ONLINE REAL-TIME ONSET DETECTION WITH RECURRENT NEURAL NETWORKS

Sebastian Böck, Andreas Arzt, Florian Krebs, Markus Schedl

Department of Computational Perception
Johannes Kepler University, Linz, Austria

ABSTRACT

We present a new onset detection algorithm which operates online in real time without delay. Our method incorporates a recurrent neural network to model the sequence of onsets based solely on causal audio signal information. Comparative performance against existing state-of-the-art online and offline algorithms was evaluated using a very large database. The new method – despite being an online algorithm – shows performance only slightly short of the best existing offline methods while outperforming standard approaches.

1. INTRODUCTION

Onset detection is the process of locating events in an audio signal (e.g., a singing voice, a played note, or any other sounds). Various methods have been proposed over the years, but most of them work only in offline mode. [1] and [2] give good overviews of standard methods, and [3] propose enhancements to several of these. Traditional onset detection methods usually incorporate only spectral and/or phase information of the signal. However, unlike current top-performance algorithms, they neither employ machine learning techniques nor use probabilistic information. For example, the approaches presented in [4, 5] use neural networks and that in [6] a Hidden Markov model. They all have in common, that they usually work only in offline mode because the peak-picking methods used rely on future information to determine the location of an onset.

Only few algorithms have been designed specifically for online scenarios [7], where the aim is to minimize the delay between the occurrence of the onset in the audio signal and its reporting. Instantaneously detected onsets are a prerequisite for all kinds of real-time applications, ranging from beat-tracking and tempo estimation methods to look-ahead compressors for live audio processing.

2. SYSTEM DESCRIPTION

The proposed system is based on the state-of-the-art onset detection algorithm that won the last two years' MIREX onset detection contests [8, 9]. The system was originally proposed in [5] and has since been modified and enhanced considerably, as the improvements in the MIREX results show. In the next sections, we present the modifications and enhancements made in order to enable the system to work in real-time online scenarios.

The system is structured as depicted in Figure 1 and comprises three main processing steps: signal pre-processing, neural network onset prediction, and peak post-processing. As input, the system takes a discretely sampled audio signal and transfers it to the frequency domain via three parallel *Short-Time Fourier Transforms* (STFT) with different window lengths. The information obtained is then fed into the recurrent neural network to detect the next occurring onset in the audio stream. Finally, simple post-processing is used to report the onsets instantaneously while minimizing the number of false detections.

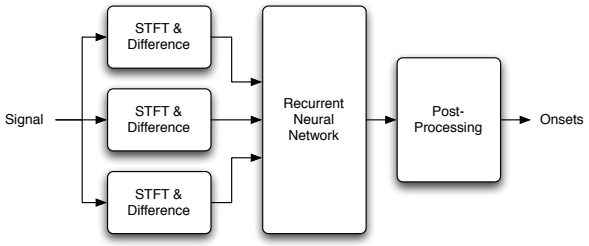

Figure 1: *Online real-time onset detection system overview.*

2.1. Audio signal pre-processing

The system processes the audio signal frame-wise with adjacent frames 10 ms apart (i.e., the resulting frame rate is 100 frames per second). The audio signal is transformed to the frequency domain with the Short Time Fourier Transform (STFT). Three parallel STFTs with different window lengths are used to capture both very recent and also 'older' information. The right edges of Hann windows are aligned at the current position of the audio signal and the windows are normalized to have equal area (cf. Figure 2). The sizes used are 512, 1024, and 2048 samples, which corresponds to periods of 11.61, 23.22, and 46.44 ms, respectively, at a sample rate of 44,100 Hz.

The linear magnitude spectrogram of each STFT is then filtered to obtain a compressed representation. We investigated various strategies to reduce the dimensionality of the input vector for the neural network. Using a filterbank with frequencies aligned to the Bark scale yielded a good compromise between performance and size of the neutral network input vector. The edge frequencies of the bins correspond to the frequencies of the 24 critical bands of the Bark scale, and triangular filters (with an area normalized to one) are used to sum the multiple frequency bins of the STFT to a single one. To transform the values to a range better suited to the downstream neural network step, we chose a logarithmic representation of the Bark spectrograms.

