Onset Detection in Somax

Rev. 0.2.0

Joakim Borg

January 21, 2022

Version History

Revision	Date	Author(s)	Description
0.2.0	21 jan 2022	JB	Added Appendix A
0.1.0	13 jan 2022	JB	Initial report

Contents

0.1 Introduction

Onset detection, i.e., the task of detecting and segmenting an audio file into musically meaningful discrete events, is crucial in the Somax system for several reasons. First of all, the system uses a form of concatenative synthesis for its generation, meaning that the corpus that Somax is trained on and from which the system generates its content requires a meaningful segmentation, For audio-based content, this is typically on the onset level. Other segmentation levels are possible (for example segmentation by beat or by bar), but even in these cases, onset detection is typically a pre-requirement. Secondly, onset detection is also necessary for the real-time listening modules that Somax use to handle audio input from a musician. The detected onsets determine both when to trigger new output from the system as well as when to segment the input stream into events and send its analysis of continuous descriptors such as pitch and chroma to the discretized model.

This effectively means that there are two cases of onset detection in Somax, the first one being computed offline (i.e. with access to information about the entire audio content it is computed on) on a corpus (typically one or several audio files) specified by the user, and the second one one being computed in real-time on an input signal from a musician.

The current Somax architecture consists of two components: a Python back-end that handles scheduling and analysis of the runtime modules on a discrete event level as well as all offline processing of corpora, and a MaxMSP front-end that handles all real-time processing of audio signals into discrete events. It's also important to be aware of that Somax is intended to be genre-agnostic, meaning that it is intended to work well with content of any genre, ranging from classical music to unpitched percussive music to modern pop/rock music as well as content that typically doesn't fit in a musical genre, for example speech. For this reason there are a number of different onset detectors implemented in both the python backend as well as the MaxMSP front-end. Among these are bonk [12], a low-latency onset detection algorithm designed for percussive sounds, yingb [5], a statistical model based on the Yin pitch detection algorithm [10] to detect onsets from monophonic audio signals, onseg [15], a spectrum-based onset detector for polyphonic music and finally a spectral flux-based algorithm for polyphonic onset detection loosely based on [8]. All four implementations are real-time compatible, but the last one has a number of additional features to improve the performance for offline computation. Bonk, yingb and onseg are implemented in the MaxMSP environment while the spectral flux-based onset detector is implemented in Python.

Several of these onset detection algorithms have been designed for real-time artistic use in particular, but lack formal evaluation. There are several reasons for this (where time and resource constraints surely is one), but the most prominent issue is the fact that there's no clear target group for

either of them – the majority of them are not designed with a particular instrumentation and/or genre in mind, thereby making formal evaluation difficult. For the HyVibe guitar, the context is different – there's a clear target group both in terms of instrumentation (acoustic guitar) and genre (pop, rock, jazz, etc.). For this reason, this report will focus on evaluating the relevant implemented onset detectors with regards to HyVibe's target group.

0.1.1 State of the Art

Little has happened in the domain of onset detection during the past couple of years. The Music Information Retrieval Evaluation eXchange (MIREX) was continuously monitoring (offline) onset detection up until 2018, where the same implementation [14] had won for five years in a row [4][1]. This implementation makes use of convolutional neural networks and is available in the `madmom` [7] library in Python, but is not real-time compatible. A real-time adaptation of an earlier version of the algorithm was made with recurrent neural networks in 2012 [6] and is to this date considered the state of the art in online onset detection. A compilation of real-time onset detection algorithms was however made by the same author in 2012, showing that there's very little difference in performance between the the RNN-based onset detector and state of the art spectral flux-based onset detectors [8], and in 2013 an improved version of the spectral flux-based onset detector consistently outperformed the RNN-based one for certain datasets [9]. The spectral-flux based classifier used in Somax is loosely based on this implementation and briefly described in [11]. A full description of the algorithm is available in section 0.2.

It's also worth noting that these classifiers have been evaluated on a dataset consisting of a mix of polyphonic and monophonic music of varying quality, where a majority (70.5%) of the excerpts are either monophonic or unpitched [3]. None of the classifiers have been evaluated on a dataset consisting solely of acoustic guitar excerpts.

