EVALUATING THE ONLINE CAPABILITIES OF ONSET DETECTION METHODS

Sebastian Böck, Florian Krebs and Markus Schedl
Department of Computational Perception
Johannes Kepler University, Linz, Austria

ABSTRACT

In this paper, we evaluate various onset detection algorithms in terms of their online capabilities. Most methods use some kind of normalization over time, which renders them unusable for online tasks. We modified existing methods to enable online application and evaluated their performance on a large dataset consisting of 27,774 annotated onsets. We focus particularly on the incorporated preprocessing and peak detection methods. We show that, with the right choice of parameters, the maximum achievable performance is in the same range as that of offline algorithms, and that preprocessing can improve the results considerably. Furthermore, we propose a new onset detection method based on the common *spectral flux* and a new peak-picking method which outperforms traditional methods both online and offline and works with audio signals of various volume levels.

1. INTRODUCTION AND RELATED WORK

Onset detection, the task of finding musically meaningful events in audio signals, is fundamental to many applications: Real-time applications such as automatic score followers [7] can be enhanced by incorporating (online) onset detectors that look for note onsets in a live performance, while (offline) onset detection is used increasingly to improve digital audio workstations with a view to event-wise audio processing.

Many different methods of solving this task have been proposed and evaluated over the years. Comprehensive overviews of onset detection methods were presented by Bello et al. in [2] and Collins in [6] (with special emphasis on psychoacoustically motivated methods in the latter). Dixon proposed enhancements to several of these in [9]. All methods were evaluated in an *offline* setting, using a normalization over the whole length of the signal or applying averaging techniques which require future information.

For *online* onset detection, only few evaluations have been carried out: Brossier et al. [5] compared four onset functions based on spectral features and proposed a method for dynamic thresholding in online scenarios, using a dataset of 1,066 onsets for evaluation. Stowell and Plumbley [18] proposed *adaptive whitening* as an improvement to short-time Fourier transform (STFT) based onset detection methods and evaluated eight detection functions using a dataset of 9,333 onsets. Glover at al. [12] applied linear prediction and sinusoidal modeling to online onset detection, but used a relatively small dataset of approximately 500 onsets for evaluation.

These traditional onset detection methods usually incorporate only spectral and/or phase information of the signal, are easy to implement, and have modest computational cost. In contrast, methods based on machine learning techniques (e.g., neural networks in [11,15]) or on probabilistic information (e.g., Hidden Markov models in [8]) depend on large datasets for training and are in general computationally more demanding, which makes them unsuited for online processing.

The onset detection process is usually divided into three parts (as shown in Figure 1): signal preprocessing, computation of the actual onset detection function (ODF), and peak detection.

Figure 1. Basic onset detection workflow.

There are generally two normalization steps that require special attention in an online context: The first can be found in the preprocessing step where many implementations normalize the audio input prior to further processing.

The second and more widespread use of normalization is in the peak detection stage, where the whole ODF is normalized before being processed further. An exception to this rule are some machine learning approaches like the neural network-based methods, since their detection function can be considered as a probability function which already has the range $[0..1]$. Furthermore, most offline methods use smoothing or averaging over (future) time to compute dynamic thresholds for the final peak-picking.

This paper is structured as follows: We combine the ODFs described in Section 2.2 with different preprocessing methods from Section 2.1 and evaluate them on the dataset described in Section 3.1 using the peak-picking method given in Section 2.3.4. In Section 4 we discuss the results,

and we give conclusions in Section 5.

2. COMPARED METHODS

Previously, onset detection algorithms used to work directly with the time signal $x(t)$. However, all current onset detection algorithms use a frequency representation of the signal. We used frames of 23 ms length (2048 samples at a sample rate of 44.1 kHz) that are filtered with a Hann window before transfer into the frequency domain by means of STFT. The hopsize between two consecutive frames was set to 10 ms, which results in a frame rate of 100 frames per second. The resulting spectrogram $X(n,k)$ (n denoting the frame and k the frequency bin number) was then processed further by the individual preprocessing and onset detection algorithms.

2.1 Preprocessing

2.1.1 Filtering

Scheirer [17] stated that, in onset detection, it is advantageous if the system divides the frequency range into fewer sub-bands as done by the human auditory system. Filtering has been applied by many authors (e.g. [6,14,17]), and neural network based approaches also use filter banks to reduce the dimensionality of the STFT spectrogram [11].