Since onsets are characterized by a rise in energy in their attack phase, the differences relative to preceding frames are also included in the input vector. The exact delay τ for calculating the difference is determined according to the STFT length such that

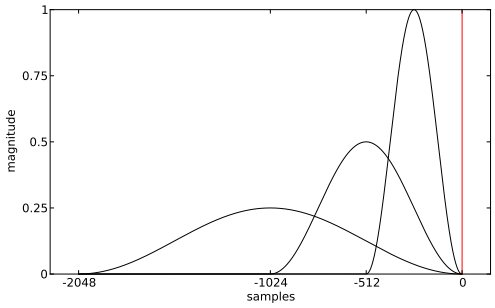

Figure 2: *Window functions applied to audio signal before STFT, with the current position of the audio signal indicated by a vertical line.*

the overlap of the two frames is 0.5. This results in τ values of 1, 2, and 4 for the STFT window lengths of 512, 1024, and 2048 samples. Although technically speaking it is a quotient because it is calculated using logarithmic representations, we use the term "difference" between two frames. The three parallel Bark-filtered spectrograms and the differences make up the 144-dimensional input vector for the neural network.

2.2. Neural Network

To work in a real-time online scenario, the neural network of the offline approach [5] had to be changed considerably. Since bidirectional neural networks violate causality, they are not suitable for this task and were replaced by a unidirectional one. Also, the Long Short-Term Memory (LSTM) units used in the hidden layer were replaced by standard units with a hyperbolic tangent activation function. This reduces the connections in the recurrent hidden layers by a factor of four, because the standard units do not require the gates of the LSTM units to be connected. Although LSTM units are able to model a wider temporal context, normal units perform similarly well because the temporal context for onset detection is limited to only a few frames. The overall topology of the network, consisting of three fully connected recurrent hidden layers with 20 units each, is retained. The modifications listed reduce the computational complexity of the system and make it suitable for real-time processing.

2.2.1. Network Training

The network was trained as a classifier with supervised learning and early stopping on a 75% portion of the complete dataset described in Section 3. Each audio sequence was pre-processed as described above and presented to the network for learning. The network weights were initialized with random values following a Gaussian distribution with mean 0 and standard deviation 0.1. Standard gradient descent with backpropagation of the errors was used to train the network. To avoid over-fitting, the performance was evaluated after each training iteration on a separate validation set (a disjoint 15% of the training set chosen at random). If no improvement was observed for 20 epochs, training was stopped, and the network state with the best performance on the validation set was subsequently used.

When training a neural network to detect an upcoming onset, various strategies for target placement are possible: placing them at the real ground-truth positions and training the classifier as in an offline scenario, or displacing the targets forward or back by one frame. Although neural networks can adapt to a target displacement, the method of training with correctly located targets in combination with the post-processing described in Section 2.3 yielded the best classification results. Thus, only the post-processing method had to be modified for an online scenario.

2.2.2. Network Testing

The output of the network is an onset activation function with values in the range of $[0 \ldots 1]$ which represent the probabilities of onsets at given positions. Figure 3 shows a typical onset activation function with clearly visible peaks at the annotated positions.

2.3. Post-processing

Since no future values are available in online mode, the traditional approach of finding local maxima in the thresholded onset activation function cannot be applied here. Instead, the onset is predicted at the center of the first frame that follows the activation function exceeding a given threshold, determined on the validation sets by 8-fold cross-validation.

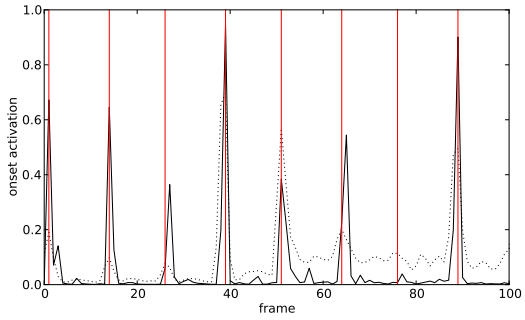

Figure 3: *Onset activation function (output of the neural network) of the system for a 1-second-excerpt of a pop song shown as a solid black line. Annotated onsets are indicated by vertical lines and the normalized detection function obtained with spectral flux is plotted as a black dotted line.*

Compared to simple signal-based onset detection methods, the main advantage of using a neural network is that its onset activation function has a very low noise floor with high peaks at the onset positions (see Figure 3, solid black line). Thus, a very low threshold can be used to detect the onsets as early as possible without risking many false detections. To prevent repeated reporting of an onset (and thus producing numerous false positive detections), an onset is only reported if no onsets have been detected in the previous two frames (20 ms).

