SHEFFIELD HALLAM UNIVERSITY

FACULTY OF ARTS, COMPUTING, ENGINEERING AND SCIENCES

On Musical Onset Detection via the S-Transform

Submitted by:

Nishal SILVA

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science

in

Telecommunication and Electronic Engineering

November 23, 2017

Abstract

A musical onset is a location in a piece of music which carries meaningful information about spectral transitions - i.e., when a musical note is played, the start of the note is considered as an onset. Musical onset detection is vital in many applications. Some of which are content delivery, compression and beat tracking. An accurate onset detection method is required as it is a trivial step in many applications.

There are several methods designed to detects onsets in music. These methods can successfully detect musical onsets in genres such as dance, pop and rock music as the transients are sharp and defined. However, genres such as classical music, opera music, and some soft pop music do not yield good results for existing onset detection as they lack sharp transients as opposed to earlier mentioned genres.

This thesis will propose a method which enables musical onsets to be detected in music genres where sharp beats are absent. The proposed method retains several steps which are common to many onset detection methods while introducing two modifications which are;

- The introduction of the S-transform in place of the short time Fourier transform (STFT).
- Splitting the s-transform into frequency bands and computing local averages.

The *beat* may be expressed as an onset envelope which is periodic, provided that the tempo is constant. By this premise, the s-transform matrix is split into relatively narrow frequency bands and each band is checked for a an acceptable onset envelope by means of thresholding the mean through time for each frequency band. The onset envelope with the highest periodicity is selected as the onset envelope for the music piece.

Results have shown that beat causing onsets generally occur in a single frequency band. For genres where sharp transients are absent, onsets may be gracefully localized through the proposed method. Results have verified that musical onsets present in static frequency bands can be successfully identified for genres such as classical, opera and instrumental music.

Acknowledgements

My gratitude to the Lord above.

My sincerest gratitude to my parents, the two people that fed me, clothed me, looked after me, gave me a good education, and paved my path towards undergraduate and postgraduate studies.

I would like to thank my supervisor, Dr. Chathuranga Weeraddana for guiding and steering me towards the completion of my masters research, for the knowledge bestowed, and for the analytical thinking ability I have gathered through the work done.

I would like to thank my sister, Dr. Nishala Silva, for the guidance and support given to me in all aspects of my life.

My sincerest thanks to Mr. Mavidu Iddagoda, who gave me an idea which led me to work on this research.

Contents

List of Figures

List of Tables

"Why is it that when one man builds a wall,
the next man immediately needs to know what's on the other side?"
Tyrion Lannister

1 Introduction

“The act of tapping one’s foot or nodding one’s head in time to music is an intuitive and often unconscious human response” [14]. The process referred to as *beat tracking*, is the computational equivalent to this human behavior. Beat tracking is performed to recover a sequence of beat causing onsets which are consistent with a human tapping his foot.

Even the simplest piece of music rely on transitions as musical timbre and tone color evolves. It can be said that without transitions, there cannot be any musical meaning [6]. These transitions are often due to changing notes played though a musical instrument or sung vocally.

The automatic detection of transitions in a piece of music are essential for many modern applications. Some of these applications include;

- Content delivery
- Compression
- Indexing and retrieval [1][6].

These detections have given rise to many audio editing algorithms and digital effects (time stretching, pitch shifting, equalization etc.).

A musical onset is a representation of a musical note or other sound. As each beat will fall on an Onset, musical onset Detection is an integral and primary segment of any beat tracking system. To continue this thesis, the concept of a musical onset needs to be defined. There are three main terms - musical onset, transient, and attack, which are similar in nature and may be used interchangeably [6].

Attack, Transient, Onset

A clear distinction must be made between the concepts of a Transient, Onset and Attack. These are very similar and may be interchangeably used with the right context. That is the term *Onset* or *transient* may be used to define the

[1] Refer 4.1 for definitions of musical terms

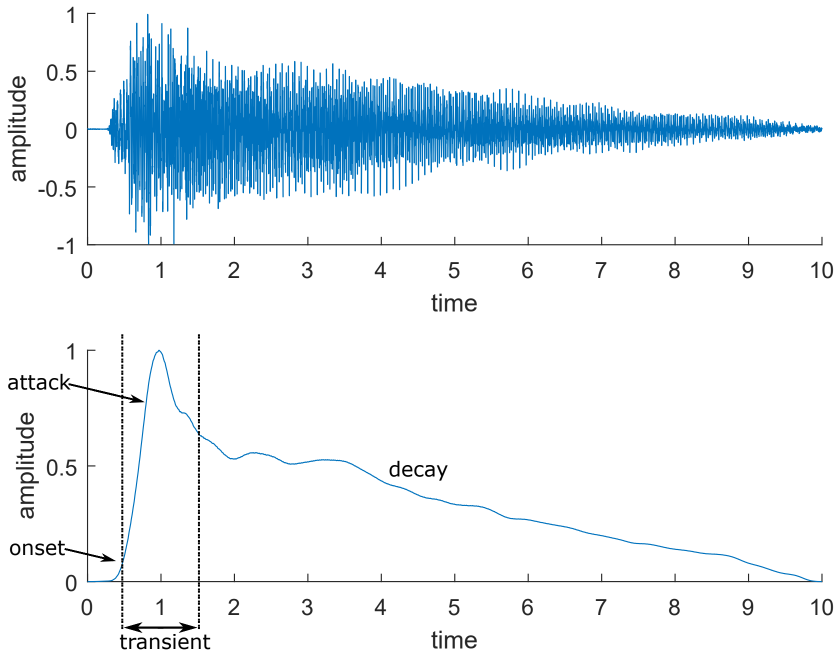

FIGURE 1.1: Attack, Transient, Onset, and Decay in the case of an isolated musical note

beginning of a musical note. Figure 1.1 shows these notions in the case of an isolated musical note.

- An *attack* can be defined as the time interval through which the amplitude envelope increases.
- A *transient* may be defined as the duration through which the signal undergoes a rapid and unpredictable evolution. The transient usually begins with an excitation (a hammer strike on a piano or xylophone), and continues till it is damped, which leads to the decaying of resonant frequencies [6].
- An *onset* of the note is an single instant used to represent the transient. The most common representation of a musical onset is at the start of the transient.

1.1 Motivation

The ability to detect musical onsets accurately, is a fundamental criterion in any beat tracking algorithm. As all beats lie on onset locations, properly identified musical onsets will lead to an accurate beat tracking and estimation.

A limitation to many existing onset detection algorithms is the inability to detect musical onsets in some genres of music where there are no clearly distinguishable transients. Many existing algorithms fail to identify opera and classical music genres while having excellent performances in identifying pop, rock, and dance music where prominent kick drum driven beats are present [30][13][14][40].

Musicians, music producers, and music production softwares usually have to deal with a myriad of genres, and cannot restrict themselves to a few where onset detection methods do work. Hence it a critical to develop a method to identify musical onsets accurately for music of any genre.

1.2 Challenges

Many of the existing methods excel in identifying musical onsets and subsequent beat locations in music genres such as pop, rock, and dance, where clear and abrupt transients are present, but the results may lose its accuracy when the music deviates away from standard kick-drum driven music. Some very common examples for this phenomenon are classical and ensemble music, opera music, and legato playing [30][13][14][40].

Even the most simple piece of music can be extremely complex in structure due to the fact that there are multiple musical instruments and vocalizations present. These instrumentations and vocalizations result in strong amplitude modulations in the resultant piece of music which may mask off beat causing onsets, rendering the detection functions inaccurate.

In most studio and live applications it is desirable to identify the *tempo* of a song. A recording engineer may need the tempo to add certain effects, Musicians will need the tempo to play along or to overdub and DJ's will need the tempo to beat-match and perform seamless transitions between songs.

In the case of a real scenario: If a DJ is to beat match two songs, and the tempo of one song is not identified accurately, the resultant transition between the songs will be jagged and unpleasant. In such cases, an accurate beat tracking is required and hence, an accurate musical Onset detection is necessary.

The general cases where a musical onset detection functions may fail may be broadly categorized into two;

- When the rhythm of the music is less pronounced,
- During rapid tempo changes [30].

1.3 Objectives and Contributions

As discussed in the previous section, there are two main cases where musical onset detection methods may fail to render accurate results. the objective of the work presented through the remainder of this thesis aims to develop an algorithm that is able to identify beat locations in music when the rhythm of the music is less pronounced. This is mainly due to string amplitude modulations in the signal and is very common in genres such as classical music, opera music, and soft pop music [30].

The objectives and contributions made by this thesis are as follows;

- To study the current musical onset detection methods and their accuracy, with a selection of songs of a broad range of genres.
- To introduce a novel method for musical onset detection where a number of significant disadvantages of currently existing methods are overcome.
- To show a comparison between existing methods, the proposed method, and the improvement in detected musical onsets for some songs.
- To explore various methods of thresholding applied to detected musical onsets to identify accurate beat locations.

1.4 Thesis Outline

The work presented by this thesis presents a new musical onset detection method, which introduces two modifications while preserving several steps common to most onset detection methods. chapter 2 discusses some of the current work done on musical onset detection along with their strengths and possible shortcomings. Chapter 3 presents a brief description on the digital signal processing (DSP) techniques that are used in the task of onset detection.

Some musical theory is required to further understand onsets and their importance from a musical point of view - i.e., the *tempo* of a piece is directly related to its beat causing onsets. A brief explanation in musical theory which is applicable for the task of onset detection, and definitions for musical terminologies used in this thesis is presented in chapter 4.

Chapters 5 and 6 present the proposed method musical onset detection method introduced in this thesis. Chapter 5, presents the methodology, whereas chapter 6 presents the results, along with a comparison of the proposed method with existing methods.

Chapter 8 shows the results of the proposed method. Plots for the s-transform split into frequency bands, and the thresholded frequency band signals are plotted in section 8.

2 Literature Review

This chapter will serve as a review of the recent work done on musical onset detection. Among various methods intended for the task of detecting musical onsets, some rely on temporal features of the audio signal [30], while others utilize its time-spectral features [14][20][13]. Most of the onset detection methods are a pre-processing step in beat tracking systems. In such cases, only the onset detection stage is discussed.

This chapter discusses several popular onset detection methods. The first to be discussed are the temporal analysis methods, followed by a discussion of the spectral analysis methods. Each author has used various nomenclatures for their respective methods and those names have been used as the title for the description of the said method. This chapter will follow the same notations and nomenclatures as the original reference material for clarity and to better convey the point made by the original authors.

2.1 Relative Difference of Filter Banks

Klapuri suggests a method where a set of first order difference functions are calculated following the filtration through a filterbank.

The filterbank consists of 21 filters, which covers the frequencies from $44Hz$ to $18kHz$. The filters consist of three one ocatave band-pass filters and eighteen third-octave band-pass filters. The output of each filter is decimated by a factor of 180 and their amplitude envelopes are calculated by a convolution between the band limited signal and a $100ms$ half-Hanning window. The window masks rapid modulations while preserving sudden changes, similar to the energy integration of the human auditory system [43][50].

The first order difference function of each amplitude envelope $A(t)$ is calculated which is divided by the amplitude envelope function to obtain a first order *relative difference function* $W(t)$ (equation 2.1). It is found that the relative difference function thus calculated is equivalent to the differentiation of the logarithm of the amplitude envelope.

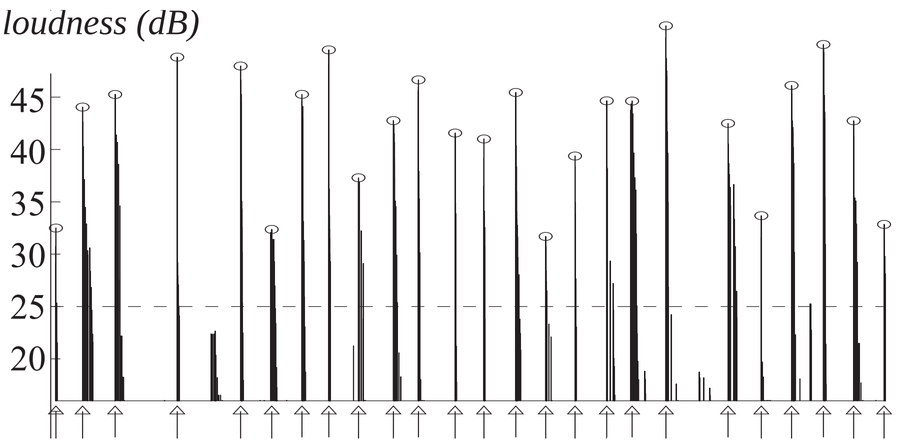

FIGURE 2.1: The detected onsets as a function of their time [30]

$$W(t) = \frac{d}{dt} log((A(t))). \tag{2.1}$$

The intensity of each *onset component* is calculated as the maxima of $W(t)$. Components which are less than 50 ms apart are dropped and the resultant onset components of each filterbank are summed. Figure 2.1 shows a set of detected onsets. The genuine onsets can be easily seperated using a global threshold.

Klapuri explains that the results of the system for symphony orchestra performances have been very poor. It is stated that strong amplitude modulation in middle frequencies may confuse the system and a primary shortcoming in this method is it's inability to deal with strong amplitude modulations present in classical music and certain instrumental sounds [30].

2.2 Frequency Analysis and Envelope Extraction

Scheirer explains in his paper that envelopes extracted from a short number of broad frequency channels are sufficient to analyze a musical signal. The musical onset detection stage of the algorithm as proposed by Scheirer employs a filterbank based detection.

The fiterbank consists of six filters. Each having a sharp cutoff frequency and covering roughly one octave. The lowest is a low pass filter with a cutoff at $200Hz$. The next four filters are band pass filters, with cutoffs at $200Hz - 400Hz$, $400Hz - 800Hz$, $800Hz - 1600Hz$ and $1600Hz - 3200Hz$. The highest is a high pass filter, with a cutoff at $3200Hz$. Each filter is implemented using a sixth order elliptic filter with $3dB$ of ripple in the passband and $40dB$ of rejection in the stopband.