2.1.2 Logarithmic magnitude

Using the logarithmic magnitude instead of the linear representation was found to yield better results in many cases, independently of the ODF used [11,14]. λ is a compression parameter and was adjusted for each method separately. Adding a constant value of 1 results in only positive values:

$$X^{log}(n,k) = log(\lambda \cdot X(n,k) + 1) \tag{1}$$

2.1.3 Adaptive whitening

Proposed in [18], adaptive whitening normalizes the magnitudes $|X(n,k)|$ of each frequency bin separately by past peak values. The iterative algorithm (with r being a floor parameter and m the memory coefficient) is given as follows:

$$P_{n,k} = \begin{cases} max(|X(n,k)|, r, m \cdot P_{n-1,k}) & \text{if } n \geq 1 \\ max(|X(n,k)|, r) & \text{otherwise} \end{cases}$$

$$|X(n,k)| \leftarrow \frac{|X(n,k)|}{P_{n,k}} \tag{2}$$

2.2 Onset detection functions

We have chose to omit other common methods such as phase deviation (PD) [3], high frequency content (HFC) [16] or rectified complex domain (RCD) [9], since they exhibited inferior performance in our tests.

2.2.1 Spectral Flux

The spectral flux (SF) [16] describes the temporal evolution of the magnitude spectrogram by computing the difference between two consecutive short-time spectra. This difference is determined separately for each frequency bin, and all positive differences are then summed to yield the detection function.

$$SF(n) = \sum_{k=1}^{k=\frac{N}{2}} H(|X(n,k)| - |X(n-1,k)|) \tag{3}$$

with $H(x) = \frac{x+|x|}{2}$ being the half-wave rectifier function. Variants of this method use the L_2-norm instead of the L_1-norm or the logarithmic magnitude [14] (cf. Section 2.1.2).

2.2.2 Weighted Phase Deviation

Another class of detection function utilizes the phase of the signal [3, 9]. The change in the instantaneous frequency (the second order derivative of the phase $\varphi(n,k)$) is an indicator of a possible onset. In [9], an improvement to the phase deviation ODF called weighted phase deviation (WPD) was proposed. The WPD function weights each frequency bin of the phase deviation function with its magnitude.

$$WPD(n) = \frac{2}{N} \sum_{k=1}^{k=\frac{N}{2}} |X(n,k) \cdot \varphi''(n,k)| \tag{4}$$

2.2.3 Complex Domain

Another way to incorporate both magnitude and phase information (as in the WPD detection function) was proposed in [10]. First, the expected target amplitude and phase $X_T(n,k)$ for the current frame are estimated based on the values of the two previous frames assuming constant amplitude and rate of phase change. The complex domain (CD) ODF is then defined as:

$$CD(n) = \sum_{k=1}^{k=\frac{N}{2}} |X(n,k) - X_T(n,k)| \tag{5}$$

2.3 Peak detection

Illustrated in Figure 2 and common to all onset detection methods is the final thresholding and peak-picking step to detect the onsets in the ODF. Various methods have been proposed in the literature; we give an overview of the different components and modifications needed to make them suitable for online processing.

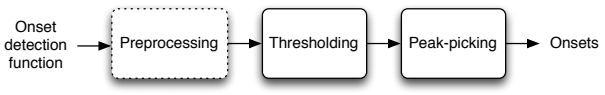

Figure 2. Peak detection process.

2.3.1 Preprocessing

The preprocessing stage of the peak detection process consists mainly of two components: smoothing of the peaky ODF and normalization. Both of them cannot be used in an online scenario. Instead, moving average techniques as outlined in Section 2.3.2 are applied to normalize the ODF locally. To prevent detecting many false positives due to a peaky ODF, the effect of smoothing can be approximated by introducing a minimal distance from the last onset w_5 as proposed in Section 2.3.4.

2.3.2 Thresholding

Before picking the final onsets from the ODF, thresholding is performed to discard the non-onset peaks. Most methods use dynamic thresholding to take into account the loudness variations of a music piece. Mean [9], median [3, 11, 18] or combinations [5, 12] are commonly used to filter the ODF. If only information about the present or past is used, the thresholding function is suitable for online processing.

