0.755
offline			
OnsetDetector.2012 [11]	0.834	0.820	0.827
LogFiltSpecFlux [9]	0.786	0.684	0.732
SuperFlux	0.836	0.701	0.762

Table 2: *Precision, Recall and F-measure of different onset detection algorithms using online (upper half) and offline (lower half) settings on the strings subset of the Böck dataset using the same parameters as used for the results in Table 1.*

For the results in Table 2 the parameters were not optimized to give the best F-measure performance on the strings subset; rather, the settings used to obtain the results in Table 1 were retained to demonstrate our algorithm's ability to outperform existing approaches on both a general-purpose dataset and string recordings with vibrato without altering settings.

3.3.3. Wang set

Table 3 shows the performance on violin music on the basis of the *Wang* dataset. The *SuperFlux* method outperforms all other algorithms in terms of false positive detections both in online and offline mode. In comparison to the *LogFiltSpecFlux* method, a reduction in false positives by 61% in online mode and 58% in offline mode can be achieved.

Compared to the algorithm described in [7], which is tuned specifically for pitched non-percussive signals with vibrato, *Su-*

perFlux is able to achieve the same low level of false positive detections, but increases the number of correctly reported onsets by 3.8%. Since the method in [7] works only in offline mode, no results for online mode can be given. Because the results of the *SuperFlux* algorithm performing in online mode are on the same level as this highly specialized algorithm for pitched non-percussive instruments (in offline mode), it can be considered a more universal approach for onset detection.

	True positives	False positives
online		
OnsetDetectorLL [10] *	92.3%	20.8%
LogFiltSpecFlux [9] *	97.1%	20.7%
SuperFlux *	92.7%	9.6%
offline		
Collins [5] *	62.4%	24.4%
OnsetDetector.2012 [11] *	96.5%	15.5%
LogFiltSpecFlux [9] *	97.0%	17.8%
Schleusing et.al. [7] *	91.2%	9.2%
SuperFlux *	94.7%	9.1%

Table 3: *True and false positive rates of different onset detection algorithms using online (upper half) and offline (lower half) settings on the Wang dataset. Results for Collins' and Schleusing's algorithms were taken from [7]. Asterisks mark the evaluation method used in [7].*

Interestingly, the methods without any dedicated vibrato suppression (*LogFiltSpecFlux* and *OnsetDetector*) outperform the one proposed in [5], which does include a vibrato suppression stage and is also tuned specifically towards pitched instruments.

Since the recordings in the *Wang* dataset are exclusively solo recordings made in a sound-absorbing room and contain only very few polyphonic parts, we consider the results given in Table 2 a much better approximation to real-world examples since they also feature accompanying instruments, which make vibrato tracking and suppression harder. Also, the evaluation criteria chosen are very lax compared to those used for all other results. With the stricter evaluation, the new *SuperFlux* algorithm achieves true and false positive rates of 89.7% and 22.8% respectively (Precision = 0.772, Recall = 0.897, and F-measure = 0.830).

3.3.4. Opera set

The last dataset for performance evaluation was the newly created dataset with male and female opera rehearsal recordings. In line with the other results, our method dramatically outperforms the *LogFiltSpecFlux* algorithm and thus closes the gap to probabilistic methods. In the case of online peak picking, the number of false detections decreased from 1198 to 498, which is a reduction by 58%. In offline mode, the false positive rate was reduced by 55%. The recalls of both algorithms are almost identical in both cases.

Note that no opera material was used to train the two neural network-based methods. Only the threshold values for peak picking were adopted to yield the best overall performance. This explains the imbalance of the recall and precision values compared to those of our new method, which exhibits a much better balance.

	Precision	Recall	F-measure
online			
OnsetDetectorLL [10]	0.588	0.744	0.657
LogFiltSpecFlux [9]	0.435	0.638	0.518
SuperFlux	0.649	0.637	0.643
offline			
OnsetDetector.2012 [11]	0.576	0.777	0.662
LogFiltSpecFlux [9]	0.480	0.632	0.546
SuperFlux	0.672	0.635	0.653

Table 4: *Precision, Recall and F-measure of different onset detection algorithms using online (upper half) and offline (lower half) settings on the Opera dataset.*

3.4. Runtime

The new *SuperFlux* algorithm has almost the same low computational complexity as the *LogFiltSpecFlux* method [9] on which it is based. On a single 2.26 GHz core of an Intel Core 2 Duo MacBook Pro, processing of a 60-second audio piece takes 2 seconds (30 times real-time) compared to 1.7 seconds of the same algorithm without any maximum filtering trajectory tracking. This is extremely fast compared to neural network-based approaches, which take approximately 14 and 20 seconds (online- and offline-mode). Additionally, they require annotated audio material for training, which takes several hours.