The envelope is extracted from each band of the filtered signal and the first order difference is calculated along the time axis. The difference signal is examined for periodic modulation. The derivative of an envelope function serves as a type of onset filter [43]. The derivative functions of the envelope signals are passed on to later stages of the algorithm.

2.3 Energy Flux

Beats tend to occur at salient features of an audio signal such as onsets, note changes, and percussion hits. Laroche explains a method to locate fast variations in the frequency domain which correspond to the above mentioned salient features. This is preferred over the temporal energy as onsets may be hidden by continuous tones of higher amplitude.

A time-frequency representation of the audio signal is obtained using the STFT. Laroche defines the STFT at the normalized frequency f and frame i, when the signal is $x(n)$, the frame time in seconds is t_i, the sampling frequency is F_s, the size of the analysis window in samples is N and the analysis window is $h(n)$ as;

$$X(f, t_i) = \sum_{n=0}^{N-1} h(n)x(n + F_s t_i)e^{2j\pi f n}. \tag{2.2}$$

The window being used is of width $10ms$ and there is zero overlap between two successive windows. A compression function $G(x)$ is applied to each FFT bin, to avoid high frequency components being masked off by higher amplitude low frequency components. Laroche gives two possible compression functions $G(x)$;

$$G(x) = x^{\frac{1}{2}}, and \tag{2.3}$$

$$G(x) = \arcsin x. \tag{2.4}$$

A first order difference is calculated for all frequency bins and the sum of all first order difference functions is calculated to obtain $E\hat{(i)}$, which is given by equation 2.5. The representation when $E\hat{(i)}$ is half wave rectified exhibits sharp maxima at onset locations and is better suited for the analysis of onset components. $\hat{E}(i)$ is half-wave rectified to obtain a *positive energy flux* $E(i)$, which is shown by equation 2.6.

$$\hat{E}(i) = \sum_{f=f_{min}}^{f_{max}} G(|X(f,t_i)|) - G(|X(f,t_{i-1})|). \tag{2.5}$$

$$E(i) = \begin{cases} \hat{E}(i) & \text{if } \hat{E}(i) > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

The energy flux E, represents the onsets of the signal. This signal is used int the beat tracking section of the algorithm to identify the tempo.

The *energy flux* signal presented by Laroche is said to contain just enough information to detect the beat, but not sufficient for applications requiring a more comprehensive analysis. The energy flux based onset detection yields good results for rhythmic contemporary music such as rock, pop, techno, and dance, but it is less successful when the rhythm is less pronounced. The authors claim that the energy flux based tempo analysis system can be applied in real time with limited access to the future of the signal [34].

2.4 Spectral Energy Flux

As onsets can be easily masked by continuous tones of higher amplitude [34], there is a higher likelihood of detecting them after the seperation into frequency channels [15][34][35][24][30][32]. Alonso et al. proposes to follow a frequency domain approach as it has been proven to outperform its time-domain counterpart.

The input audio signal is analyzed using a Short Time Fourier Transform (STFT), which leads to;

$$\bar{X}(f,m) = \sum_{n=0}^{N-1} w(n)x(n+mM)e^{-j2\pi fn}, \tag{2.7}$$

where $x(n)$ denotes the audio signal, $w(n)$ is the analysis window, M is the advancement between successive windows, m is the frame index, and f is the frequency.

Alonso defines the *spectral energy flux* $E(f,k)$, as an approximation to the derivative of the frequency content with respect to time, where $G(f,m)$ is obtained via low pass filtering with a half-Hanning window, which is followed by a logarithmic compression function [32].

$$E(f,k) = \sum_{m} h(m-k)G(f,m). \tag{2.8}$$

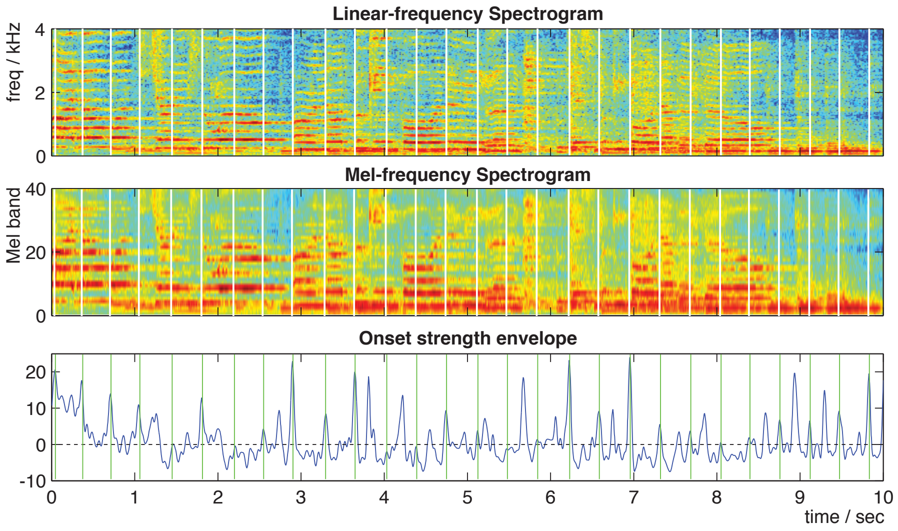

FIGURE 2.2: STFT spectrogram (top), Mel-spectrogram (middle, and the onset strength envelope (bottom) [20]

Alonso et al. proposes $h(m)$ to be a FIR filter differentiator designed by a Remez optimization procedure. This approach highly improves the extraction of meaningful features when compared with the first order difference [32][34][35].

The positive contributions of the *spectral energy flux* along the time axis are summed which produces a temporal waveform having sharp maxima at transients and note onsets. The true beats are filtered using a dynamic threshold - all peaks above the threshold are preserved, while those which are lower are discarded [1].

The method presented by Alonso et al. claims to have a relatively small computational strain on the processor. The method fails in a real time implementation as it would need access to future signal components for a successful onset detection. The results presented in the paper shows that the method shows a good performance for music with straightforward rhythm, but has difficulty in detecting onsets in genres such as classical music.

2.5 Onset Strength Envelope

The onset envelope is calculated using a similar method as many other models [24][30][29]. The magnitude STFT of the sound file is calculated using a window size of 32*ms*, and a hop size of 4*ms*. This is converted to an approximate auditory representation by mapping to 40 Mel bands [19][46]. The auditory frequency scale is used to balance the perceptual importance of each frequency band. The Mel spectrogram is then converted to dB, and the first order differentiation is calculated along time in each frequency band. Each different equation is then half-wafe rectified. The positive differences are summed across all frequency bands to obtain the *onset strength envelope* [20].

Figure 2.2 shows a comparison between the STFT and the Mel scaled spectrogram, as well as the onset strength envelope, which may be obtained by summing each onset strength difference function across each frequency band.

The method shares the same limitations as the others where onset locations in music genres such as classical music, opera music, and soft pop music may not be identified correctly whereas songs with distinct kick-drum driven beats are identified accurately.

2.6 Median Onset Aggregation

The method presented by McFee in his paper is a modification of the work done by Ellis [20]. The method proposed by Ellis can be broken into 3 stages;

1. Onset strength envelope $w(t)$ is computed.
2. Tempo estimation by peak-picking.
3. Selecting beats consistent with the estimated tempo.

The modification as proposed by McFee in [39], is to preserve steps 2 and 3 as they are, and to introduce a *median onset aggregation* in place of the *summation across frequencies* in step 1.

The method proposed by Ellis [20] uses the sum across thresholded log-magnitude difference frequency bands where S denotes the Mel scaled magnitude spectrogram;

$$w_s(t) = \sum_f \max\left(0, \log S_{f,t} - \log S_{f,t-1}\right). \tag{2.9}$$

The drawback in this method as pointed out by McFee is that $w(s)$ may respond to either a large fluctuation confined to a single frequency band, or many small fluctuations spread across multiple frequency bands. The latter case may arise from percussive events or multiple synchronized onset events, which coincide to an onset location. But the former case may arise only if a single source plays out of sync with other sources. To better capture synchronous onset events, McFee proposes a median operator;

$$w_s(t) = \operatorname*{median}_f \max\left(0, \log S_{f,t} - \log S_{f,t-1}\right). \tag{2.10}$$

The paper explains that the median onset aggregation method does not yield improvements for musical pieces consisting of instrumental sounds. This is true for other cases where the amplitude modulations are strong and can mask off musical onset causing components.

2.7 Frequency Analysis

Goto and Muraoka proposes a *frequency analysis* to obtain onset locations in [24] where the full frequency band is split into several frequency ranges, and an *onset-time vector* is calculated, where each dimension of the vector correspond to a different frequency range. This representation enables for the easy consideration of onset times in all frequency ranges simultaniously.

The frequency spectrum for each frequency range is calculated using a fast Fourier transform (FFT) using a Hanning window. The FFT is calculated with a window size of 1024 samples, and a hop size of 256 samples (75% overlap).

The system proposed by Goto utilizes seven frequency bands for which the onset components are found. The frequency bands are in the ranges $0 - 125Hz$, $125Hz - 250Hz$, $250Hz - 500Hz$, $500Hz - 1kHz$, $1kHz - 2kHz$, $2kHz - 6kHz$ and $6kHz - 11kHz$.

Each onset time is found by peak picking $D(t)$ along the time axis, where $d(f,t)$ is the degree of onset at frequency f and time t. Limiting the range of frequencies for the summation of $D(t)$ makes it possible to find onset times in different frequency ranges.

$$D(t) = \sum_f d(t,f). \tag{2.11}$$

The onset times for each frequency range is used to build a vector consisting on onset times for all frequency ranges. This vector is passed on to later stages of the algorithm [24].

The paper by Goto and Muraoka concludes that it is difficult to track music without drum sounds, as they have fewer sounds which fall on the beat. The method can be applied in real time and has been used to display a computer graphics dancer whose motion changes with musical beats. A significant disadvange shared by many onset detection applications is the genre-specific performance - i.e., the performance is satisfactory for music genres with strong beat sounds, while having significantly lesser performance for genres lacking strong sounds at beat locations.

TABLE 2.1: Comparison of several onset detection methods

Music Piece	Manually detected tempo	Onset Strength Envelope [20]	Spectral Energy Flux [1]	Energy Flux [34]	Median Onset Aggregation [39]
example 1	129	129.3	123.3	129	129.3
example 2	120	123.3	136.1	109.7	131.2
example 3	95	127.3	126	126.8	130.2
example 4	132	131.5	129.1	129.8	130.4
example 5	126	126	126.6	126.5	126
example 6	130	127.1	128.9	127.3	127.1
example 7	127	126	129.3	126.5	126
example 8	114	121.2	97.5	98.7	123.2

2.8 Critical Analysis

A common issue faced by many of the methods is the inability to detect beats in music where onsets may be masked by higher amplitude components which may reside in neighboring frequencies. For an example; if a beat causing onset resides at a frequency f and a signal component which is of a higher amplitude resides at frequency Δf where Δ is small, it will cause the onset component at f to be masked off by that at Δf.

To perform a comparison between the methods, a tempo approximation was implemented using the said methods. The onset detection mechanism for each algorithm was used to detect onsets, and an autocorrelation based tempo deduction was performed as presented in [20] to detect the tempo value in beats per minute (bpm).

2.8.1 Autocorrelation based Tempo Deduction

The onset envelope is extracted by each method and it it correlated with itself until a duration of $4s$. The autocorrelation yields the locations for which the onset envelope is most similar to a shifted version ot itself. As beat causing onsets are periodic, this approximation yields the best beat locations.

Knowing the sampling frequency and the number of samples in the signal, the time interval where successful beat correlations occur can be found. This time interval is used to deduce the tempo value in bpm. Table 2.1 shows the bpm values derived for the test dataset defined in 6.1 using several methods explained prior in this chapter.

It can be clearly seen that the tempo values for examples 2, 3 and 8, which are classical and opera music pieces have not been successfully identified using any of the said methods. The performance of methods [20], [1], [34] and [39] are satisfactory for the other music pieces in the dataset. These music are of dance, rock and pop music genres.

3 Technical Background

This chapter will establish several technical elements and theories in the digital signal processing realm, which have been used in the simulation of existing methods and in the proposed method. A separate chapter 4 is dedicated to technical elements and theories related to music.

There have been attempts to identify locations of musical onset both in the temporal and the spectral domain. The temporal methods rely on the envelope of the audio signal as a whole, or on a band-limited segment of the signal.

3.1 Envelope Extraction

Klapuri [30] and Scheirer [43] has proposed methods to analyze the audio signal in the time domain itself. Temporal onset detection methods work by finding the envelope of an input audio wave. The envelope is a smooth curve which outlines the extremities of a given wave [8]. Both Klapuri and Scheirer has obtained the amplitude envelope via convolution with a Hanning window. Figure 3.1 shows a section of an audio signal (top), its rectified version (upper middle), a Hanning window (lower middle), and the result of the convolution of the rectified audio signal and the Hanning window.

The envelope thus obtained is used to detect onset components, which are represented by peaks in the envelope. In the case of musical genres with loud beats, the peaks in the audio signal envelope directly correspond to its beat locations.

3.2 Discrete Fourier Transform

The use of computers has replaced the continuum of values in a signal by a discrete set [23]. When the signal being analyzed is discrete, the continuous Fourier transform cannot be applied. Hence the discrete Fourier Transform (DFT) has been introduced which replaces the integral operation in the continuous Fourier transform with a summation [23].

