Onset Detection
Final Project Report

Corinne Darche
MUMT501: Digital Signal Processing
April 20, 2022

1. INTRODUCTION

Onset is a music and audio term with an inconsistent definition. In their paper, Bello et al. [1] differentiated the term from the related subjects of attack and transient. Attack refers to the temporal interval in which the amplitude envelope increases. Transient is the short intervals when the signal evolves quickly in an unpredictable way before stabilizing to its steady-state. Onset specifically refers to the instant in which the transient begins. In other words, attack is the most general term for this event, whereas onset is the most specific.

Onset detection rose to prominence with advancements in audio compression techniques. File formats, like MP3, wanted to reduce bit-rate and file sizes without compromising the listener's ability to detect musical events [1]. It is important to not sacrifice such a crucial part of audio when altering it through compression or any other effect application. It should be important to note that onsets can be identified both quantitatively and perceptually by listeners [3]. For the scope of this project, only the quantitatively-defined onsets will be considered.

This report explores two different onset detection strategies. After covering the basic algorithm format, the focus shifts to Scherrer and Depalle's [2] algorithm for detecting exponentially damped sinusoids (EDS) and percussive sounds, which I implemented in a MATLAB script.

2. ONSET DETECTION ALGORITHM DESIGN

Bello et al.'s [1] paper serves as a tutorial and covers the different aspects of a basic onset detection algorithm. It is one of the foundations for onset detection research, and it is imperative to understand the basic mechanics before exploring more complex applications.

First, the audio signal gets preprocessed. This step is optional, but it yields better results as it removes any irrelevant noise from the signal. Bello et al. [1] mention that it can be done by separating the multiple frequency bands or by separating the transient and the steady-state. By splitting the signal into different frequency sub-bands, the global estimates improve, which can be useful in different applications.

After preprocessing, the signal is then reduced and subsampled into detection functions to indicate an onset. There are many options to find these, and Bello et al. [1] cover two types of methods to identify an onset: signal features and probabilistic models. Temporally, onsets are often indicated by an increase in amplitude. Spectral features revealed by the Short-Time Fourier Transform (STFT) and phase deviation are also useful for detecting an onset. As for probabilistic models, the two possible approaches include examining sequential probability or examining any potential 'surprising moments' in the signal. Bello et al. did notice that certain detection functions work better with certain types of sounds. For instance, time-domain functions are useful for percussive sounds and spectral functions are accurate for pitched sounds [1].