0.2 Compared Algorithms

As mentioned in section 0.1, four onset detection algorithms are implemented in Somax: bonk [12], yingb [5], onseg [15] and a spectral flux-based [11], but the former two are designed specifically for percussion and

[1] the CNN-based onset detector has occasionally been outperformed in certain categories, for example in the category "Solo Plucked Strings" in the same year by Roebel et al. who implemented another CNN-based offline onset detector [13], but the overall performance has consistently been higher for [14]. See [4] for details.

monophonic instruments respectively, and will therefore not be evaluated here as neither category is relevant for the HyVibe guitar.

0.2.1 Onseg (PiPo)

The algorithm used for real-time onset detection in the MaxMSP front-end of Somax is not formally described in any paper, but it was introduced in [15] and the source code is available at [2]. The algorithm can be described as follows:

Given a real-time monophonic signal $x[n] \in \mathbb{R}^{\infty}_{[-1,1]}$, we compute the spectrogram $|X(m,k)|^2 \in \mathbb{R}$ for each frame $m = \lfloor n/L \rfloor$ with $k \in N$ frequency bins (base-2) and a hop length $L \in \mathbb{N}$. By applying ITU-R 468 weighting on the spectrogram, we get the equal-loudness spectrogram $\left|X_m^{(\text{loud})}(m,k)\right|^2$. The energy $\xi[m] \in \mathbb{R}$ for frame m is simply defined as the sum of the loudness spectrogram, i.e.

$$\xi[m] = \sum_{k=0}^{N-1} \left|X^{(\text{loud})}(m,k)\right|^2, \tag{1}$$

and the onset strength function $\text{ODF}[m]$ is defined as the difference between the energy of the current frame and of the median of the F previous frames, i.e.

$$\text{ODF}[m] = \xi[m] - \text{med}\left(\xi[n-1], \ldots, \xi[n-F]\right), \tag{2}$$

where $F \in \mathbb{Z}_+$ is a user-controlled parameter ("filter length"). The frame m is an onset if

$$\begin{cases} \text{ODF}[m] \geq T & (3) \\ m - m_{\text{prev}} \geq M & (4) \end{cases}$$

for some user-defined threshold $T \in \mathbb{R}_+$ ("threshold") and minimal inter-onset interval $M \in \mathbb{N}$ ("minimum interval") where m_{prev} denotes the index of the previously selected onset frame.

Note that the Onseg algorithm is not only designed to detect onsets but to segment a signal, i.e. detecting both onsets (start of a segment) and offsets (end of a segment). For this reason, it consists of a number of additional steps after detecting the onset, but these are not relevant for the online functionality of the Somax library as it only uses onsets.

0.2.2 Spectral Flux (Librosa)

Another onset detector is implemented in the Python back-end. This is primarily an offline onset detector loosely based on the best-performing spec-

tral flux onset detector in [6], but is implemented to be real-time compatible. This onset detector was introduced in [11] and the source code is available at [1]. At the moment, this classifier is only used in Somax when constructing the corpus but could easily be implemented in the MaxMSP front-end or directly in the HyVibe guitar.

Given a real-time monophonic signal $x[n] \in \mathbb{R}^{\infty}_{[-1,1]}$, we compute the Log Mel-spectrogram $\left|X^{(\text{mel})}(m,k)\right|^2 \in \mathbb{R}$ for each frame $m = \lfloor n/L \rfloor$ with $k \in N$ frequency bins (base-2) and a hop length $L \in \mathbb{N}$. We then define our onset strength function at frame m as the mean of the positive values of the difference between the current frame and the previous, i.e.

$$\text{ODF}[m] = \frac{1}{N}\sum_{k=0}^{N-1}\left[H\left(\left|X^{(\text{mel})}(m,k)\right|^2 - \left|X^{(\text{mel})}(m-1,k)\right|^2\right)\right]. \tag{5}$$

For offline onset detection, i.e. given a finite signal $x[n]$ segmented into $D \in \mathbb{N}$ frames, the onset detection function is normalized, i.e.