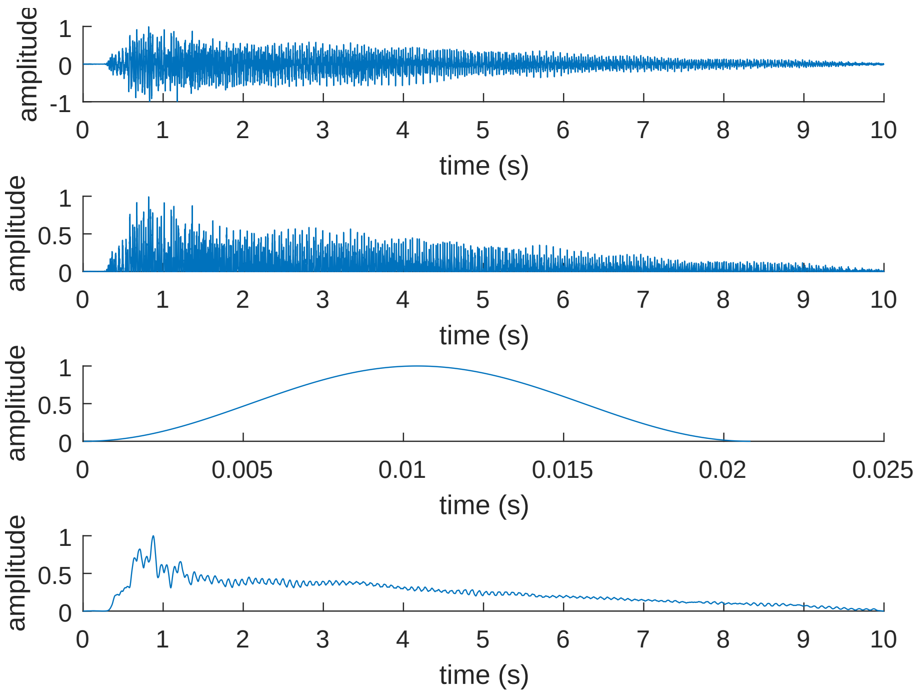

FIGURE 3.1: Audio signal segment (top), its rectified version (upper middle), Hanning window of length 20 ms (lower middle), and the convolution output (bottom)

Consider a discrete series $x(n)$. The discrete Fourier transform for $x(n)$ can be written as;

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{\frac{-2\pi jnk}{N}}, \tag{3.1}$$

where $0 \leq k \leq N-1$, and the DFT of $x(n)$ is calculated in the interval $[0, N-1]$. $X(k)$ is complex in nature. The complex norm of $X(k)$, $|X(k)|$, is called the discrete magnitude spectrum and the complex phase, $arg(X(k))$, is called the discrete phase spectrum of $x(n)$ for $0 \leq k \leq N$.

Consider equation 3.2, a signal consisting of a summation of five sinusoids;

$$y(n) = \cos 2\pi n + \cos 4\pi n + \cos 6\pi n + \cos 8\pi n + \cos 10\pi n. \tag{3.2}$$

Figure 3.2 shows $y(n)$ plotted as a function of time (top), and its magnitude spectrum (bottom) obtained using the DFT. The five frequency components which contribute to equation 3.2 can be easily observed using the plotted DFT.

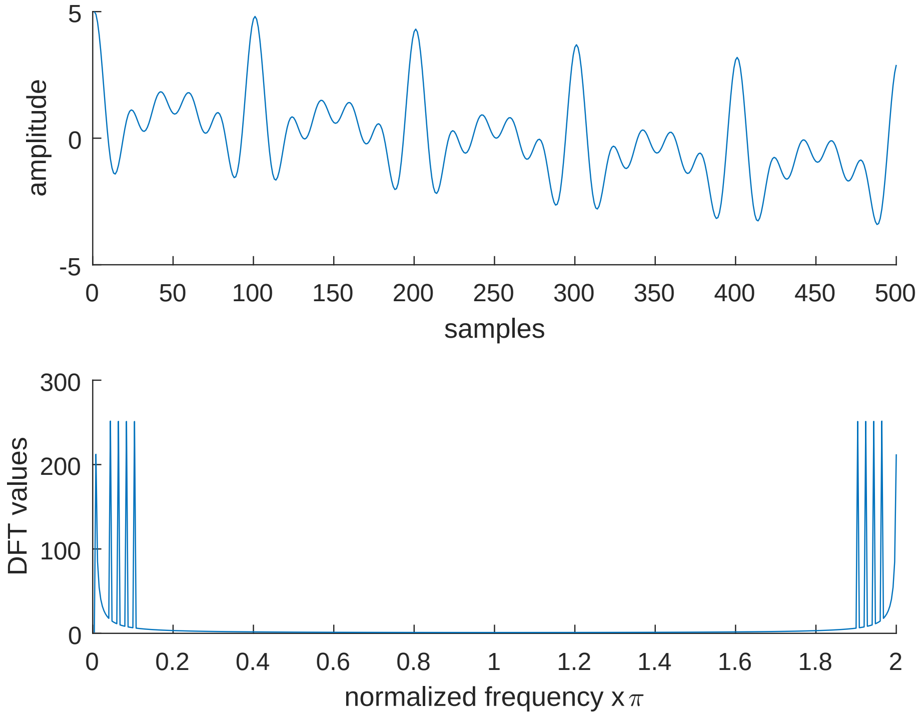

FIGURE 3.2: Signal $y(n')$ and its magnitude spectrum

3.3 Fast Fourier Transform

Consider the computation of $X(k)$ in equation 3.1. To calculate $X(k)$ in the interval $0 \leq k \leq N-1$, it would require N complex multiplications and $N-1$ sums. It would require N^2 complex multiplications and $N^2 - N$ complex summations to compute all N of $X(k)$.

Complex arithmetic is computed as floating point operations on pairs of floating point values in digital computers. Hence, each complex multiplication would require four floating point multiplications and four floating point additions; whereas each addition will require a further two floating point additions. The total floating point computation count for a N-point DFT would be $4N^2$ multiplications and $2(N^2 - N) + 2N^2$ additions.

The number of floating point computations becomes a serious factor affecting the speed of the DFT as N increases. As the number of said computations are proportional to N^2, it can be deemed that the DFT is an order-N^2, or $O(N^2)$ operation [33].

The fast Fourier transform (FFT) is an optimized method of computing the DFT with greater speed. The sped up performance has allowed it to be used in most beat tracking and onset detection methods for the computation of the DFT.

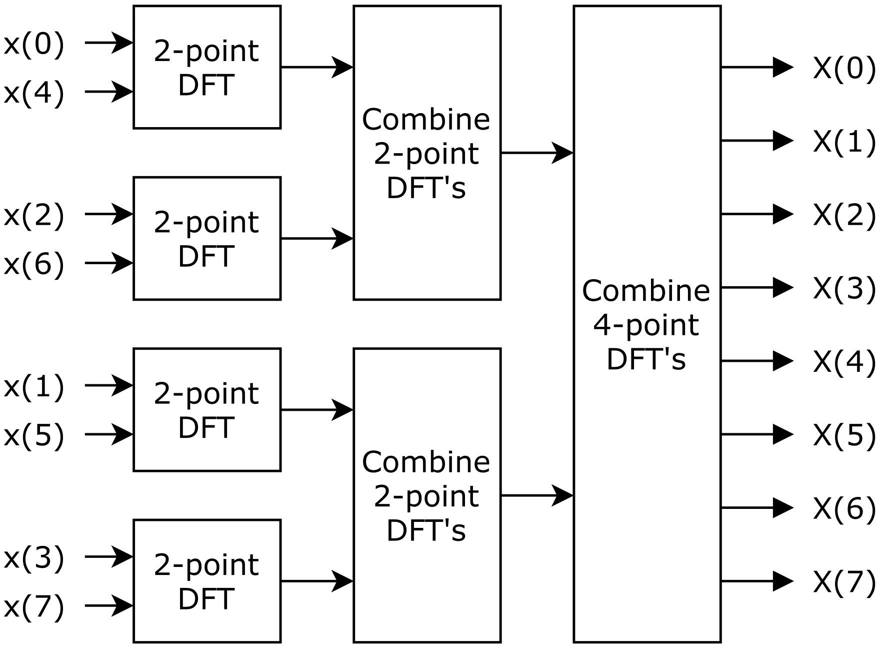

FIGURE 3.3: Three stages in the computation of an $N = 8$-point DFT

The FFT will do a rapid computation of the DFT by factorizing the DFT matrix into a product of sparse factors [51]. Hence the FFT will reduce the complexity of computing the DFT from $O(N^2)$, which may arise if the DFT is applied by itself, to $O(N \log N)$, where N is the data size.

There are two fundamental methods of computing the FFT, which will be briefly discussed.

- Decimation-in-time.
- Decimation-in-frequency [33].

3.3.1 Decimation-in-Time

Decimation-in-time FFT algorithms will reduce the DFT into a succession of smaller DFT analysis equations. The N-point DFT will resolve into two $\frac{N}{2}$-point equations, each of which resolves into two $\frac{N}{4}$-point DFT's and so on. Figure 3.3 shows the first three stages of a decimation-in-time operation for a 8-point FFT [33].

3.3.2 Decimation-in-Frequency

Figure 3.3 shows a decimation-in-time FFT, where $x(n)$ is pre-ordered to obtain $X(k)$ in the correct order. When computing a decimation-in-frequency, $x(k)$ remains in the correct order but the FFT $X(k)$ appears in an incorrect order.

This *disorder* in the indices is predictable. If an 8-point FFT is considered and the indices are expressed using 3 bits, the indices in the correct order are 0, 1, 2, 3, 4, 5, ... and if a 3-bit binary representation is used, the indices are 000, 001, 010, 011, 100, 011,...

The disordered bits are 0, 4, 2, 6, 1, 5, ... and its 3-bit binary representation is 000, 100, 010, 110, 001, 101, ... It can be observed that the disordered sequence can be obtained by *string-flipping* the bits of the ordered sequence [4].

3.4 Short Time Fourier Transform

A signal which has varying frequency components with time (figure 3.5) cannot be successfully recovered with the DFT as the time where each frequency component is present cannot be identified. This is a primary requirement in analyzing audio files. Hence a representation is needed to identify varying frequencies in a signal with relation to their time.

The short time Fourier transform (STFT) is a time-frequency representation of a distribution. The function $x(t)$ is multiplied by a window function $g(t)$, which is nonzero for a short duration. The Fourier transform of the windowed segment of the signal is calculated as the window slides along the time axis. The STFT for a continuous distribution may be expressed as follows;

$$STFT_x(t', f) = \int_{-\infty}^{\infty} x(t)g(t - t')e^{-j2\pi ft}dt. \tag{3.3}$$

Where $x(t)$ is a function of time t, while its short time Fourier transform is a function of both time t', and frequency f.

The transformation $x(t) \rightarrow STFT_x(t', f)$ is linear and depends on the window function $g(t - t')$ [25]. The most commonly used window functions are Hanning or Gaussian windows (refer figure 3.4) as they tend to taper toward the edges, the centered values of the windowed signal have a greater emphasis on the DFT.

The STFT for a discrete signal $x(n)$ can be written as follows, where the integral from equation 3.3 is replaced by a sum;

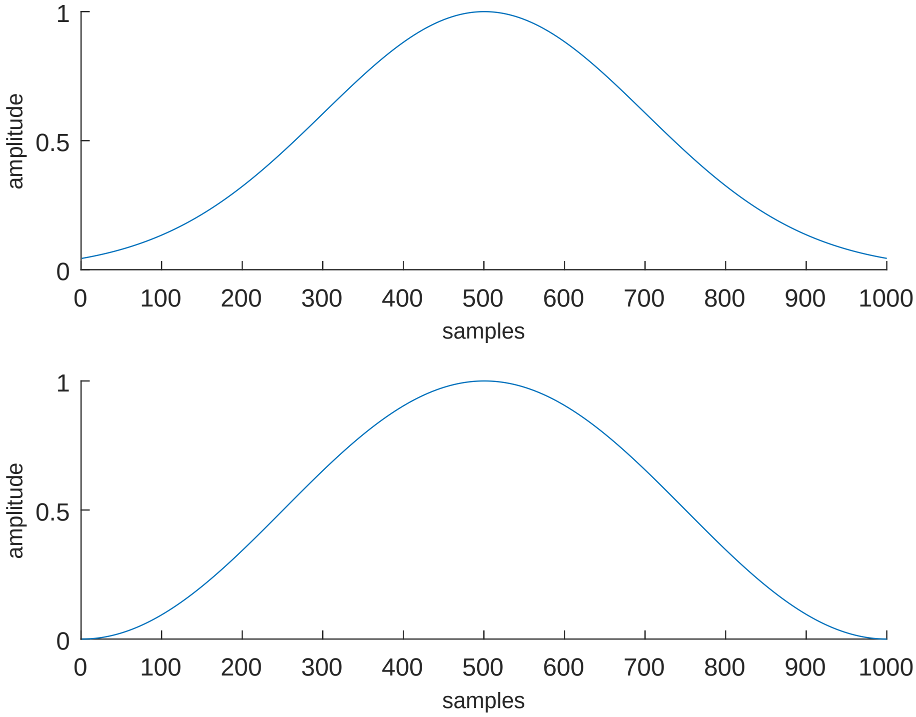

FIGURE 3.4: Gaussian window(top), and Hanning window (bottom)

$$STFT_x(n', f) = \sum_{-\infty}^{\infty} x(n)g(n-n')e^{-j2\pi fn}. \tag{3.4}$$

A compound sinusoid is used to evaluate the STFT effect variation with the window size. The sinusoid $x(n)$ is created where $1 \leq n \leq 1000$. Each interval contains only one frequency($n(1:200) - 10Hz$, $n(201:400) - 100Hz$, $n(401:600) - 200Hz$, $n(601:800) - 300Hz$, $n(801:1000) - 400Hz$). Figure 3.5 shows the compound waveform thus created;

Figure 3.6 shows the STFT plots for the waveform shown in figure 3.5 with varying window sizes. The STFT is plotted with the number of samples in the x-axis, and the frequency in Hz in the y-axis. It can clearly be seen that shorter window lengths lead to poorer frequency resolution. When the window size is larger, the time localization will not be as accurate as when a smaller window is used, but there will be a better frequency resolution.