2.3.3 Peak-picking

Two peak-picking methods are commonly used for final detection of onsets. One selects all local maxima in the thresholded detection function as the final onset positions. Since detecting a local maximum requires both past and future information, this method is only applicable to offline processing.

The other method selects all values above the previously calculated threshold as onsets and is also suitable for online processing. The downside of this approach is its relatively high false positive rate because the threshold parameter must be set to a very low level to detect the onsets reliably.

2.3.4 Proposed peak detection

We use a modified version of the peak picking method proposed in [9] to also satisfy the constraints for online onset peak detection. A frame n is selected as an onset if the corresponding $ODF(n)$ fulfills the following three conditions:

1. $ODF(n) = max(ODF(n - w_1 : n + w_2)$
2. $ODF(n) \geq mean(ODF(n - w_3 : n + w_4)) + \delta$
3. $n - n_{last\,onset} > w_5$

where δ is a fixed threshold and $w_1..w_5$ are tunable peak-picking parameters. For online detection, we set $w_2 = w_4 = 0$. Our online experiments experiments showed that, on average, onsets are detected one frame earlier than annotated in the dataset (using the values specified in Section 3.3). As we want to find the perceptual onset times (as annotated), we report the onset one frame later than detected. Note that this does not mean that we predict the onset, it only means that the onset can be recognized in the signal before it is perceived.

Unlike in previous studies [5, 12, 18] we do not use the same thresholding parameters for all ODFs. This is mainly because some of the ODFs have fewer peaks and hence need less averaging in the thresholding stage than others.

2.4 Neural network based methods

For reference, we compare the presented methods with two state-of-the-art algorithms, the *OnsetDetector* [11] and its online variant *OnsetDetector.LL* [4]:

OnsetDetector uses a bidirectional neural network which processes the signal both in a forward and backward manner, making it an offline algorithm. The algorithms showed exceptional performance compared to other algorithms independently of the type of onsets in the audio material, especially in its latest version tested during the MIREX contest in 2011 [1].

OnsetDetector.LL incorporates a unidirectional neural network to model the sequence of onsets based solely on causal audio signal information.

Since these methods show very sharp peaks (representing the propability of an onset) at the actual onset positions, the before mentioned peak detection method is not applied, and a simple thresholding is used instead.

2.5 New method

We propose a new onset detection method which is based on the spectral flux (cf. Section 2.2.1), drawing on various other author's ideas.

As a first step, we filter the linear magnitude spectrogram $|X(n,k)|$ with a filter bank. We investigated different types of filter banks (Mel, Bark, Constant-Q) and found that they all outperform the standard spectral flux. Since they all perform approximately equally well when using a similar number of filter bands, we chose a pseudo Constant-Q, where the frequencies are aligned according to the frequencies of the semitones of the western music scale over the frequency range from 27.5 Hz to 16 kHz, but using a fixed window length for the STFT. Overlapping triangular filters sum all STFT bins belonging to one filter bin (similarly to Mel filtering). The resulting filter bank $F(k,b)$ has $B = 82$ frequency bins with b denoting the bin number of the filter and k the bin number of the linear spectrogram. The filters have not been normalized, resulting in an emphasis of the higher frequencies, similar to the HFC method. The resulting filtered spectrogram $X_{filt}(n,b)$ is given by:

$$X_{filt}(n,b) = |X(n,k)| \cdot F(k,b) \tag{6}$$

Applying Equation 1 to the filtered linear magnitude spectrogram $X_{filt}(n,b)$ yields the logarithmic filtered spectrogram $X^{log}_{filt}(n,b)$. The final ODF O is then given by:

$$O(n) = \sum_{k=1}^{k=\frac{N}{2}} H\left(\left|X^{log}_{filt}(n,b)\right| - \left|X^{log}_{filt}(n-1,b)\right|\right) \tag{7}$$

where H is the half-wave rectifier function defined in Section 2.2.1.

3. EXPERIMENTS

To evaluate the methods described, we conducted three experiments: First, the methods were evaluated under online conditions: no future information was used to decide

whether there is an onset at the current time point. Second, the same methods were evaluated under offline conditions (enabling prior data normalization or computing averages that incorporate future information) to determine the maximum performance achievable by each method. Third, we attenuated the volume of the audio data to an increasing degree to test the online methods' abilities to cope with signals of different volume without access to normalization.