$$\text{ODF}[m] = \frac{1}{\max\limits_{m\in D}\text{ODF}[m]}\text{ODF}[m] \qquad \forall\, m \in 0,\dots D-1 \tag{6}$$

but this step is omitted for real-time onset detection. Finally, a frame m is an onset if

$$\begin{cases} \text{ODF}[m] = \max\left(\{\text{ODF}[m-w_1],\dots,\text{ODF}[m+w_2-1]\}\right) & (7)\\ \text{ODF}[m] \geq \text{mean}\left(\{\text{ODF}[m-w_3],\dots,\text{ODF}[m+w_4-1]\}\right) + T & (8)\\ m - m_{\text{prev}} \geq M & (9)\end{cases}$$

for user-defined parameters $w_1, w_2, w_3, w_4 \in \mathbb{N}$ controlling the window length, a minimum amplitude threshold $T \in \mathbb{R}_+$ and a minimum onset interval $M \in \mathbb{N}$. In the real-time case, $w_2 = w_4 = 0$, resulting in the following system:

$$\begin{cases} \text{ODF}[m] = \max\left(\{\text{ODF}[m-w_1],\dots,\text{ODF}[m]\}\right) & (10)\\ \text{ODF}[m] \geq \text{mean}\left(\{\text{ODF}[m-w_3],\dots,\text{ODF}[m]\}\right) + T & (11)\\ m - m_{\text{prev}} \geq M. & (12)\end{cases}$$

0.3 Evaluation

0.3.1 Dataset

For evaluation, we're using the GuitarSet dataset [17], a manually annotated dataset consisting of 183 minutes of acoustic guitar excerpts in five genres: rock, jazz, funk, bossa nova and singer-songwriter. The excerpts

are divided into two types of arrangements: "solo", where the musicians were asked to perform a solo over a given chord progression, resulting in a mostly (but not strictly) monophonic content, and "comp", where the musicians were asked to provide an accompaniment for the same chord progression, resulting in (mostly) polyphonic content. There are 30 different excepts in each arrangement recorded by six different musicians or in total 180 excerpts per arrangement. The content was recorded with hexaphonic microphones, but in this report, a monophonic mixdown of the six tracks was used (the folder `audio_mono-pickup_mix` provided in the dataset), as this was considered to be the closest to the audio quality of the HyVibe guitar's monophonic input. The excerpts were recorded with a sampling rate of $f_s = 44100$ Hz and the content is not normalized, therefore providing the entire spectrum of dynamics in a way that is similar to a real-time situation. To increase the size of the dataset and provide a more complex case, a third category "combined" was created, where the corresponding "solo" and "comp" tracks for each musician were mixed into a single track. By doing so, the final dataset consists of 540 excerpts over the three arrangements.

0.3.2 Methodology

There are a number of problems associated with evaluating onset detection on polyphonic content. First of all, the exact position of the onset can in many cases be difficult to determine when annotating the dataset (for the full description of the the annotation procedure of the dataset, see [17]). For this reason, we will consider two onsets to be aligned if the estimated onset is within ± 25 ms of the ground truth (i.e., the manually annotated onset), as per the standard for real-time onset detection defined in [8].

The second problem is related to the guitar in particular as well as the practical use-case in Somax, namely the fact that when several strings are struck close to simultaneously (e.g. when a chord is played), we only want to detect a single onset in order to avoid several almost simultaneous generation phases in Somax. For this reason, a minimal onset interval $\tau \in \mathbb{R}_+$ was defined so that any onset in the annotation occurring within τ ms of a previous onset was discarded. In this report, a value of $\tau = 100$ ms was selected through manually listening to the content with the onsets sonified, evaluating the quality of the result with different values for τ. In general, an interval smaller than 100 ms would for the given dataset in many cases result in multiple onsets detected for a single chord, while an interval larger than 100 ms would frequently lead to missed onsets in fast-paced excerpts. The user-controlled parameter M in equations 4 and 9 was adjusted accordingly so that

$$M = \left\lceil \frac{\tau}{1000} \cdot \frac{f_s}{L} \right\rceil \tag{13}$$

where f_s denotes the sampling rate and L the hop length as described in section 0.2.

The onset detection algorithms were evaluated with regards to precision, recall and F-measure according to the MIREX standard [3], i.e., where precision is defined as the proportion of detected onsets being real onsets,

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{14}$$

recall is defined as the fraction of actual onsets detected

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

and F-measure is defined as the harmonic mean of the precision and recall, i.e.

$$\text{F-measure} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \tag{16}$$

where TP, FP and FN denote number of true positives, false positives and false negatives respectively. The MIREX standard also defines a category "Doubled Onset Rate" (several detections for one ground-truth onset), but as the precision for this category (± 25 ms, i.e. a window of 50 ms) is below the given τ, this was not used in the evaluation.