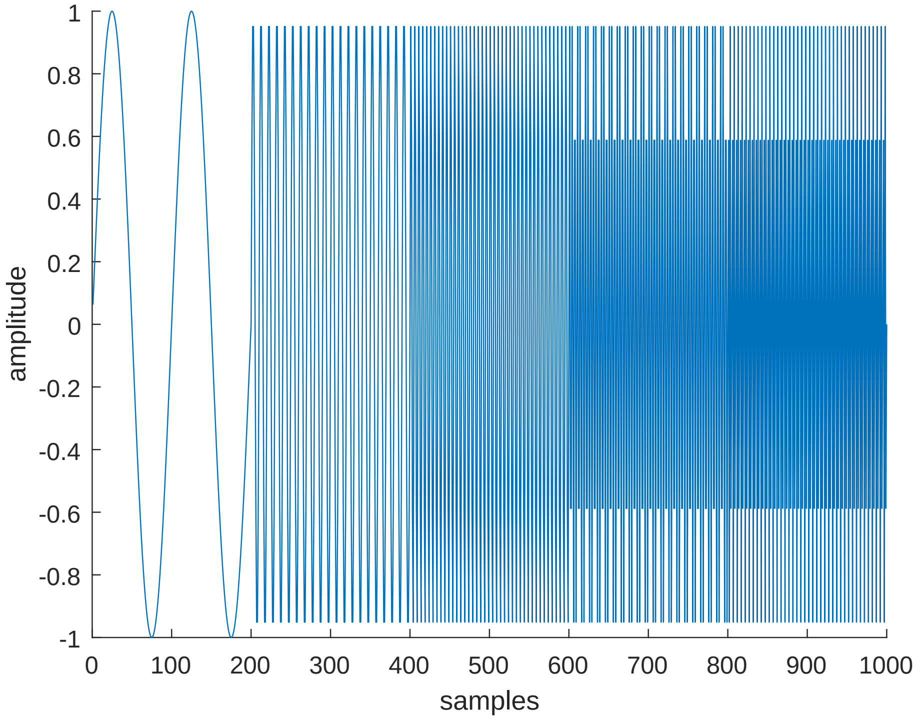

FIGURE 3.5: Sinusoid consisting of five frequencies

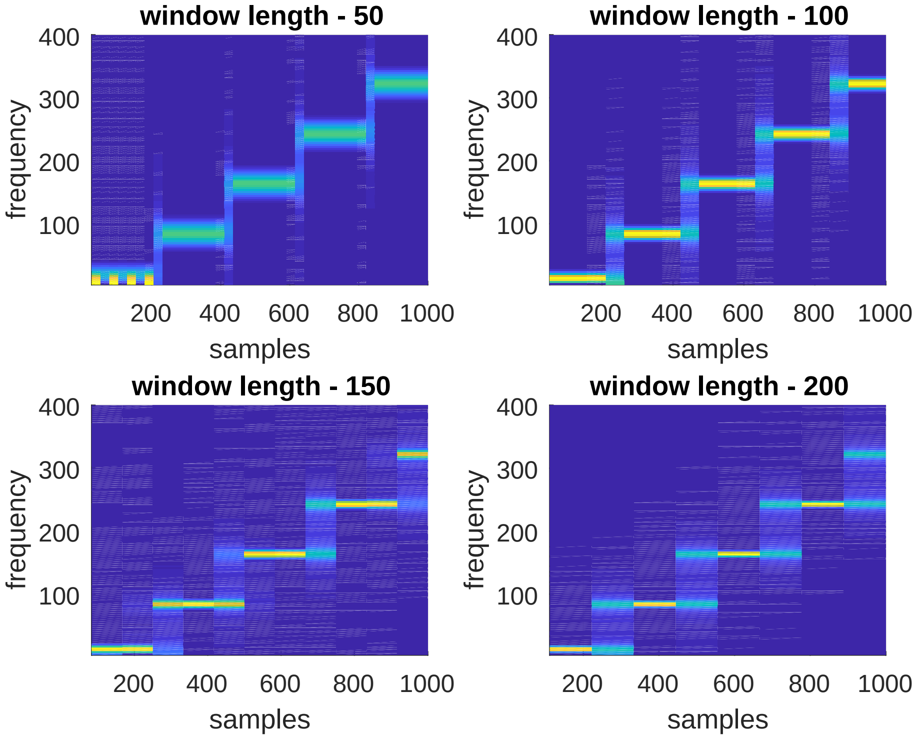

FIGURE 3.6: STFT of the signal with varying window sizes.

3.5 S-Transform

"The S transform is variable window of short time Fourier transform (STFT) or an extension of wavelet transform" [53]

A fundamental limitation is the short time Fourier transform (STFT) is the analysis window of fixed width. This static window will cause a fixed time-frequency resolution for all spectral components. An improperly applied windowing function may render the STFT information useless for interpreting the evolution of the signal through time.

A narrow window will result in a more precise time frame where frequencies of interest occur. But for discrete data, the number of samples within the window may not be sufficient enough to compute the discrete signal frequencies if the window width is too narrow [33][48][53][47].

Stockwell et al. introduces the S-transform, as an extension of the continuous wavelet transform (CWT). The S-transform is based on a moving and scalable Gaussian window and provides a frequency-dependent resolution while maintaining a direct relationship with the Fourier spectrum [47].

3.5.1 Continuous S-transform

The STFT for a continuous signal has been defined in section 3.4, where t' and f correspond to the time of spectral localization and Fourier frequency respectively, and $g(t)$ denotes the window function.

$$STFT_x(t', f) = \int_{-\infty}^{\infty} x(t) g(t - t') e^{-j2\pi ft} dt. \tag{3.3}$$

The S-transform is derived from equation 3.3 by replacing the window function $g(t)$ with a Gaussian function;

$$G(t) = \frac{|f|}{\sqrt{2\pi}} e^{-\frac{t^2 f^2}{2}}. \tag{3.5}$$

By combining equations 3.3 and 3.5, the continuous S-transform may be derived as;

$$STFT_x(t', f) = \int_{-\infty}^{\infty} x(t) \frac{|f|}{\sqrt{2\pi}} e^{-\frac{t^2 f^2}{2}} e^{-j2\pi ft} dt. \tag{3.6}$$

The S-transform is a special case of the STFT, where the window is a frequency dependent Gaussian function. If the window $G(t)$ is wider in time domain, the resultant S-transform will produce a better frequency resolution for lower frequencies. Also, when in the case of $G(t)$ is narrower, it will provide a better time resolution for higher frequencies [53].

The S transform may be expressed as operations on the Fourier spectrum $H(f)$ of $h(t)$, where $f \neq 0$;

$$S(t', f) = \int_{-\infty}^{\infty} H(\tau + f) e^{\frac{-2\pi^2 \tau^2}{f^2}} e^{j2\pi\tau t} d\tau. \tag{3.7}$$

3.5.2 Discrete S-transform

Given a discrete time series $h[kT]$, where $k = 0, 1, ..., N - 1$. The discrete Fourier transform may be expressed as;

$$H\left[\frac{n}{NT}\right] = \frac{1}{N} \sum_{k=0}^{N-1} h[kT] e^{\frac{j2\pi nk}{N}}, \tag{3.8}$$

where $n = 0, 1, ..., N - 1$. Using equations 3.7 and 3.8, the S-transform of a discrete time series $h[bT]$ may be expressed as ($f \rightarrow \frac{n}{NT}$ and $t' \rightarrow kT$) [47];

$$S\left[kT, \frac{n}{NT}\right] = \sum_{m=0}^{N-1} H\left[\frac{m+n}{NT}\right] e^{-\frac{2\pi^2 m^2}{n^2}} e^{\frac{j2\pi mk}{N}}. \tag{3.9}$$

3.6 Comparison between STFT and S-Transform

As explained earlier, the S-transform allows for the tracking of changes in amplitude and frequency with better precision than STFT as the STFT shows a sharp localization of basic components and improves tracking dynamism of the transient components [36]. This is due to the inverse frequency dependence of the Gaussian window as opposed to the fixed width window used in the STFT [47].

Figures 3.8 and 3.8 shows a time frequency representation of a signal using the STFT and the S-transform. The signal, shown by figure 3.7 is synthesized so as it contains two chirps and two high frequency bursts and is expressed as;

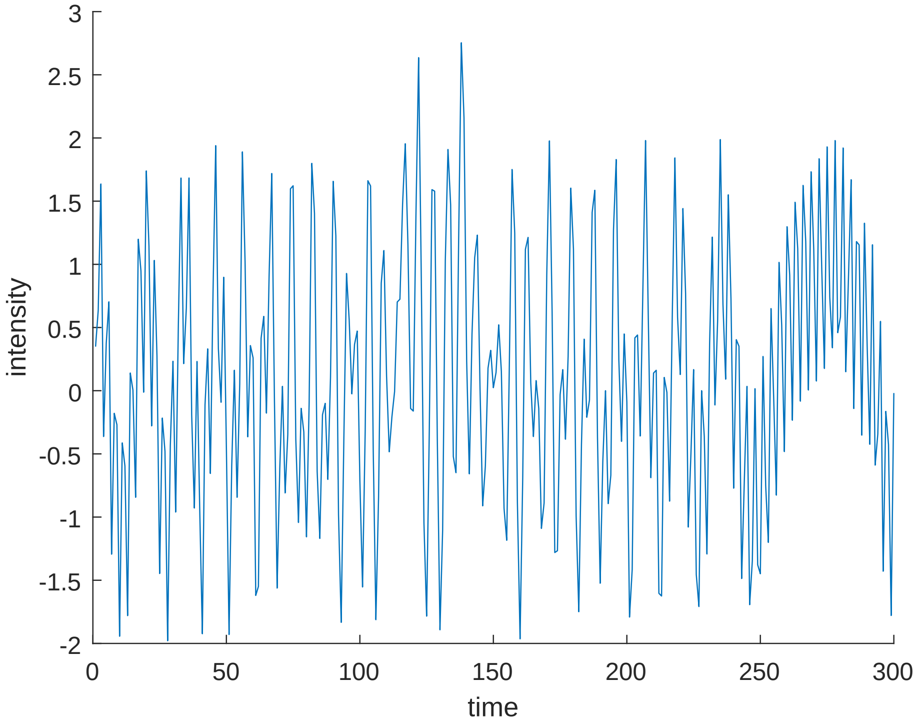

FIGURE 3.7: Synthetic signal consisting of twp chirps and two high frequency bursts

$$\begin{aligned} h(1:300) &= cos(2\pi(10+\frac{t}{7})*\frac{t}{256})+cos(2\pi(\frac{256}{2.8}-\frac{t}{6.0})*\frac{t}{256}), \\ h(114:122) &= h(114:122)+cos(2\pi t0.42), \\ h(114:122) &= h(114:122)+cos(2\pi t0.42). \end{aligned} \tag{3.10}$$

It can be observed that in figure 3.8, which is the STFT matrix using a Gaussian window, both the chirps are detected, but the two high frequency bursts have not been detected. But the s-transform, which is shown by figure 3.9 shows that the two chirps as well as the two high frequency bursts have been detected. Hence the proposed method replaces the STFT with the s-transform for better frequency resolution.

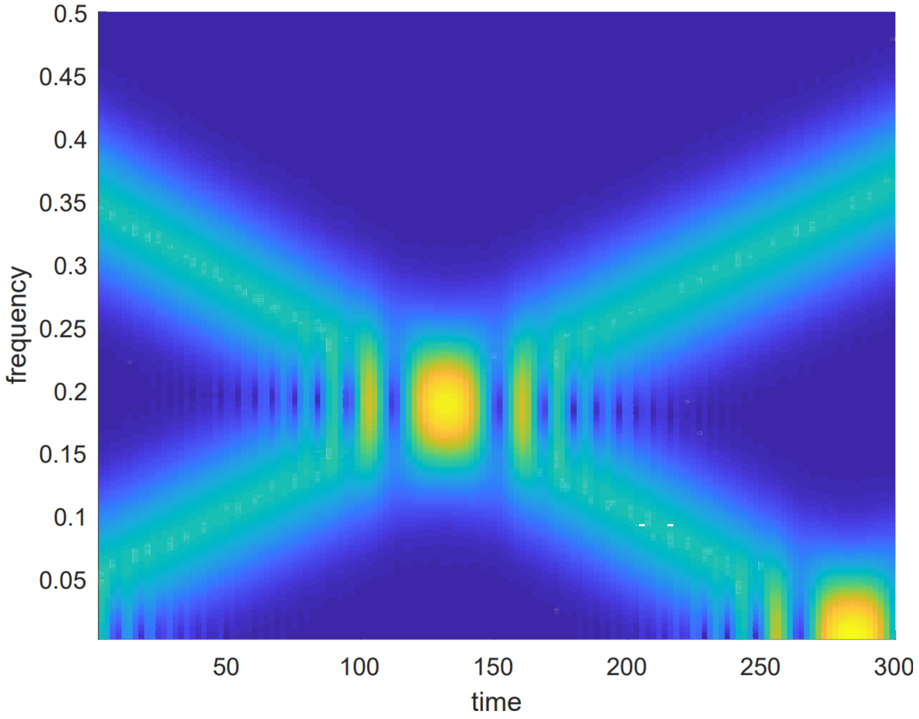

FIGURE 3.8: STFT matrix for for the signal represented in figure 3.7

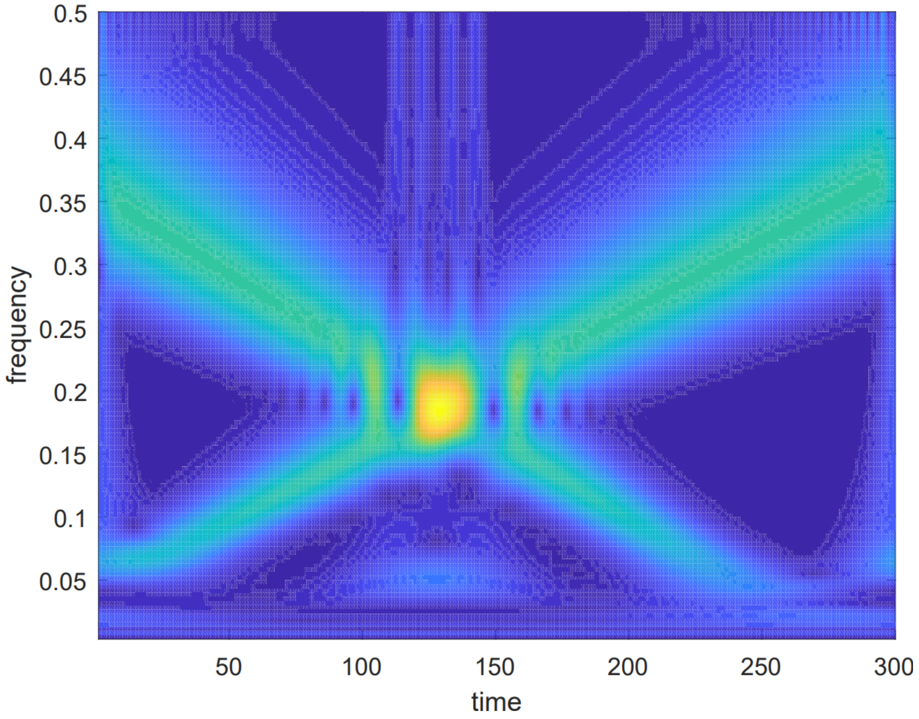

FIGURE 3.9: S-transform matrix for for the signal represented in figure 3.7

4 Musical Theory

4.1 Musical Terminology Definitions

Several musical terminologies used in this thesis are defined as follows;