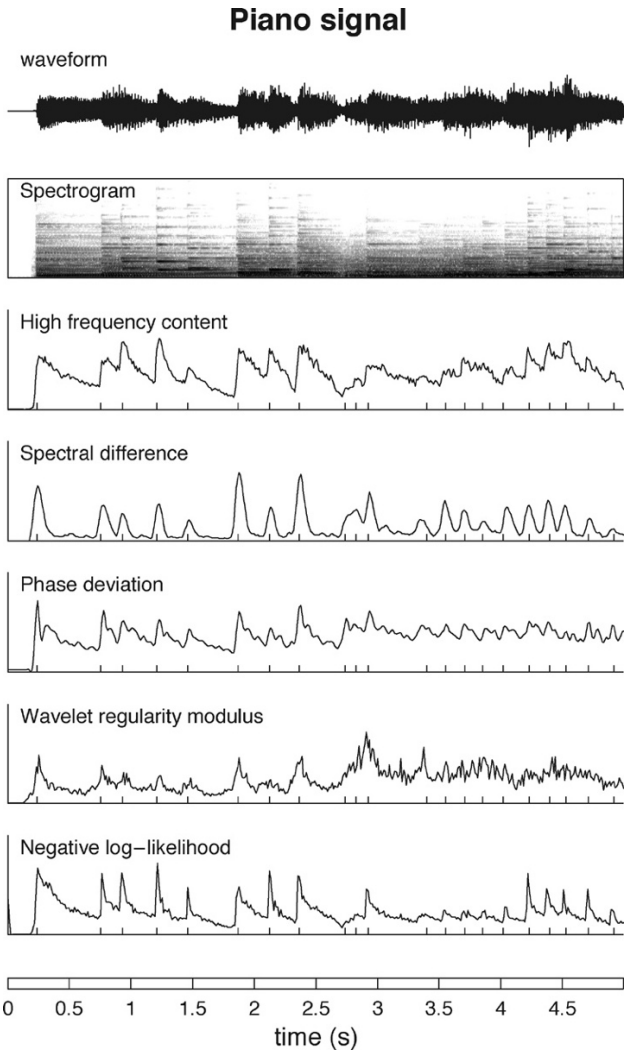

Figure 1: Comparison of different detection functions applied to a five second piano signal, as presented by Bello et al. [1]

These functions are then post-processed through smoothing and thresholding to improve peak-picking results. For their respective experiments, Bello et al. normalized the detection functions by subtracting them by their mean, dividing them by their maximum absolute deviation, and then low-pass filtering them. Peak-picking then selects the local maxima above the defined threshold.

This is how a typical onset detection algorithm is designed. In the next sections, we will explore an example that was based on this outline to detect the onset of pitched percussive sounds.

3. ONSET DETECTION FOR EDS AND PERCUSSIVE SOUNDS

In their 2014 paper, Scherrer and Depalle [2] create an onset detection algorithm to detect the onset of pitched percussive sounds, like a guitar or a marimba. They test their algorithm both

on synthetic signals, created by exponentially damped sinusoids (EDS), and on real pitched percussive sounds.

They used a two-step approach to first roughly estimate the onset location and then refine them using a more specific detection function. The rough onsets are estimated using the STFT with a Hanning analysis window:

$$X[l,b] = \sum_{n=0}^{N-1} w[n]\, x[n + lH]\, e^{\frac{j2\pi nb}{N}} \text{ with } b \in [0; N-1]\,. \quad (1)$$

The STFT is then used to create the frequency-domain detection function, which measures the difference between two back-to-back STFT frames:

$$d_f[l] = \sqrt{\sum_{b=0}^{N/2} (|X[l,b]| - |X[l-1,b]|)^2}\,. \quad (2)$$

The refined stage introduces a time-domain detection function, which makes up for the rough estimator's delay by performing the onset detection a few hop sizes before each rough onset. It is calculated as follows:

$$d_t[n] = \frac{1}{J} \log\left(\frac{\sum_{m=n+1}^{n+J} x^2[m]}{\sum_{l=n-J}^{n-1} x^2[l] + v}\right) \cdot \sum_{k=n+1}^{n+J} x^2[k]. \quad (3)$$

The frequency-domain function and the time-domain function are both post-processed using the same method as described by Bello et al. [1]. The peaks are determined using parabolic interpolation and then pruned through adaptive thresholding. This thresholding process provides a minimum for a peak value at a given integer, based on the median-filtered detection function and a defined absolute threshold. The final set of onsets are then filtered through one last time to remove any repeated peaks, defined as peaks occurring within a given number of samples.

4. IMPLEMENTATION OF THE ONSET DETECTION ALGORITHM

For my implementation, I created a MATLAB script which runs a signal, either synthesized or real, through each described detection function and pruning mechanism to detect the onset. Scherrer and Depalle provide tuned parameters for each variable depending on the type of signal used (Fig. 2, Fig. 3). The defined parameters are the following:

- N: FFT size
- H: Hop size
- J: Number of samples for Eq. 3
- v: Regularization factor for Eq. 3
- γ: Coefficient for the normalized derivative filter at the post-processing stage
- τ: Absolute threshold for the adaptive threshold