The evaluation procedure was done in two steps where the first step was to find the optimal parameters for each onset detector using a grid search and the second step to evaluate the entire dataset with the optimal parameters

0.3.3 Grid Search

For each onset detection algorithm, the grid search was performed over the given user-controlled parameters in several steps, where in each step, the onsets were estimated and compared with the ground truth in regards to F-measure according to the procedure described in section 0.3.2. The initial values for each parameter were selected in a window around the default values (as defined in the source code) for each parameter. Given k evaluated parameters and a window of n different values for each parameter, we get a time complexity of $\mathcal{O}\left(n^k\right)$ per search per evaluated excerpt.[2] Two excerpts

[2]Note that a grid search is in this case applicable despite the exponential complexity due to the fact that there are a fairly low number of relevant user-controlled parameters: three for Onseg and four for the spectral flux-based onset detector. When fully parallelized on a MacBook Pro 2019 2.3GHz 8-core Intel Core i9, a single search for the spectral flux-based onset detector over all 30 excerpts and an average of $n = 6$ is computed in 37 minutes. The corresponding values for a similarly-performing classifier with seven parameters would be 5.5 days and 200 days for nine parameters.

were randomly selected from each arrangement (solo, comp, combined) for each genre (rock, jazz, funk, bossa nova, singer-songwriter), in total 30 excerpts. The result of each search was evaluated with respect to the F-measure, where the set of parameters resulting in the highest F-measure were selected as the next center for the following search. The following search would either

1. Continue the search with the same grid resolution centered around the selected parameters if one of the parameters was the end point of the search (for example, if some parameter $\theta \in \mathbb{Z}_{[0,10]}$ yields the highest F-measure at $\theta = 10$, a new grid search would be performed around $\theta \in \mathbb{Z}_{[5,15]}$).

2. Refine the resolution of the search (if applicable, e.g. for real-valued parameters), if none of the estimated parameters from the previous search were end-points of the current window. In this case, for a given parameter $\theta \in \mathbb{R}$ with an initial window of $n \in \mathbb{N}$ different values and a resolution $\rho \in \mathbb{R}$ between consecutive values for n, and a value $\hat{\theta}$ that optimizes the F-measure, the new window for the search was defined as $\mathbb{R}_{[\theta-\rho,\theta+\rho]}$ with a new resolution of ρ/n.

3. If the increase in F-measure of the optimal parameter set computed through step two is greater than $\epsilon = 0.1$ compared to the previous search, repeat step two, otherwise terminate the search.

For the Onseg algorithm described in section 0.2.1, the grid search was performed over the parameters L (hop length) F (filter length), and T (threshold). For the spectral flux-based algorithm described in section 0.2.2, the grid search was performed over the parameters L (hop length), w_1 (max window length), w_2 (mean window length) and T (threshold).

0.3.4 Evaluation on Full Dataset

Once the ideal parameters were estimated through the grid search over the 30 selected excerpts, the performance of each onset detection algorithm was evaluated over the entire dataset of 540 excerpts, using the procedure described in section 0.3.2, and are presented in the following section.

0.4 Results

0.4.1 Overall results

When evaluated over the entire dataset, the Onseg algorithm achieved a precision of 67.75%, recall of 59.82% and F-measure of 62.75% using the optimal parameters $L = 128$, $F = 26$ (corresponding to 75 ms) and $T = 2.2$.

The grid search was computed in five steps and achieved an optimal F-measure of 65.30% for the 30 excerpts. The spectral flux-based algorithm achieved a precision of 69.38%, a recall of 60.02% and an F-measure of 63.54% using the optimal parameters $L = 128$, $T = 0.45$ $w_1 = 70$ (corresponding to 203 ms), $w_3 = 215$ (corresponding to 624 ms). In this category, the spectral flux-based algorithm slightly outperformed the Onseg algorithm in all three measures.

Onset Detector	Precision	Recall	F-Measure
Onseg	0.6775	0.5982	0.6275
Spectral Flux	**0.6938**	**0.6002**	**0.6354**

Figure 1: Overall results for the Onseg and spectral flux-based onset detectors

0.4.2 Results per Arrangement

The results per arrangement for the Onseg algorithm and the spectral flux-based algorithm can be seen in figures 2 and 3 respectively. We see that the Onseg algorithm outperforms the spectral flux-based algorithm in both the "solo" and "combined" categories with F-measures of 61.42% and 68.71% respectively, while the spectral flux-based algorithm greatly outperforms the Onseg algorithm in the "comp" category with an F-measure of 67.86%.