4. CONCLUSIONS

This paper has presented a new method for vibrato suppression with maximum filtering. Our *SuperFlux* onset detection algorithm is based on the common spectral flux method and is able to reduce the number of false positive detections originating from vibrato by up to 60% compared to current state-of-the-art implementations. It does so without missing any onsets otherwise detected.

In comparison to highly specialized vibrato suppression mechanisms for monophonic pitched music, our method achieves the same precision rate but improves the recall rate by 4%. The same rise in recall rate can be observed on complex polyphonic mixed audio signals. This underlines the universal suitability of the new algorithm.

Since our method's vibrato suppression mechanism is based solely on past information, it can be used in online real-time applications without any fundamental modifications. In online scenarios, the method closes the performance gap to the best neural network-based approach but has the advantage of a much lower computational complexity. Because of this low processing demands it can be considered the first choice for a universal onset detection method suitable for all kinds of music. An open-source (BSD-licensed) reference Python implementation of the method can be found at https://github.com/CPJKU/SuperFlux.

5. ACKNOWLEDGMENTS

This work is supported by the European Union Seventh Framework Programme FP7 / 2007-2013 through the PHENICX project (grant agreement no. 601166). We would like to thank Ye Wang and Olaf Schleusing for providing access to the dataset used in [7].

6. REFERENCES

[1] "MIREX 2012 onset detection results," `http://nema.lis.illinois.edu/nema_out/mirex2012/results/aod/`, 2012, accessed 2013-03-27.

[2] J.P. Bello, C. Duxbury, M. Davies, and M. Sandler, "On the use of phase and energy for musical onset detection in the complex domain," *IEEE Signal Processing Letters*, vol. 11, no. 6, pp. 553–556, June 2004.

[3] J.P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler, "A tutorial on onset detection in music signals," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 1035–1047, September 2005.

[4] S. Dixon, "Onset detection revisited," in *Proceedings of the 9th International Conference on Digital Audio Effects (DAFx-06)*, Montreal, Quebec, Canada, September 2006, pp. 133–137.

[5] N. Collins, "Using a pitch detector for onset detection," in *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, September 2005.

[6] R. Zhou, M. Mattavelli, and G. Zoia, "Music onset detection based on resonator time frequency image," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1685–1695, November 2008.

[7] O. Schleusing, B. Zhang, and Y. Wang, "Onset detection in pitched non-percussive music using warping-compensated correlation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*, April 2008, pp. 117–120.

[8] A. Holzapfel, Y. Stylianou, A.C. Gedik, and B. Bozkurt, "Three dimensions of pitched instrument onset detection," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1517–1527, 2010.

[9] S. Böck, F. Krebs, and M. Schedl, "Evaluating the online capabilities of onset detection methods," in *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR 2012)*, Porto, Portugal, October 2012, pp. 49–54.

[10] S. Böck, A. Arzt, F. Krebs, and M. Schedl, "Online real-time onset detection with recurrent neural networks," in *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx-12)*, York, UK, September 2012.

[11] F. Eyben, S. Böck, B. Schuller, and A. Graves, "Universal onset detection with bidirectional long short-term memory neural networks," in *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, August 2010, pp. 589–594.

[12] J. Glover, V. Lazzarini, and J. Timoney, "Real-time detection of musical onsets with linear prediction and sinusoidal modeling," *EURASIP Journal on Advances in Signal Processing*, vol. 68, 2011.

[13] P. Masri, *Computer Modeling of Sound for Transformation and Synthesis of Musical Signals*, Ph.D. thesis, University of Bristol, UK, 1996.

[14] R. Sonnleitner, B. Niedermayer, G. Widmer, and J. Schlüter, "A Simple and Effective Spectral Feature for Speech Detection in Mixed Audio Signals," in *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx-12)*, York, UK, September 2012.

[15] N. Collins, "A comparison of sound onset detection algorithms with emphasis on psychoacoustically motivated detection functions," in *Proceedings of the AES Convention 118*, 2005, pp. 28–31.