Rough onsets	Refined onsets
N : 2048	J : 200
H : 1024	υ : 10^{-4}
γ : 0.3	γ : 0.1
τ : 0.1	τ : 0.5
p : 5	p : 5
ℓ : 0.5	ℓ : 0.5
α : 6dB	α : 6dB
I : 900	I : 900

Figure 2: Parameters used for the synthetic sound testing, as presented by Scherrer and Depalle [2]

Rough onsets	Refined onsets
N : 1024	J : 400
H : 512	υ : 10^{-4}
γ : 0.3	γ : 0.1
τ : 0.15	τ : 0.5
p : 5	p : 5
ℓ : 0.5	ℓ : 0.5
α : 6dB	α : 6dB
I : 2205	I : 900

Figure 3: Parameters used for the real sound testing of a guitar sound, as presented by Scherrer and Depalle [2]

- p: Order for the median filter
- l: Control for how much the median filtered function impacts the absolute threshold
- α: Threshold for peak-finding using parabolic interpolation (in dB)
- I: Time interval used for pruning repeated onsets

The algorithm implementation went smoothly with a few points of confusion. First, the Hanning window length is not specified in the paper's parameters. So, I made the window size equal to the Fast Fourier-Transform (FFT) frame size N. The second difficult aspect was the repeated onset pruning. After some creative programming strategies and some nested while loops, I was able to create a dynamic pruning algorithm that works for both onset detection functions.

The paper does mention an accompanying website where the reader can go download all the sound files used in the experiment and recreate it on their own machine. Unfortunately, the website is no longer in service eight years after this paper's publication. There were details on how the authors created the synthetic sounds, however they are not precise enough to recreate

them accurately. So, I made a simple synthetic sound using an EDS model that follows the paper's basic instructions and tested it in Section 4.1. For real sounds, one of the external websites mentioned still works, so I downloaded the dataset and tested one of the sounds in Section 4.2.

4.1 Synthetic Sounds

I attempted to create a synthetic sound using the paper's description. Scherrer and Depalle created their own synthetic signals that reproduce sounds from a guitar or a piano [2]. I used this formula to create the EDS signals:

$$x_{EDS}[n] = Ae^{-an}\cos(2\pi ft + \phi) \quad (4)$$

where A is the signal's amplitude, a is the damping factor, f is the frequency in Hertz, and ϕ is the phase. Following the paper's instructions, I used five partials, each one consisting of a pair of EDS with slightly different values for A, a, f, and ϕ. In my example sound, I created a C4 piano note with five of its partials. I created two transients with no overlap so that the onsets can be easily verified visually from the graph.