- Timbre: The characteristic quality of a sound - the attribute that allows a listener to judge that two nonidentical sounds, having the same loudness and pitch, are dissimilar [2].
- Tone Color: A characteristic which allows for sound of two instruments to be different from each other. Terms such as warm, bright, dark or buzzy may be used to describe tone color [11].
- Beat: The basic unit of time of the mensural level - the musical notation system used for Europian vocal polyphonic music [52][3].
- Tempo: The pace of a piece of music. Tempo is typically associated with the rate of periodic events (beats) that listeners perceive to occur at regular temporal intervals [38].
- Time Stretching: Changing of the length, and subsequently the tempo, without affecting its pitch. An audio track, stretched to twice its length, will take twice as long to play, hence having half its original tempo [12][26].
- Pitch: A perceptual property of sound that allows it to be ordered on a frequency-related scale [31].
- Pitch Shifting: Changing the pitch of an audio with or without affecting its length [12].
- Equalization: The process of adjusting the balance between frequency components in an audio signal [49].
- Beat Matching: A disc jockey (DJ) technique to match the tempo of an upcoming track with that of the currently playing track by means of time-stretching of pitch shifting [7].
- Legato: The successive playing of notes where no perpetual gaps are left between two notes. The extreme opposite of *staccato* - where notes are cut very short [31].

4.2 Frequency Ranges of Musical Instruments

A musical instrument is a device of which the primary capability is producing musical sound. Musical instruments may be classified into several categories based upon the method of producing sound [27];

- Persuccion - A musical instrument which produces sound by vibration of its body. Sound is typically proced by striking the instrument. Percussion instruments are categorized into two categories;
  - Idiophone - An instrument whose own substance vibrates to emit sound. A few examples for idiophone instruments are bells, clappers, and rattles.
  - Membranophone - An instrument which produces sound by means of a stretched membrane. The primary example for these type of instruments are drums [16].
- Strings - A stringed instrument will produce sound by the vibration of stretched strings, usually made of plant fibre, metal, animal gut, silk, plastics, and nylons. Most of the stringed instruments use a resonating chamber or a soundboard to amplify the sound. Stringed instruments may be struck, plucked, rubbed or blown to displace the string from its rest position which causes it to vibrate in complex patterns [21].
- Keyboards - A series of keys, levers or push buttons which are pressed to produce sound in a keyboard instrument. In western music, the said keys correspond to consecutive notes in the chromatic scale, and run from the lower bass notes located towards the left of the keyboard, to the higher treble notes in the right. Keyboard instruments have gained a significant importance as they enable a performer to play a large number of notes simultaneously as well as in close succession [17]. There are several methods for the pressed keys to generate sound. Some examples are striking taut strings (piano), vibrating air columns (pipe organ, accordion) or electronic means (synthesizers, electronic keyboards).
- Wind - Wind instruments (aerophones) typically employ a vibrating air column as a medium to produce sound. The primary method of generating different sounds is the change of the length of the vibrating air column. In western music, aerophones are categorized into two main categories based on their composition materials;
  - Woodwinds - Aerophones that are made of wood or other composite materials such as flutes and clarinets. Some modern woodwind instruments are made using metals.

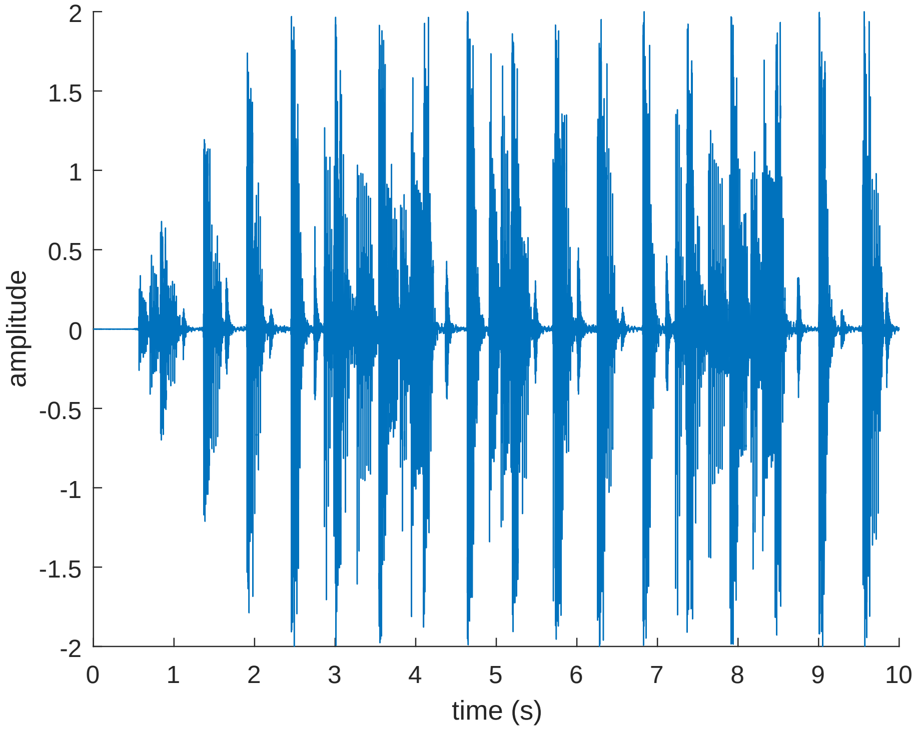

FIGURE 4.1: A popular rock song, of which the visibly obvious beats have been created using the kick-drum and bass guitar

- Brass - Aerophones that are exclusively made out or metal. The most common metal for brass instruments, as the name implies, is brass. Some examples are trumpets and saxophones [28].

- Electronic - Electronic instruments produce or modify sound by electric or electronic means. The music which is produced, and its tonal characteristics (sound), are decided by the composer and hence electronic instruments has given way for an extremely large number of different musical sounds. Acoustic or mechanical instruments, which may amplify the sound electrically or electronically may also be termed electronic instruments although their construction and the produced sound are similar to their acoustic counterparts [42].

These instruments produce a myriad of different sounds which spread through, and sometimes a little away from the audible spectrum. Of diverse sounds musical instruments make, instruments which emit lower frequency sounds are deemed bass instruments and are often, if not exclusively, the primary sources for beats. Figure 4.1 is an example for a popular rock song, with visible beats for which the primary sources are the kick-drum and the bass guitar.

Different instruments which produce sound in different frequencies contribute to music in different ways. Bass instruments typically give way to the *beat* and

TABLE 4.1: Acoustical categorization of frequencies

Frequency	Octave	Description
16Hz - 32Hz	1st	Lower human hearing threshold. The lowest notes of a pipe organ.
32Hz - 512Hz	2nd to 5th	Rhythm frequencies. The upper and lower bass notes lie in this region.
512Hz - 2.048kHz	6th to 7th	Defines human speech, gives a horn-like or tinny quality to sound.
2.048kHz - 8.192kHz	8th to 9th	Gives a presence to speech. Labial and fricative sounds lie here.
8.192kHz - 16.384kHz	10th	*Brilliance* - the sounds of bells and ringing cymbals. Represents sibilance in speech.
16.384kHz - 32.768kHz	11th	*Beyond Brilliance* - nebulous sounds, and passing the upper human hearing threshold

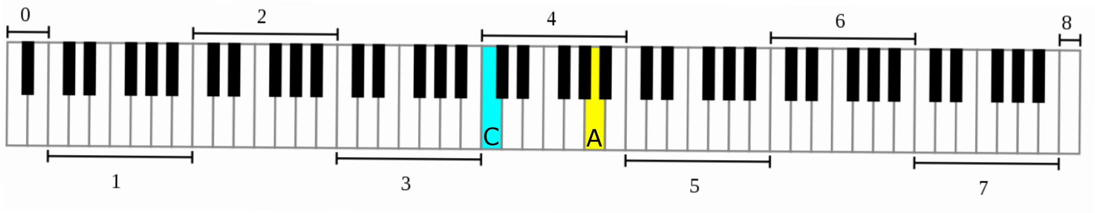

FIGURE 4.2: Standard 88-key piano keyboard with numbered octaves

the *groove*. Instruments of higher frequencies sometimes contribute to the beat as well. A common example for this type of an instrument is the hi-hat cymbal. Medium to lower frequency instruments give way to the *rhythm* which maybe explained as a musical background over which, a main vocal or an instrument, which may be of a slightly higher frequency is sung, or played. Table 4.1 shows how different frequency ranges can be categorized acoustically.

The musical *octave*, is the distance between one musical note and another which is double its frequency. Figure 4.2 shows a standard 88-key piano layout, with the octaves numbered and marked. The key with the marking C represents the note middle C (table 4.2) while the key marked A represents the A440 note.

Table 4.2 shows the frequencies of several notes in different musical instruments. This information shows that much of the audible frequencies f lie in the range of $32Hz \leq f \leq 4.1kHz$. This information is used when defining a coefficient for decimation in equation 5.5 and in deciding the frequency interval that needs to be extracted.

TABLE 4.2: Frequencies of musical instrument notes

Frequency	Description
8.18 Hz	Lowest organ note.
16.35Hz	Lowest note for tuba and large pipe organ. Lowest note in a Bosendorfer Imperial 97-key grand piano [10].
32.70Hz	Lowest C note on a standard 88-key piano.
65.41Hz	Lowest cello note.
130.81Hz	Lowest note for viola and mandola.
261.63Hz	Middle C.
523.25Hz	C note in the middle of the treble clef.
1046.5Hz	The approximate highest note reproducible by the average female human voice.
2093Hz	Highest note for a flute.
4186	Highest note on a standard 88-key piano.

5 Methodology

The method of onset detection presented in this thesis replaces the short time Fourier transform (STFT) used in many beat tracking and onset detection systems [34][20][24][39] with the *s-transform*. This is due to the fact that the s-transform provides a better frequency resolution at lower frequencies and a better time resolution at higher frequencies as explained in section 3.6.

Every onset detection system which rely of spectral features utilize a STFT to obtain a time frequency representation of the audio waveform [30][13][14][40]. But, due to a few limitations in the STFT which have been discussed in the previous chapter, the STFT spectral feature based onset detection methods do not yield good results for some genres of music. Hence the modified system presented in this thesis will utilize a s-transform in place of the standard STFT for higher accuracy levels.

5.1 Proposed method - overview

Although onset detection itself is a pre processing step in any beat tracking system, this thesis will consider onset detection as a whole step and break it into smaller sections. Figure 5.1 shows a flowchart of the broken down sections of the proposed onset detection operation. This chapter will explain each block in the flowchart and how it contributes to the presented method of onset detection.

5.1.1 Read Audio/ Single Channel Conversion

The audio file excerpt is read and converted to a workable format. In the simulations, the audio file format is *.wav* and each file is read as a two channel vector. The two channels in the vector correspond to the left and right channels in the stereo audio file. Figure 5.2 shows the two channels of a stereo song excerpt, plotted separately. The primary criterion for an audio to be stereo is the presence of two *nonidentical* channels. If the two channels are identical, or if one channel is blank, the audio file can be considered mono although there are two channels present [18].