Arrangement	Precision	Recall	F-Measure
Solo	0.6191	0.6158	**0.6142**
Comp	0.6232	0.5652	0.5811
Combined	0.7901	0.6136	**0.6871**

Figure 2: Results for the Onseg algorithm by arrangement

Arrangement	Precision	Recall	F-Measure
Solo	0.5825	0.5558	0.5645
Comp	0.7143	0.6642	**0.6786**
Combined	0.7845	0.5807	0.6632

Figure 3: Results for the spectral flux algorithm by arrangement

0.4.3 Results per Genre

The results per genre for the Onseg algorithm and the spectral flux algorithm can be seen in figures 4 and 5 respectively. The Onseg algorithm achieves better results in the jazz and singer-songwriter categories, while the spectral flux algorithm achieves better results in the rock, bossa Nova and funk categories. The differences in performance between the two are however not as emphasized as the difference between the two in terms of arrangement, apart from in the Funk category, where the difference in precision is huge (0.105).

Genre	Precision	Recall	F-Measure
Jazz	0.6466	0.6017	**0.6153**
Rock	0.7315	0.6057	0.6549
Bossa Nova	0.6729	0.6124	0.6344
Singer-Songwriter	0.7475	0.6055	**0.6609**
Funk	0.5888	0.5657	0.5718

Figure 4: Results for the Onseg algorithm by genre

Genre	Precision	Recall	F-Measure
Jazz	0.6648	0.5855	0.6127
Rock	0.7377	0.6101	**0.6604**
Bossa Nova	0.6899	0.6275	**0.6499**
Singer-Songwriter	0.7157	0.5999	0.6432
Funk	0.6938	0.5782	**0.6111**

Figure 5: Results for the spectral flux algorithm by genre

0.5 Discussion & Future Work

In the state of the art of onset detection algorithm evaluation, we're nowadays used to seeing F-measures close to 90% for offline algorithms [4] and above 80% for online algorithms [9] on certain datasets. In that context, the overall F-measures of 62.75% (Onseg) and 63.54% (spectral flux) might seem discouraging at first. It's however important to be aware of that these

are evaluated on much simpler datasets, for example the MIREX05 dataset where less than a third of the excerpts are polyphonic [3]. For the particular case of real-time onset detection in polyphonic acoustic guitar excerpts, there is no baseline before this report.

While the spectral flux-based onset detector is performing slightly better than the Onseg algorithm, the difference between the two algorithms is not big enough to prefer one over the other in the general case. In the current Somax architecture, Onseg is used in the front-end environment (as it is the one available in MaxMSP) and the spectral flux algorithm is used in the back-end (as it is the one available in Python). One benefit of both of these algorithms is their relative simplicity, both in terms of how complex they are to implement, computational complexity as well as memory requirements.

For further research that eventually would lead up to the implementation of an onset detector in the HyVibe guitar, the spectral flux-based onset detector would most likely serve as a better basis, as it is more aligned with the state of the art and could therefore benefit from the numerous preprocessing methods for real-time onset detectors suggested in [8], for example by using Adaptive Whitening (first introduced in [16]), for normalizing the onset strength function. Worth noting is that if the normalization step described in equation 6 is applied, the result of the initial search for the spectral flux-based onset detector achieves an F-measure of 0.7581 in the overall case (an improvement of .1047) and 0.8387 in the solo arrangement (an improvement of .2742!). Applying this step directly is obviously not possible in real-time case, but adaptive whitening could ideally somewhat approximate such normalization.

By having a baseline established, evaluating new onset detection algorithms as well as estimating the optimal parameters for the particular use cases of the HyVibe guitar will also be greatly simplified using the same code basis. While this report only evaluated onset detection algorithms already implemented in Somax, the same code basis could be used to evaluate both adaptations of the existing algorithms, other spectral flux-based approaches or even the neural network-based onset detection algorithms of the state of the art, should they be considered viable in terms of memory- and computation-complexity for the HyVibe guitar.

Bibliography

[1] Github: Librosa/onset.py. https://github.com/librosa/librosa/blob/main/librosa/onset.py. Accessed: 2022-01-10.

[2] Github: pipo/pipoonseg.h. https://github.com/ircam-ismm/pipo/blob/master/modules/PiPoOnseg.h. Accessed: 2022-01-10.

[3] Mirex 2005: Audio onset detect. https://www.music-ir.org/mirex/wiki/2005:Audio_Onset_Detect. Accessed: 2022-01-10.