3.1 Dataset

To evaluate the presented onset detection and peak-picking methods we use a dataset of real world recordings.

An onset is usually defined as the exact time a note or instrument starts sounding after being played. However, this timing is hard to determine, and thus it is impossible to annotate the real onset timing in complex audio recordings with multiple instruments, voices, and effects. Thus, the most commonly used method for onset annotation is marking the earliest time point at which a sound is audible by humans. This instant cannot be defined in pure terms (e.g., minimum increase of volume or sound pressure), but is a rather complex mixture of various factors.

The annotation process is very time-consuming because it is performed in multiple passes. First, onsets are annotated manually during slowed down playback. In the second pass, visualization support is used to refine the onset positions. Spectrograms obtained with different STFT lengths are used in combination to capture the precise timing of an onset without missing any onset due to insufficient frequency resolution. This multi-resolution procedure seems to be a good approach since the best onset detection algorithms also use this mechanism. If multiple onsets are located in close vicinity, they are annotated as multiple onsets.

The dataset contains 321 audio excerpts taken from various sources. 87 tracks were taken from the dataset used in [11], 23 from [2], and 92 from [13]. All annotations were manually checked and corrected to match the annotation style outlined above. The remaining 119 files were newly annotated and contain the vast majority of the 27,774 onsets of the complete set.

Although musically correct, the precise annotations (raw onsets) do not necessarily represent human perceptions of onsets. Thus, all onsets within 30 ms were combined into a single one located at the arithmetic mean of the positions [1], which resulted in 25,966 combined onsets used for evaluation. The dataset can be roughly divided into six main groups (Table 3.1).

Type of audio	Files	Raw onsets	Combined
Complex mixtures	193	21,091	19,492
Pitched percussive	60	2,981	2,795
Non-pitched perc.	17	1,390	1,376
Wind instruments	25	822	820
Bowed strings	23	1,180	1,177
Vocal	3	310	306
ALL	321	27,774	25,966

Table 1. Description of the used dataset: Pitched percussive (e.g., piano, guitar), non-pitched percussive (e.g., percussion), wind instruments (e.g., sax, trumpet), bowed string instruments (e.g., violin, kemence), monophonic vocal music and complex mixtures (e.g., jazz, pop, classical music)

3.2 Measures

For evaluation, the standard measures precision, recall, and F-measure were used. An onset is considered to be correctly detected if there is a ground truth annotation within ± 25 ms around the predicted position. This rather strict evaluation method (also used in [11] and [6] for percussive sounds) was chosen because it gives more meaningful results - especially in online onset detection - than an evaluation window of ± 50 ms as used in [2, 9, 18].

An important factor in the evaluation is how false positives and negatives are counted. Let us assume that two onsets are detected inside the detection window around a single annotation. If tolerant counting is used, no false positives are counted. Every single detection is considered a true positive, since there is an annotated onset within the detection window. This is often referred to as *merged* onsets. If counted in a strict way, all annotated onsets can only be matched once, i.e., two detections within the detection window of a single onset are counted as one true positive and one false positive detection.

Since many papers do not explicitly describe the criteria, it must be assumed that the results were obtained with the first method (usually yielding better results). In this paper, we evaluated the stricter way, but with combined annotated onsets (not to be confused with merged onsets). The combining of onsets leads to less false negative detections if the algorithm reports only a single onset where multiple ones are annotated. Since most of the algorithms are not capable reporting multiple consecutive onsets, this results in a more fair comparison.

3.3 Parameter selection

The peak-picking parameters $w_1 ... w_5$ and the fixed threshhold δ introduced in Section 2.3.3 were optimized by a grid search over the whole set for each method separately. As in [2, 9], we report the best performance for each method using the optimized global parameter set. For online detection ($w_2 = w_4 = 0$), the optimal values for w_3 were found to be between 4 and 12, $w_1 = 3$, and $w_5 = 3$. For the offline case, $w_2 = 3$, $w_4 = 1$ and $w_5 = 0$ yielded the best results (w_1 and w_3 were left unchanged). The adaptive whitening parameters $m = 10$ and $r = 0.005$ were found to be generally good settings and were used for all ODFs in the experiments. The compression parameter λ (Section 2.1.2) was chosen to be between 0.01 and 20. The neu-

[1] To better predict the perceived position of an onset, psychoacoustical knowledge must be applied. Since the masking effects involved depend on both loudness and frequency of an onset, they are not applied here. For the evaluation of onset detection methods as in this paper, the selected method of combination is adequate.

ral networks are trained and evaluated using 8-fold cross validation on disjoint training, validation, and test sets.