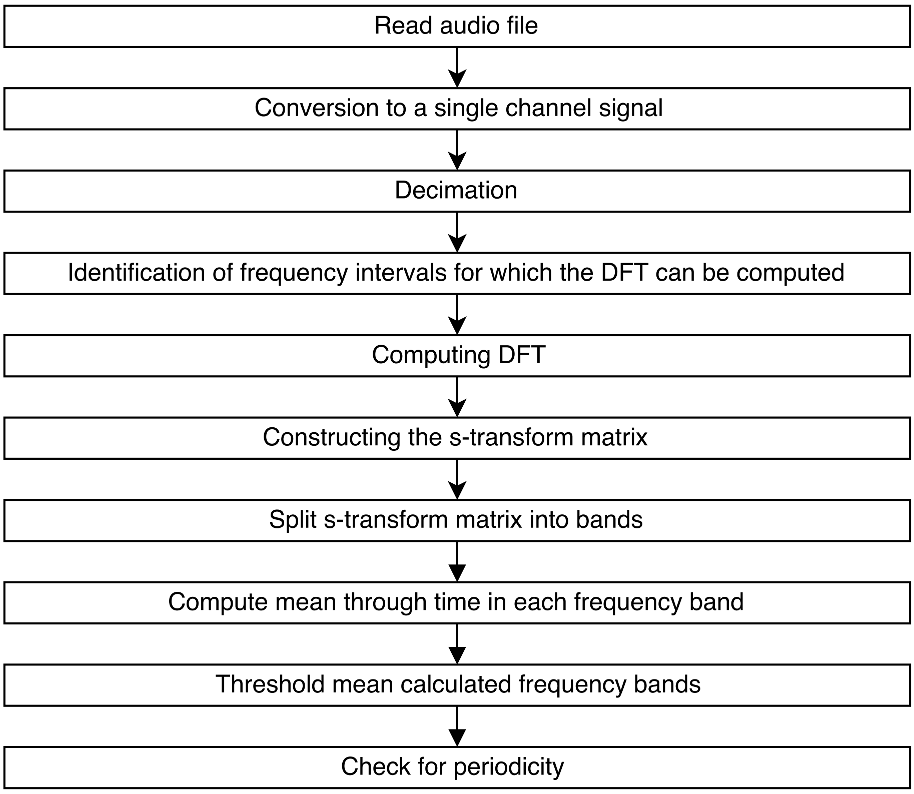
Read audio file
Conversion to a single channel signal
Decimation
Identification of frequency intervals for which the DFT can be computed
Computing DFT
Constructing the s-transform matrix
Split s-transform matrix into bands
Compute mean through time in each frequency band
Threshold mean calculated frequency bands
Check for periodicity

FIGURE 5.1: Flowchart of the proposed onset detection system

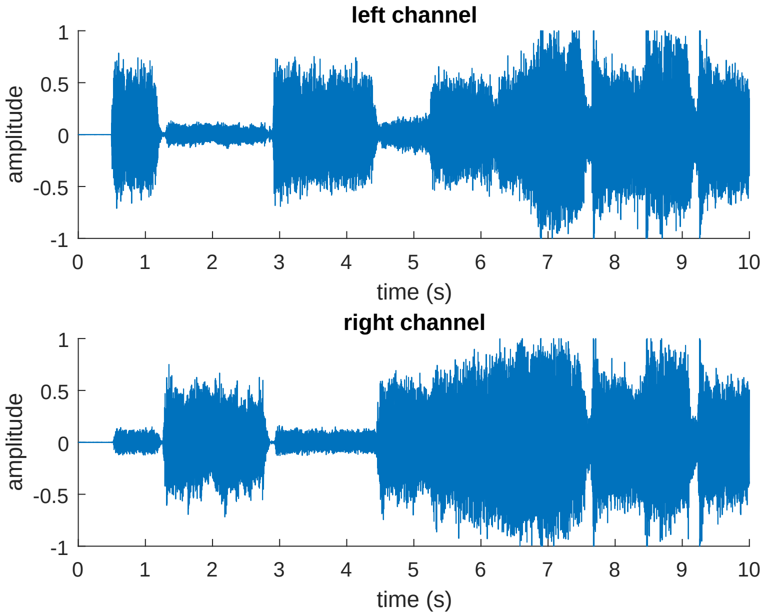

FIGURE 5.2: Left and right channels of a stereo song excerpt, plotted separately

For ease of calculations, the two channels are converged to a single channel vector. In musical terms this is equivalent to a *stereo to mono* conversion. A very simple conversion method was used which is;

$$x(n) = \frac{L(n)}{2} + \frac{R(n)}{2}. \tag{5.1}$$

Where $x(n)$ represents the converted single channel vector and $L(n)$ and $R(n)$ represent the left and right tracks of the audio file respectively. This is true in the case of a stereo recording, where a recording source is placed halfway between the left and right audio sources, and it would capture an approximate amount of $\frac{L(n)}{2} + \frac{R(n)}{2}$ [5].

5.1.2 Decimation

The single channel audio signal is decimated to reduce the sampling rate of the audio file. The term *downsampling* can be used interchangeably with decimation [37]. The following notation can be used to represent the decimation of a sequence $x(n)$, by a factor of D;

$$y(n) = x(nD), \tag{5.2}$$

where $y(n)$ represents the decimated sequence. $y(n)$ is obtained by taking a sample from the sequence $x(n)$ for every D samples. Consider the following sequence $x(n)$;

$$x(n) = 4, 7, 8, 6, 5, 4, 7, 8, 5, 1, 4, 7, -9, 8, 9, -6, 4, 4, -8, -1, 0. \tag{5.3}$$

If $x(n)$ is decimated by a factor of 3 to obtain $y(m)$, $y(m)$ may be expressed as;

$$y(m) = 4, 6, 7, 1, -9, -6, -8. \tag{5.4}$$

When a decimation is performed on a sequence of data, the sampling rate is reduced by the same amount as the *decimation factor*. If a signal of sampling frequency f_s is decimated by a factor of D, the new sampling rate $f_s D$may be expressed as;

$$f_{sD} = \frac{f_s}{D}. \tag{5.5}$$

A decimation factor of d, where $10 \leq d \leq 20$, was chosen as it would eliminate aliasing.

5.1.3 Frequency Ranges

The theory proposed by Nyquist and Shannon specifies the upper bound for the sampling interval of a discretized signal such that the sampled signal contains all of the available frequency information present in the original signal [45]. A signal sampled at a interval finer than the Nyquist interval may be perfectly reconstructed via interpolation [22].

The sampling frequency of the sampled and decimated signal plays a vital component in determining the range of frequencies for which meaningful information can be captured. If the sampling rate of the signal is f_s, the sampling rate after decimation f_sD is expressed in equation 5.5. Hence, according to the Shannon-Nyquist theorem, the range of frequencies f of the decimated signal for which meaningful information can be retrieved is;

$$1 \leq f \leq \frac{f_{sD}}{2}. \tag{5.6}$$

In recorded music and many acoustic performances audio waves are typically sampled at $44.1kHz$, $48kHz$, $88.2kHz$ or $96kHz$ to capture the entire audible spectrum of $20 - 20000Hz$. In many recorded music, such as compact disc's (CD), the sampling rate is of a standard value of $44.1kHz$ [44].

For a recorded piece of music which is sampled at $44100Hz$, which is decimated by a factor of 10, the new sampling rate is $\frac{44100}{10} = 4410Hz$. The range of frequencies f for which meaningful information could be retrieved is $1 \leq f \leq 2250Hz$ (equation 5.6). This range of frequencies are sufficient to detect meaningful frequency content in a piece of music [41].

5.1.4 DFT Computation/ S-Transform

The discrete Fourier transform is calculated for the decimated audio signal. The DFT is calculated using the fast Fourier transform (FFT) algorithm. The DFT thus obtained is used to compute a S-Transform matrix which will be elaborated in a later section.

5.1.5 Band Splitting

Many of the current implementations employ the STFT matrix and a summation in frequency [1], or a mean along the frequency [20], or a median along the frequency [39]. As in the case of onsets being masked by continuous tones of higher amplitude [34] in the case of a temporal analysis, the detection may be hindered by neighboring frequency components of higher amplitudes.

It was found that many beat components occur at relatively low frequencies with the exception of hi-hat notes. Most beats are a resultant of bass instruments - such as kick drums, bass guitars, double basses, bass piano keys, and pedal keyboards. As these instruments produce frequencies which may barely be audible (refer table 4.2), and they are the primary creators of beats, the detection of such frequency components is crucial in an accurate onset detection.

The s-transform matrix is split into a small frequency bands of $50 - 100Hz$. These narrow frequency bands allow for the accurate retrieval of *true* onset components.

5.1.6 Mean Calculation

The mean through time for each frequency band is calculated to obtain an onset envelope of the band. Several of the existing methods has experimented with computing the summation, mean and median through frequency. Section 6.3 elaborates on this and provides a comparison of results for the different operators. It can be seen from figure 6.10 that all three operators act in a similar way. The mean was chosen to obtain an onset envelope in the presented onset detection method as the mean is capable of accurately representing the data in each time band [39].

5.1.7 Thresholding

In signal analysis, thresholding is a is a nonlinear, time-invariant operation and is used to separate the signal into segments according to the value assumed by the signal at each interval. A global threshold, or a local threshold may be applied [33].

Each *frequency band* in the s-transform matrix is applied with a cutoff threshold, where any value above the threshold will be conserved, and any value below the threshold will be disregarded to isolate onset components from the rest of the signal. The most basic thresholding operation is the application of a fixed global threshold. Let the components of signal $f(t)$, above the threshold τ be preserved, and components below τ be discarded to create a thresholded signal $f_{thresh}(t)$;

$$f_{thresh}(t) = \begin{cases} f(t) & \text{if } f(t) \geq \tau, \\ 0 & \text{if } f(t) < \tau. \end{cases} \tag{5.7}$$

As peaks need to be identified in each frequency band, a unique global threshold will be applied to each band. The human auditory system cannot distinguish two sounds if they are less than 50ms apart [6][9][30]. Hence if there are multiple peaks situated in very close proximity to each other they will be considered as a single peak.

Beat causing onsets occur in static frequency bands. Hence the thresholded filtered signal will be checked for periodicity - i.e., the frequency band containing *true* onsets will consist of periodic peaks. The frequency band consisting of the most periodic peaks (one of the bands if there are several periodic bands found - i.e., appendix 8.1.6) will be selected as the onset components of the signal.

5.2 Application of the S-Transform to Musical Onset Detection

As made evident by the previous sections, the S-transform is indeed a more suitable choice for the task of localizing onset components in music signals.

The algorithm, as explained in section 5.1, is quite similar to a majority of the existing methods [30][20][39] with the exception of a few steps. The major difference in the presented method is the usage of the s-transform in place of the STFT, and post-bandwise splitting.

It can be seen from the results presented in chapter 6, that the s-transform based approach provides a significant advantage in a number of cases where the onset detection may not be accurate for traditional methods. This is due to the frequency dependent analysis window found in the s-transform over the fixed length window used in STFT.

6 Results

6.1 Test Data

There have been a number of onset detection and beat tracking systems proposed with varying accuracy levels. Onset locations for kick-drum driven songs such as pop, rock and dance music are easy to identify using existing methods, but as the degree of the strength of transients decreases, the accuracy levels of these methods too, will decrease [30][13][14][40].

A total of 5 such methods were simulated and tested using a selection of music pieces. The pieces of music are of diverse genres and include pop, dance, and classical music. For ease of simulation, 10 second excerpts of each piece was chosen.

Of the selection of music, it can be observed that genres such as dance and pop has very strong kick-drum driven beats, while the classical music pieces do not have visibly identifiable transients. Figure 6.1 shows a comparison of the audio waveforms of a dance music piece and a classical music piece.

It can be clearly seen that the dance music has very strong and visually identifiable beats while the classical piece lacks such visually observable beat locations. This is due to the strong amplitude modulations which mask off onsets carrying beat information. In such cases, it is difficult for most systems to detect onset locations accurately. But in the case of the dance music piece, as the beat locations are emphasized, it is easy for most systems to detect onset locations.

A selection of song excerpts, from a wide array of musical genres have been used to simulate existing onset detection methods. Sections 6.2.1 through 6.2.4 is a brief breakdown of the algorithms and their results. A total of eight song excerpts have been used in the simulations, and the eight song excerpts will be named *example*1 through 8. Each song excerpt is 10 seconds long and have been selected so as the 10 second section represents as much information as possible. Table 6.1 gives a brief description of the song styles and their content.

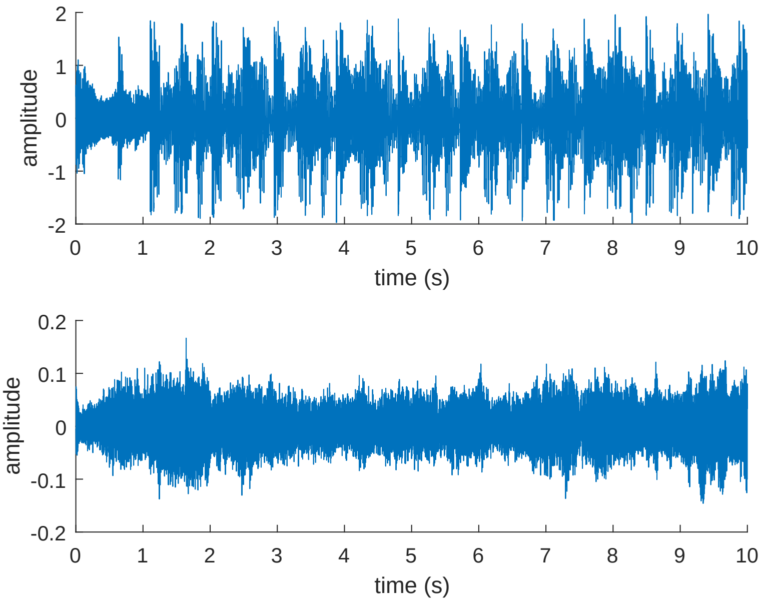

FIGURE 6.1: A comparison of the waveforms of a dance music piece (top), and a classical piece (bottom)

TABLE 6.1: Test dataset

Song	Style	Descriptions
example 1	Electronic dance song	The beats are kick-drum driven and very stong and loud.
example 2	Classical piece	An instrumental section, dominated by violions, cellos and other strings.
example 3	Classical piece	Drums and percussion is lacking. Prominent orchestral string sections.
example 4	Pop song	Beats are maintained by the synthesizer. Kick-drums or bass guitars are not very prominent.
example 5	Pop/dance song	An instrumental section and the melody is driven by saxophone. Kick drum driven beats are present.
example 6	Rock/dance song	Very strong kick-drum and hi-hat driven beats. Possibility of double beat errors due to the kick-drum and hi-hat rhythms
example 7	Pop song	Strong kick drum driven beats are present. The synthesizer is dominant.
example 8	Opera music	A dominant string section is present. Percussion instruments are present.