[4] Mirex 2018: Audio onset detection - mirex05 dataset. https://nema.lis.illinois.edu/nema_out/mirex2018/results/aod/index.html. Accessed: 2022-01-10.

[5] G. Assayag, G. Bloch, M. Chemillier, A. Cont, and S. Dubnov. Omax brothers: a dynamic topology of agents for improvization learning. In Proceedings of the 1st ACM workshop on Audio and music computing multimedia, pages 125–132, 2006.

[6] 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. sn, 2012.

[7] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer. Madmom: A new python audio and music signal processing library. In Proceedings of the 24th ACM international conference on Multimedia, pages 1174–1178, 2016.

[8] S. Böck, F. Krebs, and M. Schedl. Evaluating the online capabilities of onset detection methods. In ISMIR, 2012.

[9] S. Böck and G. Widmer. Maximum filter vibrato suppression for onset detection. In Proc. of the 16th Int. Conf. on Digital Audio Effects (DAFx). Maynooth, Ireland (Sept 2013), volume 7, 2013.

[10] A. De Cheveigné and H. Kawahara. YIN, a fundamental frequency estimator for speech and music. The Journal of the Acoustical Society of America, 111(4):1917–1930, 2002.

[11] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto. librosa: Audio and music signal analysis in python. In Proceedings of the 14th python in science conference, volume 8. Citeseer, 2015.

[12] M. S. Puckette, T. Apel, and D. D. Zicarelli. Real-time audio analysis tools for Pd and MSP. In Proceedings, ICMC 98. Citeseer, 1998.

[13] A. Roebel, C. Jacques, and A. Aknin. Mirex 2018: Training cnn onset detectors with artificially augmented datasets. In UMR STMS-IRCAM, CNRS. Paris Sorbonne University, 2018.

[14] J. Schlüter and S. Böck. Improved musical onset detection with convolutional neural networks. In 2014 ieee international conference on acoustics, speech and signal processing (icassp), pages 6979–6983. IEEE, 2014.

[15] N. Schnell, D. Schwarz, J. Larralde, and R. Borghesi. Pipo, a plugin interface for afferent data stream processing modules. In International Symposium on Music Information Retrieval (ISMIR), 2017.

[16] D. Stowell and M. Plumbley. Adaptive whitening for improved real-time audio onset detection. In Proceedings of the 2007 International Computer Music Conference, ICMC 2007, pages 312–319. Citeseer, 2007.

[17] Q. Xi, R. M. Bittner, J. Pauwels, X. Ye, and J. P. Bello. GuitarSet: A Dataset for Guitar Transcription. In ISMIR, pages 453–460, 2018.

Appendix A

Hexaphonic Onset Detection

The main premise for the evaluation procedure described in section 0.3 and the following results in section 0.4 was that only a single pickup is used to record the input from the guitar, as is currently the case for the HyVibe guitar. In this case, we have a monophonic signal that may consist of either monophonic or polyphonic content, and the problem of onset detection becomes quite complex as we've seen in the report.

Another strategy would be to use a hexaphonic pickup to record each string of the guitar individually. In this case, we have six monophonic signals where each signal only has monophonic content, which would significantly reduce the complexity of the problem. Of course, this comes at the cost of a sixfold increase in computational cost, as the same algorithm has to be computed for each of the six strings, but if this cost can be afforded, the performance is likely to increase dramatically.

Evaluating these onset detection algorithms for the case of the hexaphonic pickup would require its own report. The GuitarSet [17] dataset that was used in this report could still be used in this case, as it consists of both monophonic and hexaphonic recordings, but the code and evaluation procedure would require significant revision, and is therefore out of scope for this report. What's presented in this appendix is merely a pre-study that can serve as a basis for the hexaphonic case. Here, we will instead evaluate how the two algorithms perform when evaluated over monophonic content. The same procedure as was described in section 0.3 will here be repeated with just the 180 excerpts in the "solo" category, using the same monophonic mixdown files as were described in section 0.3.1.[1] The purpose of this is to

[1] Note that this content is not strictly monophonic, it is rather of solistic character and certain excerpts may therefore contain some polyphony, but it's sufficiently close to true monophony to serve as a basis for this pre-study.

determine (a) whether the parameters for the grid search would be significantly different for monophonic content and thereby (b) if the two evaluated algorithms would perform better in this case.