All parameters were optimized on the dataset and left unchanged for the unnormalized penalty task.

4. RESULTS AND DISCUSSION

4.1 Comparison of different ODFs

Table 2 lists the results for all algorithms working in online mode on the complete dataset using the peak detection method described in Section 2.3.4. It shows that application of adaptive whitening and use of a logarithmic magnitude both outperform the traditional methods without any preprocessing. Both preprocessing methods compress the magnitude and hence emphasize higher frequency bands that are important for detecting percussive onsets. Furthermore, our proposed method (*SF log filtered*) clearly outperforms all the other methods (apart from the reference *OnsetDetector.LL*). In particular, it is characterized by a high precision value due to the reduced number of false positives compared to the other methods. We believe that the filtering process reduces the spectrum to the most relevant components for onset detection. This may facilitate better distinction between signal changes that are arising from an onset and spurious, non-onset-related changes.

Online algorithm	% F-meas.	% Prec.	% Rec.
SF	74.5	76.3	72.8
SF aw	75.7	78.0	73.4
SF log	76.1	78.3	74.0
SF log filtered	**80.3**	**88.3**	73.5
CD	71.1	72.4	69.8
CD aw	75.8	76.4	**75.1**
CD log	74.1	77.4	71.1
WPD	69.7	68.8	70.6
WPD aw	71.4	70.8	72.0
WPD log	70.9	74.6	67.5
OnsetDetector.LL [4]	81.7	85.0	78.7

Table 2. F-measure, precision and recall of different onset detection algorithms using *online* peak-picking, where *aw* denotes adaptive whitening, *log* denotes the use of a logarithmic magnitude and *SF log filtered* is the method proposed in Section 2.5

.

Our tests showed that - if the parameters are properly chosen - the offline results are in the same range as the online results [2]. We deem this is a remarkable finding and think that the reasons for this behavior are the following: First, the audio tracks of the dataset have similar volume levels, which renders the normalization step less important. Second, when looking only at single independent frames, it seems reasonable that frames after the current onset frame do not carry much additional information. However, the superior results of the offline *OnsetDetector* (F-measure 86.6, precision 90.6, recall 83.0) suggest that using both past and future information contained in the magnitude spectrogram can be valuable to detect also the "harder" onsets (as reflected by the much higher recall value of this method).

[2] We observed an average gain in F-measure of 0.25% in offline mode

4.2 Unnormalized penalty

When dealing with unnormalized data, the investigated onset detection methods experience different levels of performance loss. As shown in Figure 3, our proposed onset detection method exhibits superior performance at all attenuation levels and is only beaten by the $OnsetDetector.LL$, that is unaffected by any volume changes. This shows the power of machine learning techniques that do not depend on predefined peak-picking threshholds. The methods using adaptive whitening score third, which seems reasonable as these methods include an implicit normalization using past frames. Computing the difference of two adjacent frames of the logarithmic spectrum (*SF log*) has the effect of dividing the magnitude at frame n by that at frame $n-1$, resulting in the relative magnitude change rather than the absolute difference. This makes the spectral flux obtained with logarithmic magnitudes more robust against absolute volume changes, compared to the standard variant (SF).

Finally, methods using the logarithmic magnitude spectrum performed better at lower volume levels when using a high value of the compression parameter λ.