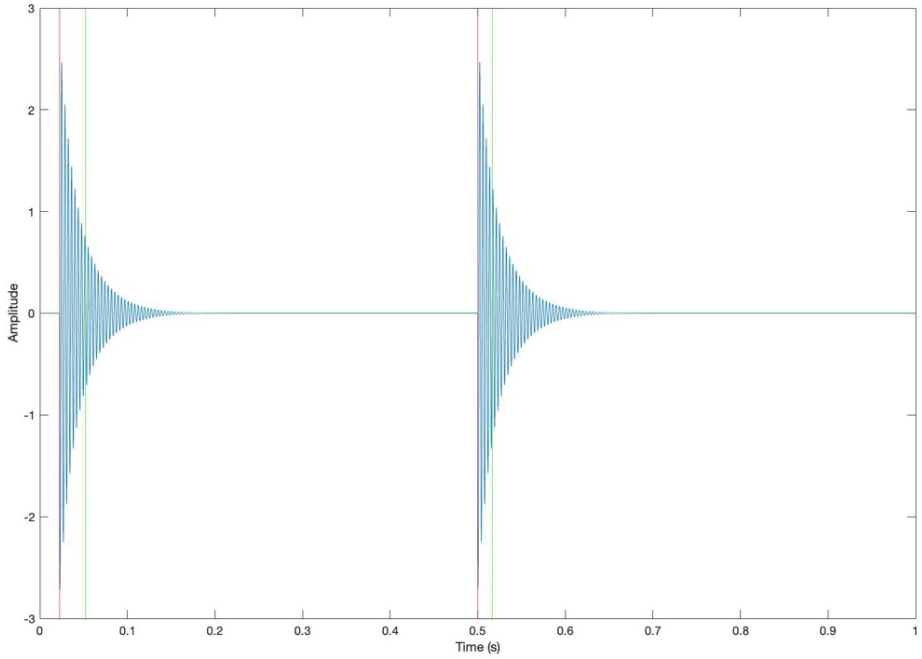

Figure 4: Results from the implementation of the Scherrer and Depalle [2] algorithm using a synthesized EDS-based sound. The rough onset is indicated in green, and the refined onset is indicated in red.

As previously mentioned, the rough onset is consistently late due to the nature of STFT's windowing and hopping. They are close to the actual onset, but this delay can be detrimental, especially in the context of audio compression. This is supported by the results in Fig. 4. In this case, with a very clean signal, the refined onset detection function can correctly identify both onsets.

It should be noted that this is a very simple signal. Scherrer and Depalle's synthetic sounds have significantly more transients and introduce noise to the signal. When testing with different types of signals, they also noticed that the refined onset detection function is significantly more accurate than the rough onset detection function.

4.2 Real Sounds

To test out the real sounds, I downloaded the Leveau dataset for onset detection from ADASP Group, as mentioned in the Scherrer and Depalle paper [2]. I used the "guitar2.wav" file and used the parameters defined in Fig. 3. The paper does not specify what audio file was used when tuning these guitar sound parameters. So, I anticipated some inaccuracy in the results.