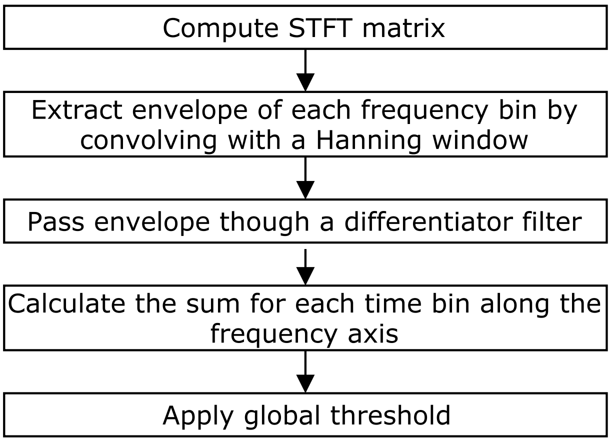
Compute STFT matrix
Extract envelope of each frequency bin by convolving with a Hanning window
Pass envelope though a differentiator filter
Calculate the sum for each time bin along the frequency axis
Apply global threshold

FIGURE 6.2: A flowchart of the algorithm presented in [1]

6.2 Benchmarks

As explained in the earlier section, 5 existing onset detection methods have been chosen as benchmark tests to compare the performance of the proposed method. The selected methods have been elaborated in section 2. This section provides a flowchart for each step in the onset detection method being considered, while stating the parameters of the algorithm used for simulations. Following which, the results of the 8 selected songs are plotted for the method being considered.

6.2.1 Spectral Energy Flux method

The method proposed by Alonso et al. uses a summation of the filtered envelope of the STFT matrix along the frequency axis. Figure 6.2 shows a flowchart of the algorithm.

The STFT is calculated with a window of length 512 samples, the overlap between two succesive windows is 50%. The number of FFT points is taken to be 1024 [1].

Figure 6.3 shows the resultant onset envelope obtained using the algorithm presented in [1]. It can be observed that this method works well for some pieces of music, but fails to identify onsets in some.

The x-axis represents time and the y axis represents onset strengths. It can be observed that the algorithm gives distinct peaks for examples 1, 4, 6, and 7 while it fails to give distinct peaks for the rest of the examples.

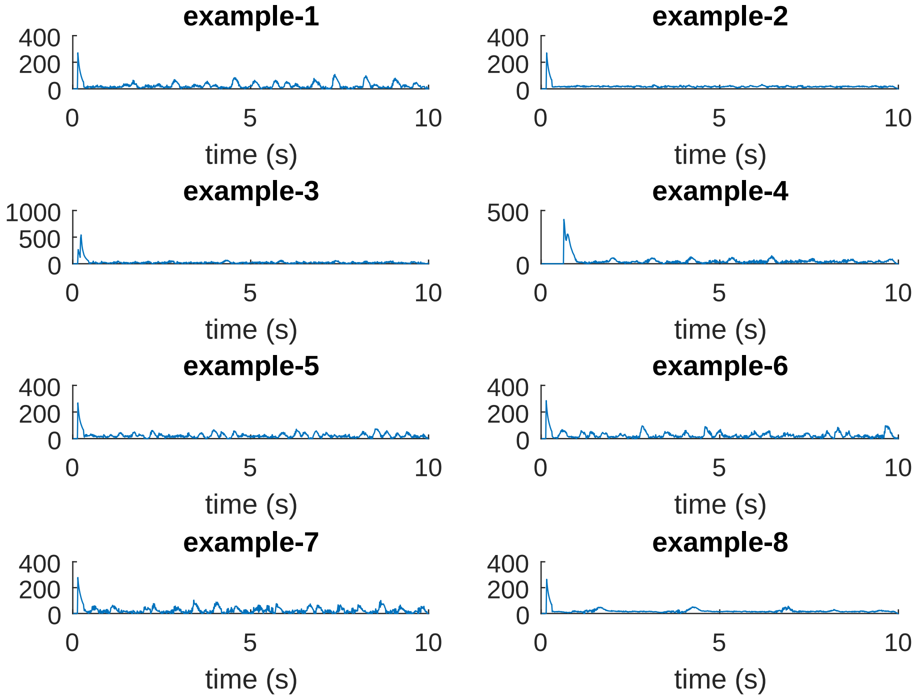

FIGURE 6.3: A comparison of results for [1]

6.2.2 Onset Strength Envelope method

The algorithm presented by Ellis in [20] uses a STFT matrix converted to the Mel scale. Figure 6.4 shows a flowchart of the algorithm. The STFT is calculated with a window length of $32ms$, the window is shifted by $4ms$ so as the overlap is 87.5%. The FFT is calculated with 2048 points [20]. The first order difference along the time for each frequency bin is summed to obtain the onset strengths.

Figure 6.5 shows the results of the algorithm as presented in [20] for the eight examples. The x-axis is the time, and the y-axis is the onset strength. It can be seen that the performance is better than the previous method, but examples 2, 3, and 9 are not detected.

6.2.3 Median Onset Aggregation method

McFee et al. suggests a modification for the work done in [20], by introducing a median operator in place of the summation. The Mel scaled STFT matrix is used to identify onset components. Figure 6.6 shows a flowchart of the algorithm presented in [39].

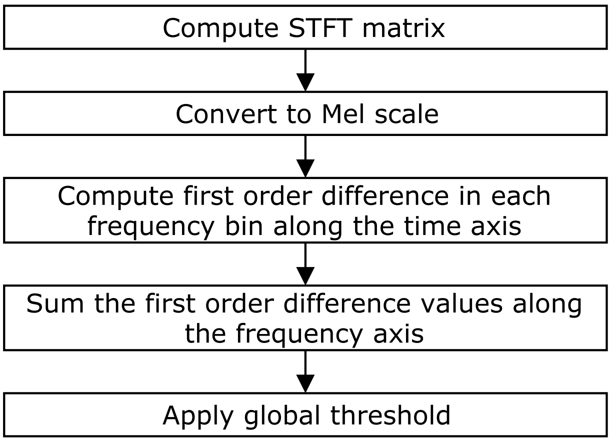
Compute STFT matrix
Convert to Mel scale
Compute first order difference in each frequency bin along the time axis
Sum the first order difference values along the frequency axis
Apply global threshold

FIGURE 6.4: A flowchart of the algorithm presented in [20]

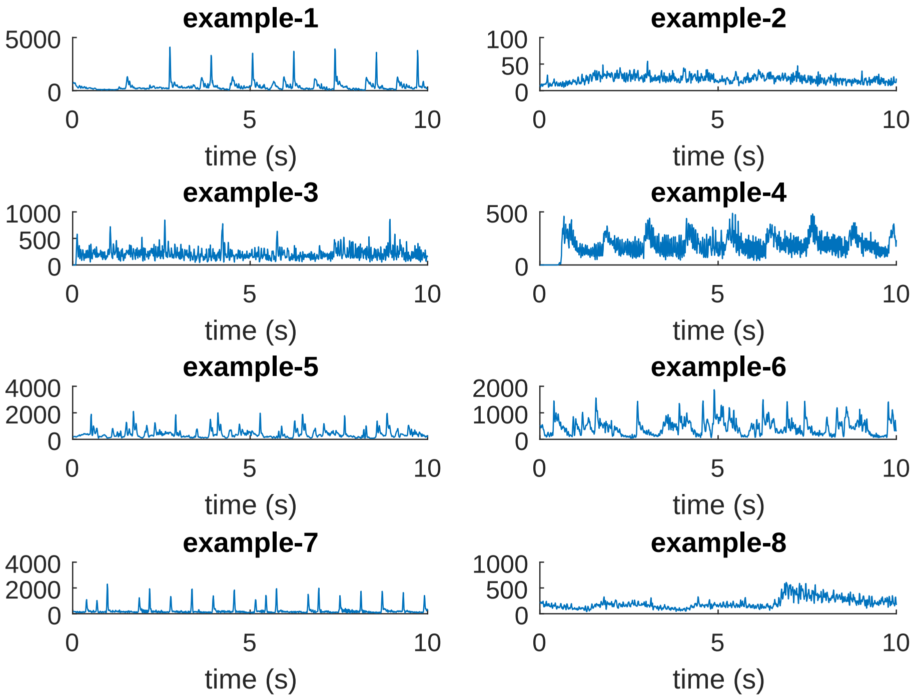

FIGURE 6.5: A comparison of results for [20]

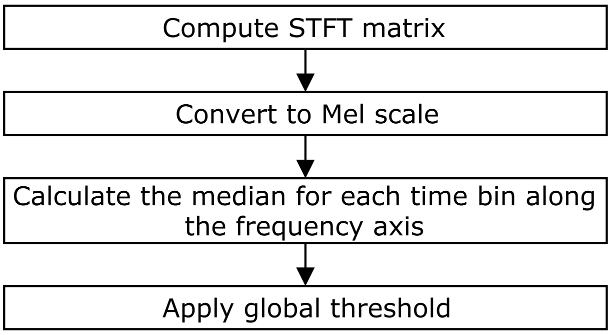
Compute STFT matrix
Convert to Mel scale
Calculate the median for each time bin along the frequency axis
Apply global threshold

FIGURE 6.6: A flowchart of the algorithm presented in [39]

The results for the eight examples when used as an input to [39] is given in figure 6.7. The STFT was calculated with a window length of 64ms and a hop size of 8ms. The number of FFT points used is 1024.

6.2.4 Energy Flux method

Another modification to the algorithm presented by [20] is given by Laroche et al. The modification calls for the Mel scale conversion to be be replaced with a square root operator. Figure 6.8 shows a flow chart of the modified algorithm as presented by [34]. The STFT is calculated with a window of 10ms and a hop of 2ms. The number of FFT points is 1024.

Figure 6.9 shows the results of the modified algorithm when tested with the eight examples. It can be seen that while results are similar to [20] and [39], examples 2, 3, and 8 have still not been recognized.

The sections show that existing methods do not yield graceful results for musical genres such as classical music and opera music. To better capture information on beat causing onsets in these styles, the s-transform was used in place of the STFT as explained in section 5. The mean is calculated in frequency bands of 50Hz - 100Hz for better capturing on onset information. Section 6.3 provides several simulation results to justify the splitting of the s-transform matrix into frequency bands.

6.3 S-Transform - Mean, Median and Sum operators through Complete Frequency Bands

As explained in section 6.1, many of the existing systems have difficulty in localizing onset components in genres such a classical music and opera music

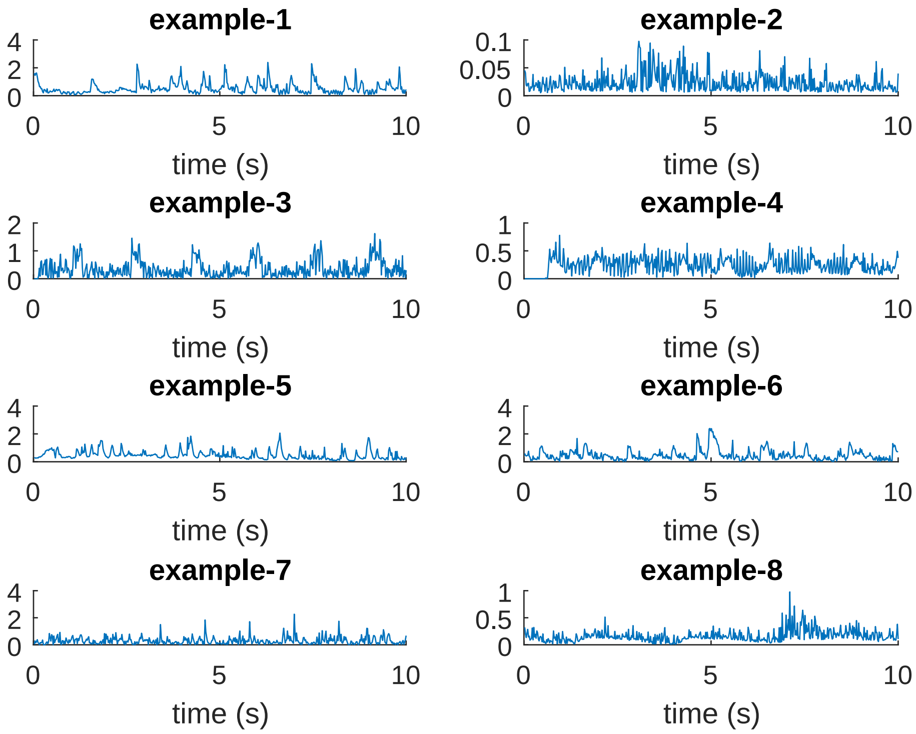

FIGURE 6.7: A comparison of results for [39]

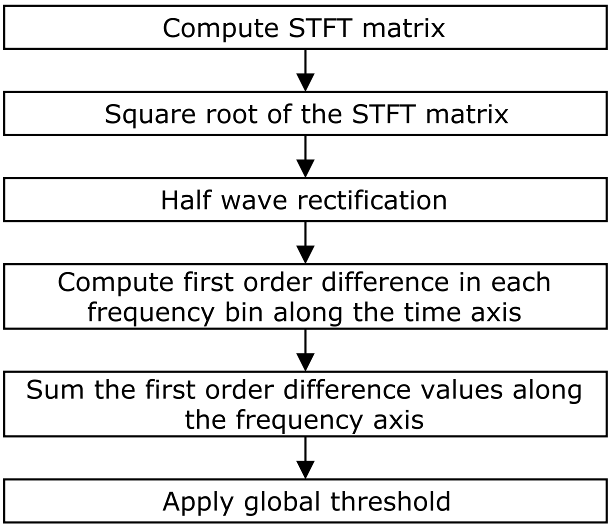
Compute STFT matrix
Square root of the STFT matrix
Half wave rectification
Compute first order difference in each frequency bin along the time axis
Sum the first order difference values along the frequency axis
Apply global threshold

FIGURE 6.8: A flowchart of the algorithm presented in [34]