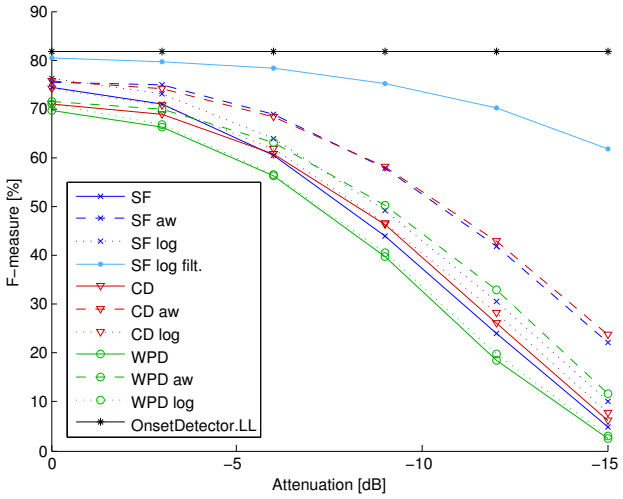

Figure 3. Performance of the online methods at different attenuation levels.

4.3 Remarks

In this paper, we give only results for the complete dataset. Results for subsets (organized by audio type and author) obtained with different detection window sizes can be found online at `http://www.cp.jku.at/people/boeck/ISMIR2012.html`.

5. CONCLUSIONS

In this paper we have evaluated various onset detection algorithms in terms of their suitability for online use, focus-

ing on the preprocessing and peak detection algorithms. We have shown that using logarithmic magnitudes or adaptive whitening as a preprocessing step results in improved performance in all methods investigated. When the parameters for peak detection are chosen carefully, online methods can achieve results in the same range as those of offline methods.

Further, we have introduced a new algorithm which outperforms other preprocessing methods. It copes better with audio signals of various volume levels, which is of major importance for onset detection in real-time scenarios.

Apart from that, machine learning techniques like neural network based methods are much more robust against volume changes in online scenarios and are the methods of choice if enough training data is available.

6. ACKNOWLEDGMENTS

This research is supported by the Austrian Science Funds (FWF) under the projects P22856-N23, TRP-109 and Z159 "Wittgenstein Award". For this research, we have made extensive use of free software, in particular python and GNU/Linux. Further, we are grateful to the authors Bello and Holzapfel for making their onset dataset publicly available.

7. REFERENCES

[1] MIREX 2011 onset detection results. `http://nema.lis.illinois.edu/nema_out/mirex2011/results/aod/`.

[2] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047, 2005.

[3] J. P. Bello, C. Duxbury, M. Davies, and M. B. Sandler. On the use of phase and energy for musical onset detection in the complex domain. *IEEE Signal Processing Letters*, 11(6):553–556, 2004.

[4] 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)*, 2012.

[5] P. Brossier, J. P. Bello, and M. D. Plumbley. Real-time temporal segmentation of note objects in music signals. In *In Proceedings of the International Computer Music Conference (ICMC)*, 2004.

[6] N. Collins. A comparison of sound onset detection algorithms with emphasis on psychoacoustically motivated detection functions. In *Proceedings of the 118th AES Convention*, pages 28–31, 2005.

[7] R. B. Dannenberg. An on-line algorithm for real-time accompaniment. In *Proceedings of the 1984 International Computer Music Conference*, pages 193–198, 1984.

[8] N. Degara, M. Davies, A. Pena, and M. D. Plumbley. Onset event decoding exploiting the rhythmic structure of polyphonic music. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1228–1239, 2011.

[9] S. Dixon. Onset detection revisited. In *Proceedings of the 9th International Conference on Digital Audio Effects (DAFx)*, pages 133–137, 2006.

[10] C. Duxbury, J. P. Bello, M. Davies, and M. B. Sandler. Complex domain onset detection for musical signals. In *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx)*, 2003.

[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 Conference on Music Information Retrieval (ISMIR)*, pages 589–594, 2010.

[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*, 2011(1):1–13, 2011.

[13] 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*, 18(6):1517–1527, 2010.

[14] A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 6, pages 3089–3092, 1999.

[15] A. Lacoste and D. Eck. A supervised classification algorithm for note onset detection. *EURASIP Journal on Applied Signal Processing*, pages 153–153, 2007.

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

[17] E. D. Scheirer. Tempo and beat analysis of acoustic musical signals. *The Journal of the Acoustical Society of America*, 103(1):588–601, 1998.

[18] D. Stowell and M. D. Plumbley. Adaptive whitening for improved real-time audio onset detection. In *Proceedings of the International Computer Music Conference (ICMC)*, 2007.