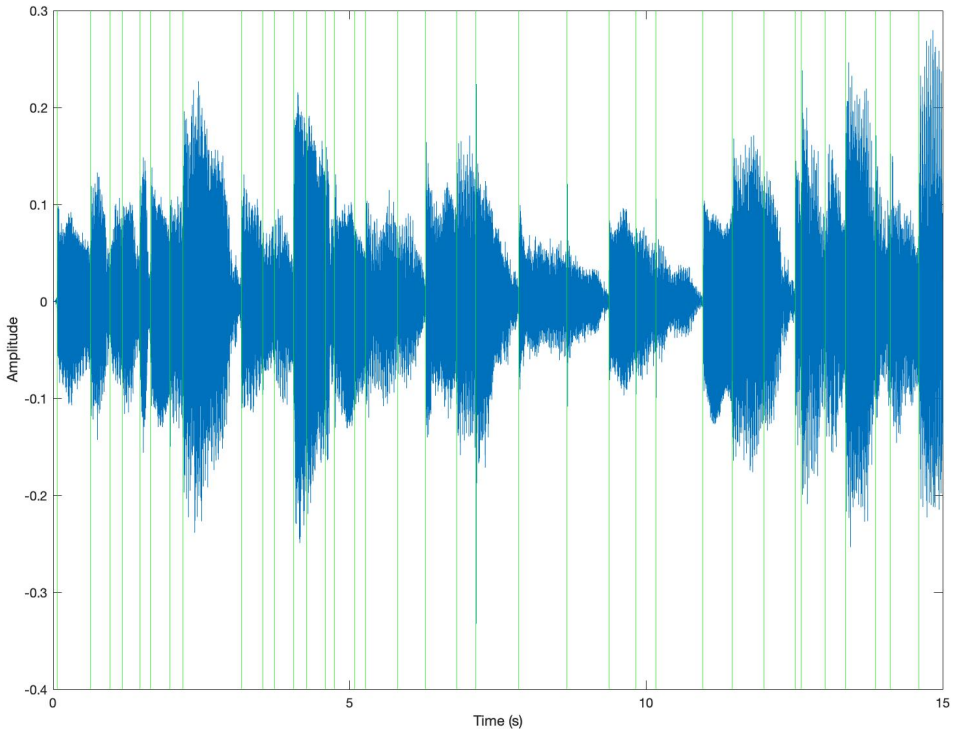

Figure 5: Correct onsets for the Leveau dataset's "guitar2.wav" file

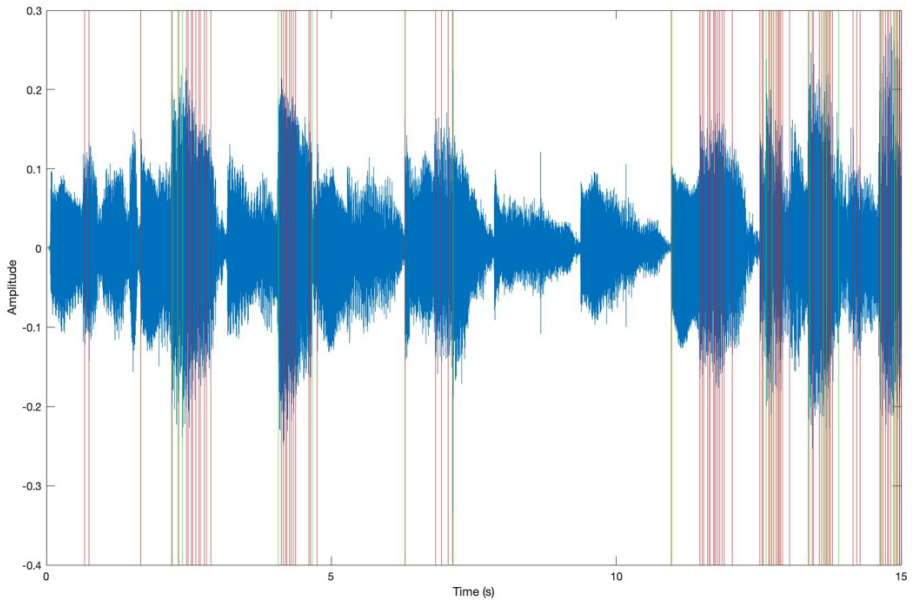

Figure 6: Results from the implementation of the Scherrer and Depalle [2] algorithm using a real guitar sound. The rough onset is indicated in green, and the refined onset is indicated in red.

As seen in Fig. 5 and Fig. 6, my implementation performed poorly when presented with a real sound. While the rough onset detection was also inaccurate and incorrect, it was more selective and correct compared to the refined onset detection, which detected too many onsets. The correct answers provided by the Leveau dataset indicate 36 onsets. The rough and refined onset detection functions detected 21 and 86 onsets respectively.

While these answers raise concern for the algorithm's accuracy, the results from Section 4.1 ensure that both onset detection strategies are reliable when used correctly. In addition, there are some strong onsets in this sound that are correctly identified. Extensive parameter tuning will significantly improve the results.

5. CONCLUSION

The main objective for this project was to learn strategies for onset detection used in digital signal processing. After discussing the basic structure of an onset detection algorithm, we explored a specific example for detecting onsets in pitched percussive sounds. The Scherrer and Depalle [2] paper outlined the algorithm design very well but were vague on the parameters chosen and did not provide the exact sounds used to recreate their experiments. Nevertheless, the algorithm was highly successful with a simple synthetic piano sound and has high potential for real sounds when used with properly-tuned parameters.

References

[1] 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*, vol. 13, no. 5, Sep., pp. 1035-1047, 2005.

[2] B. Scherrer and P. Depalle, "Onset Time Estimation for the Exponentially Damped Sinusoids Analysis of Percussive Sounds," In Proc. of the 17th International Conference on Digital Audio Effects, 2014, pp. 1-7.

[3] S. Dixon, "Onset Detection Revisited," In Proc. of the 9th International Conference on Digital Audio Effects 2006, pp. 1-6.