In the rare case of a slowly rising onset activation function (which exceeds the threshold), this peak-picking method could lead to some early false positive detections. To give an estimate of the penalty, we also evaluated our new algorithm with an offline peak picking algorithm which uses only local maxima after thresholding of the onset activation function.

3. DATA

In online real-time processing, the definition of onsets is crucial. An onset is usually defined as the exact time a note or instrument starts sounding after being played. However, this timing is difficult to determine, and it is therefore impossible to annotate the real onset timing in complex audio recordings with multiple instruments, voices, and effects.

The most commonly used method for onset annotation is marking the earliest time point at which a sound is audible to humans. This instant cannot be defined by pure measures (e.g., minimum increase of volume or sound pressure), but is a complex mixture of various factors. All annotations of the dataset try to match the onset time as accurately as possible. Compared to synthesized sounds generated from MIDI, this generally leads to a delay in the range of a few milliseconds (determined by manual correction of a piece of synthesized piano music to match the style of other annotations).

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 together to capture the precise timing of an onset without missing any onsets due to insufficient frequency resolution. This multi-resolution procedure seems to be a good approach since the best onset detection algorithms also use it internally. If multiple onsets are located in close vicinity, they are annotated as multiple onsets. For reference, Figure 4 shows a piano chord in which the individual notes were not played perfectly simultaneously with two individual annotations.

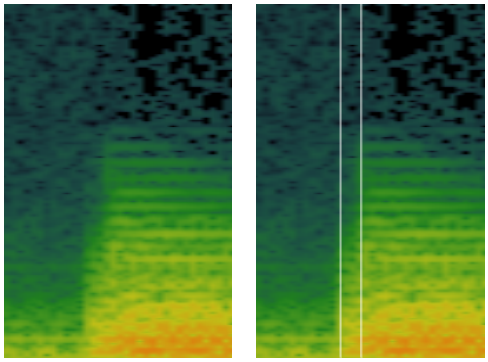

Figure 4: *Zoomed-in spectrogram of a piano chord with two notes played 4 ms apart. The pictures show a period of 50 ms (identical to the detection window used for evaluation) taken with an STFT length of 512 samples and 86% overlap.*

The dataset consists of 327 audio excerpts taken from different sources. 87 tracks were taken from the dataset used in [5], 23 from [1], and 92 from [10]. All annotations were manually checked and corrected where needed. The remaining 125 files were newly annotated during the evolutionary process of the offline OnsetDetector [9]. The complete set contains 28,067 onsets according to the annotation style outlined above. Although musically correct, the precise annotations do not necessarily represent the human perception of onsets. Thus, all onsets within 30 ms are combined into a single one located at the arithmetic mean of the positions, which results in 26,223 combined onsets used for evaluation.

4. RESULTS AND DISCUSSION

We used 8-fold cross-validation with the standard measures precision, recall, and F-measure to evaluate our approach. An onset is considered correctly detected if there is a ground-truth annotation within ± 25 ms around the predicted position. We refer to this region as *detection window*. We chose this relatively strict evaluation method (also used in [5] and [2] for percussive sounds) because it gives more meaningful results – especially for the task of online onset detection – than the detection window of ± 50 ms used in [1, 3, 7]. All annotated onsets can only be matched once: two detections within the detection window of a single annotated onset are counted as one true positive and one false positive detection.

4.1. Performance Comparison

Table 1 lists the results for the complete dataset described in Section 3. The new online algorithm OnsetDetector.LL[1] is compared to its offline variant and also to all detection methods implemented by `aubio`[2]. Our new online algorithm was evaluated by 8-fold cross-validation with disjoint training, validation, and test sets. Since the offline variants of OnsetDetector [8, 9] were trained with audio material originating from the same dataset, we took particular care to avoid using the same audio excerpts for both training and testing. The parameters of the `aubio` results were optimized on the complete dataset (optimized threshold and shifting of the reported onsets to best match the annotations), and hence represent the maximum achievable performance by traditional algorithms when perfectly adapted to the test set.

It should be noted that the upper part of Table 1 lists offline algorithms which incorporate future information for onset detection, while our new algorithm solely relies on past information to detect upcoming onsets.