Similarly to the procedure in section 0.3.1, here 30 excerpts (six from each genre) out of the 180 solo excerpts were randomly selected for the grid search, which was performed in the same manner as described in section 0.3.3 for each algorithm. The parameters from the grid search were then used to compute the performance of each algorithm over the entire set of the 180 solo excerpts.

A.1 Results

When evaluated over the set of the 180 excerpts in the "solo" category, the Onseg algorithm achieved a precision of 86.37%, recall of 68.71% and F-measure of 75.91% with the optimal parameters $L = 128$, $F = 5$, $T = 11.2$. The grid search was computed in three steps with an optimal F-measure of 76.92% for the 30 test excerpts. The spectral flux-based algorithm achieved a precision of 89.09%, a recall of 71.30% and an F-measure of 78.72%, thereby outperforming the Onseg algorithm in every category, with the optimal parameters $L = 128$, $T = 0.67$, $w_1 = 0$ (corresponding to 0 ms) and $w_3 = 1$ (corresponding to 2.90 ms). The grid search was computed in six steps with an optimal F-measure of 78.96% for the 30 test excerpts. Per genre, the differences were small—none of the measurements deviated more than ± 3 percentage points (± 4 for precision in Onseg)—and will for this reason not be presented individually.

Onset Detector	Precision	Recall	F-Measure
Onseg	0.8637	0.6871	0.7591
Spectral Flux	**0.8909**	**0.7130**	**0.7872**

Figure A.1: Overall results for the Onseg and spectral flux-based onset detectors over the set of the 180 excerpts in the "solo" category.

There are two important takeaways here. First of all, there's a dramatic increase in performance in comparison to table 1, with an increase in F-measure of 0.1316 for Onseg and 0.1518 for the spectral flux-based onset detector. The increase in precision is even larger with 0.1862 for Onseg and 0.1971 for the spectral flux-based. For recall, the increase is not as dramatic but significantly better with 0.0889 for the former and 0.1128 for the latter.

Secondly, the ideal parameters computed by the grid search were almost the opposite in every category compared to the parameters presented

in section 0.4. For the set of solo excerpts evaluated here, we have short windows (F, w_1, w_3) and high thresholds (T), while in the general evaluation in section 0.4 we have long windows and low thresholds. From a theoretical perspective it's reasonable that sparse (monophonic) signals can be evaluated over a shorter window, but it could technically be a result of the quantization of the grid search.

The grid search can essentially be seen as the solution to an optimization problem of a parametric function $f(\mathbf{v}) \in \mathbb{R}$ where $\mathbf{v}$ denotes our vector of parameters that may be either discrete or real-valued. Due to the computational cost of computing $f(\mathbf{v})$ over our dataset, we cannot evaluate all possible values of the hyperplane $\mathbf{v}$, we will have to use some quantization of $\mathbf{v}$, but as $f(\mathbf{v})$ lacks analytical solutions, we have no understanding of its gradient and there's therefore no guarantee that our grid search will converge into the global maximum.

Thus, in order to ensure that the difference in optimal parameters for the solo case and for the general case was significant, a final experiment was conducted where the spectral flux-based onset detector was evaluated over all 540 excerpts of the entire database using the parameters of the "solo" excerpt grid search (i.e., $L = 128$, $T = 0.67$, $w_1 = 0$ and $w_3 = 1$). This experiment achieved a precision of 87.77%, a recall of 45.68% and an F-measure of 56.17% (in the "all" category from section 0.4, i.e. over all 540 excerpts). In other words, these parameters would dramatically increase the precision of the algorithm at the cost of a significantly lower recall, thereby resulting in an overall lower F-measure. This indicates that the convergence towards these two different sets of parameters was indeed relevant for the two different cases.

A.2 Conclusion

No definite conclusions can be inferred for the hexaphonic case based on this study, but the results indicate that both onset detection algorithms evaluated in this report perform significantly better in the monophonic case. The parameters of the grid search converged towards short windows and high thresholds in the monophonic case, in comparison to the polyphonic case where they converged towards long windows and low thresholds. The short window/high threshold parameters used here would however result in a significantly higher precision in the polyphonic case, but at the cost of a much lower recall and thereby yielding an overall result lower F-measure. The precision and recall are already accounted for through the F-measure, but for future studies, the differential between the two might also be worth accounting for in the optimization criteria, so that each step of the search would yield the set of parameters that optimize both the recall and precision along the given axis, rather than optimizing the mean of the two.