Algorithm	Precision	Recall	F-measure
OnsetDetector.2010	0.857	0.796	0.826
OnsetDetector.2011	0.906	0.830	0.866
`aubio` default *	0.718	0.690	0.704
`aubio` specdiff *	0.653	0.650	0.652
`aubio` phase *	0.516	0.600	0.555
`aubio` complexdomain *	0.700	0.690	0.695
`aubio` hfc *	0.750	0.733	0.742
`aubio` energy *	0.608	0.572	0.590
`aubio` kl *	0.672	0.670	0.671
`aubio` mkl *	0.647	0.599	0.622
NEW OnsetDetector.LL	0.850	0.787	0.817
NEW OnsetDetector.LL †	0.897	0.789	0.840

Table 1: *Comparison of performance results using the complete dataset with a detection window of ± 25 ms. All algorithms in the upper part operate offline, while only the new one works in online mode. Asterisks mark results obtained with parameters optimized on the complete dataset. The last result, denoted with † was obtained with an offline peak-picking method.*

[1] The onset detector is named after Lucky Luke, the cowboy known to "shoot faster than his shadow", because it is able to detect an onset before a human can hear it.

[2] `http://aubio.org/` version 0.3.2

As expected, the new OnsetDetector.LL falls short of the performance of state-of-the-art offline onset detection algorithms (i.e., the offline version of OnsetDetector) but clearly outperforms other onset detection methods such as *spectral flux*, *complex domain*, *high frequency content*, and combinations thereof (as reflected by the result obtained with the `aubio` algorithm), even when they were perfectly adapted to the test set. This shows the strength of the new online onset detection algorithm.

As mentioned in Section 2.3, the chosen online peak-picking method can lead to false positive detections. The last line of Table 1 shows the performance obtained with an offline peak-picking method, which yields a reduced number of false positive detections as reflected by absolute increase in precision of almost 5%. Comparison of this result with the offline *OnsetDetector* suggests that using future information is advantageous for onset detection.

4.2. Detailed Evaluation

Table 2 presents the performance results of the new algorithm evaluated on the complete data set with various detection window sizes. It exhibits remarkable and stable performance down to a window size of ± 25 ms around the ground-truth positions of the onsets, regardless of the quality measure used. Only when evaluated with smaller window sizes does the performance drops considerably. Although a detection window of ± 10 ms is close to the accuracy of a manual annotation (which is typically around ± 2 ms for percussive sounds and up to ± 10 ms for soft onsets generated by instruments such as string or woodwind instruments), the algorithm continues to identify the majority of onsets correctly.

Window	Precision	Recall	F-measure	Error
± 50 ms	0.885	0.786	0.833	0.1 ± 11.4 ms
± 35 ms	0.880	0.781	0.828	0.2 ± 9.1 ms
± 30 ms	0.876	0.778	0.824	0.2 ± 8.6 ms
± 25 ms	0.870	0.772	0.818	0.3 ± 7.9 ms
± 20 ms	0.852	0.757	0.802	0.5 ± 7.2 ms
± 15 ms	0.811	0.720	0.763	0.6 ± 5.9 ms
± 10 ms	0.722	0.642	0.680	0.5 ± 4.7 ms

Table 2: *Performance results of the new algorithm on the complete dataset with different detection windows. Additionally, the mean and standard deviation error of all correctly detected onsets relative to their ground-truth annotations are given.*

4.3. Computational Cost

The system works on a frame-by-frame basis with a hop-size of 10 ms. For each audio frame, pre-processing, computing the output activations of the neural network, and post-processing take a constant amount of time and are easily done in real time on a single core of a 2.26 GHz Intel Core 2 Duo CPU.

5. CONCLUSIONS

We have presented a new onset detection algorithm specifically designed for real-time online detection of musical onsets in audio signals. It achieves performance close to current state-of-the-art offline onset detection algorithms while introducing zero delay between the audio signal and the reporting of an onset. On modern hardware, the computational processing can easily be achieved in real time.

6. ACKNOWLEDGMENTS

This research is supported by the Austrian Science Funds (FWF): P22856-N23, TRP-109, and Z159 "Wittgenstein Award". Special thanks go to Martin Gasser for providing the *Flower* real-time framework to implement the system.

7. REFERENCES

[1] 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.

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

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

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

[5] 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)*, 2010, pp. 589–594.

[6] N. Degara, M. Davies, A. Pena, and M. Plumbley, "Onset event decoding exploiting the rhythmic structure of polyphonic music," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1228–1239, October 2011.

[7] D. Stowell and M. Plumbley, "Adaptive whitening for improved real-time audio onset detection," in *Proceedings of the International Computer Music Conference (ICMC)*, 2007, pp. 312–319.

[8] "MIREX 2010 onset detection results," `http://nema.lis.illinois.edu/nema_out/mirex2010/results/aod/`, 2010.

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

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