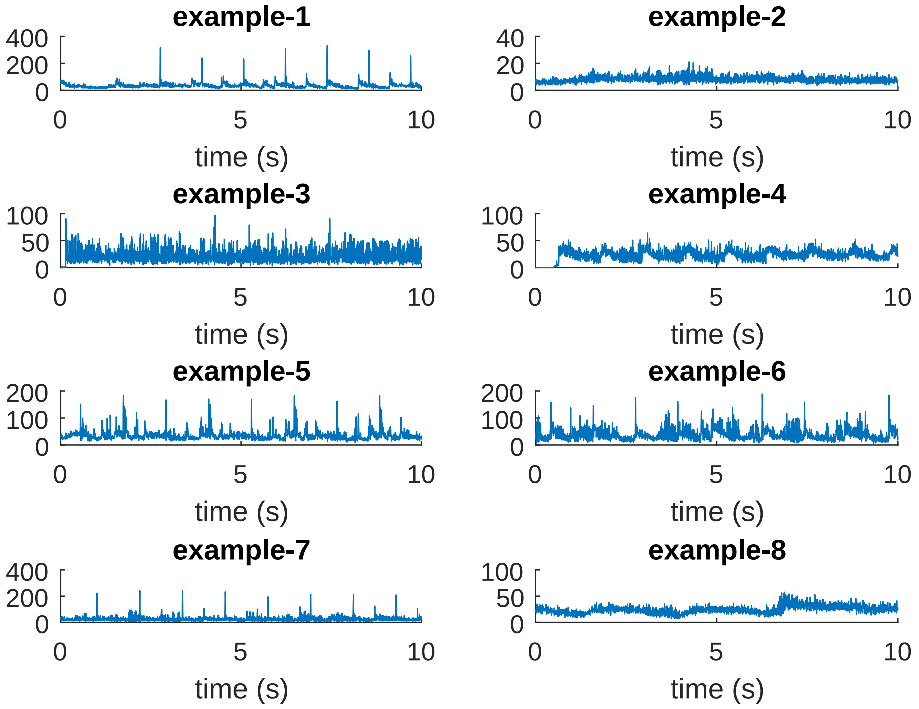

FIGURE 6.9: A flowchart of the algorithm presented in [34]

[30]. The method presented by this thesis uses a s-transform in place of the STFT for better frequency component localization.

Several methods were tried to isolate onset components form the s-transform matrix. These methods have been inspired from existing algorithms. Some of these include;

- Summation through each time bin [20]
- Median through each time bin [34]
- Mean through each time bin [39]

Figure 6.10 shows a comparison between the three operations for three of the eight examples considered in section 6.1.

It can be seen that the s-transform yields better results for some but the overall result is fairly meaningless unless the beats are very prominent in the song. As existing methods yield somewhere simpler computations for songs with prominent beats, and similar results, a bandwise analysis is performed to better capture onset components.

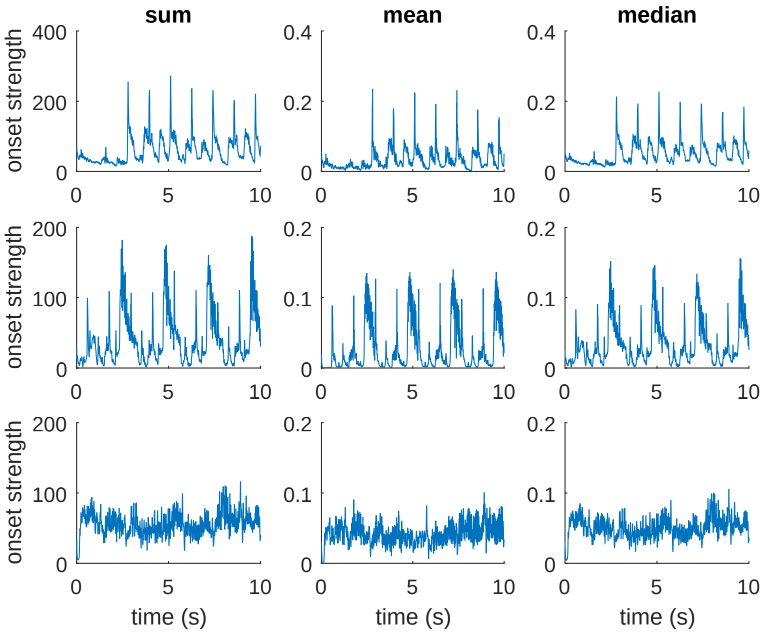

FIGURE 6.10: Difference between the summation, median, and mean through each time bin in the s-transform matrix

6.4 Bandwise Analysis

Beat causing onsets generally occur in a static frequency band - i.e., Onsets which influence on the beats generally have the same - or very similar frequencies. This is due to the fact that the beat will generally be driven by a single instrument. *A bass guitar and a trumpet will not create alternating beats.* Generally, the beat will be maintained by a bass guitar, kick drum, double bass or an instrument which would emit sounds of relatively lower frequencies when compared to others it is being played with.

To successfully localize and identify beat causing onsets the s-transform matrix, is split into bands of $50Hz - 100Hz$. The median operator is performed inside the frequency band which will yield onset envelopes. The median, mean, and sum operators retain similar results when computed (refer figure 6.10) and the median operator is selected as it is the best representation for the data. True onsets will be filtered using a global threshold, and a comparison of periodicity will be performed - i.e., beat causing onsets will be periodic and if a frequency band contains equally spaced distinct onsets, it can be assumed that they are beat causing onset components.

Figure 6.11 shows the mean operator for each complete time bin - which is the mean of all frequency components at a single time instant, plotted against onset strength. It can be seen that the information presented in figure 6.11 is meaningless in the case where beat causing onsets are concerned.

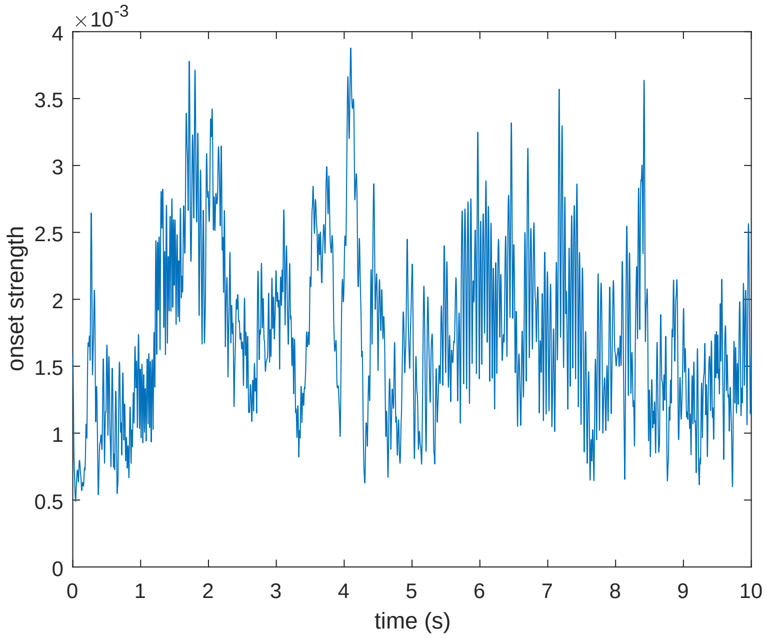

FIGURE 6.11: Example 2 - the mean operator is applied to each time bin in the s-transform matrix

To extract meaningful information, the s-transform is split into frequency bands. The s-transform matrix for example 2, has been split into frequency bands each of $100Hz$. The first 10 bands are plotted in figure 6.12.

It can be clearly seen that there are distinct peaks in frequency ranges $300Hz - 400Hz$, and $400Hz - 500Hz$. Table 4.1 shows that this frequency range is between the 2^{nd} and 5^{th} octaves where upper and lower bass note frequencies lie. This information supports the claim that beat causing onsets will occur at static frequency bands.

Consider example 4. Figure 6.10 shows that example 4 did not yield good results for the complete summation, mean or median across each time bin in the s-transform matrix. Figure 6.13 shows that when split into bands, some frequency bands retain meaningful information which have been masked by other bands. In the case of example 4, frequencies $900Hz - 1000Hz$ retains meaningful information.

The method was applied to all 8 examples with successful results. Refer appendix 8 for more detailed results for all example music pieces. Upon successfully localizing onset components within their respective frequency bands, they were extracted by means of a threshold. And the most suitable onset component was selected by periodicity.

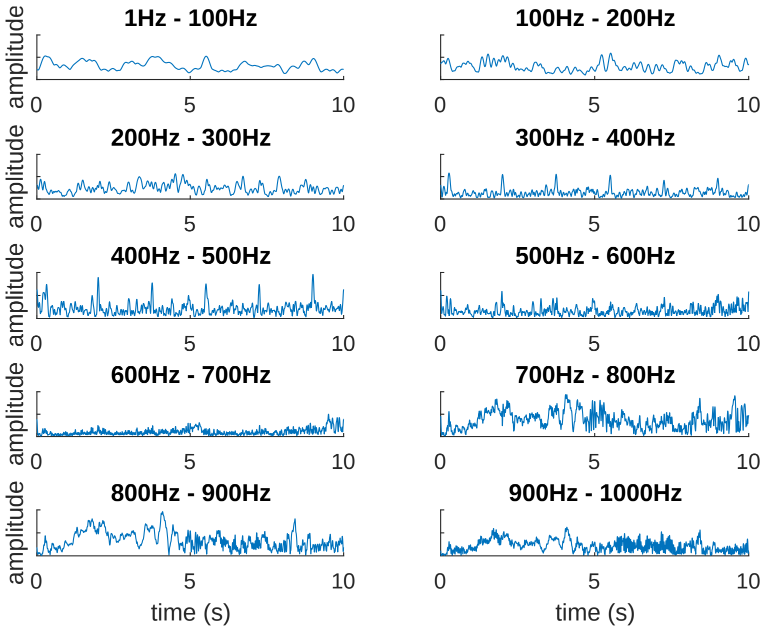

FIGURE 6.12: Example 2 - s-transform matrix is split into frequency bands of 100*Hz*

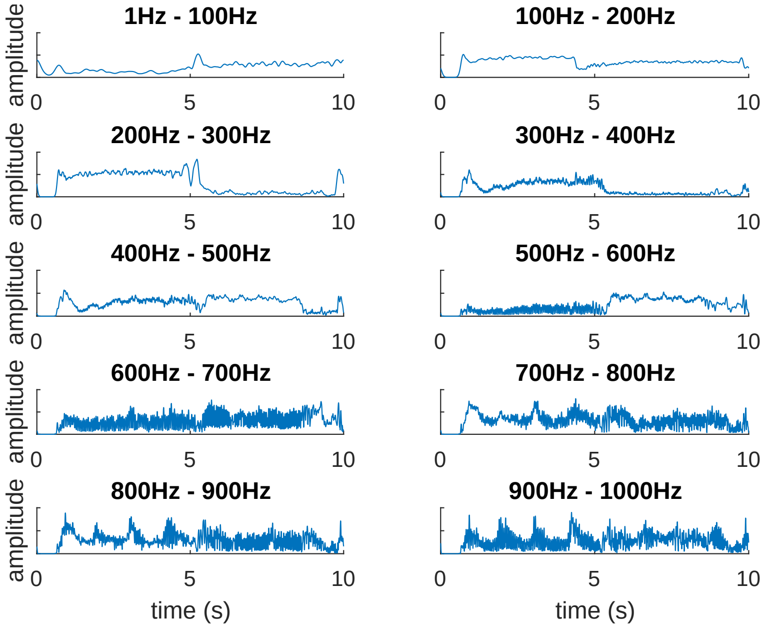

FIGURE 6.13: Example 4 - s-transform matrix is split into frequency bands of 100*Hz*

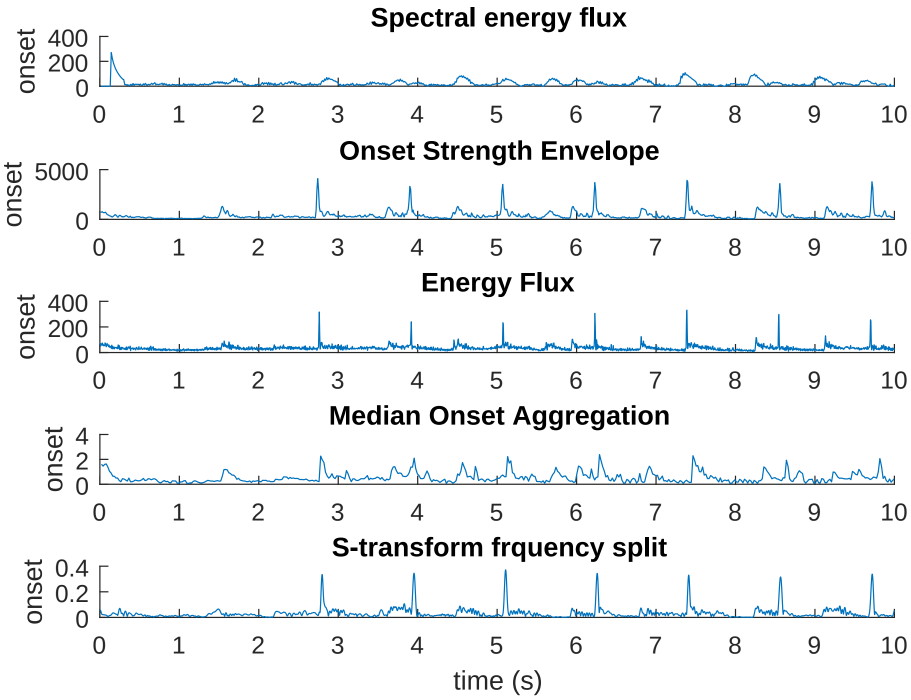

FIGURE 6.14: Example 1 - A comparison of existing methods and the proposed method

6.5 Comparison with existing methods

The proposed method: splitting of the s-transform matrix into frequency bands has significant advantages over existing methods for some styles of music for which they do not exhibit good performance. For music genres where sharply defined onsets are present, the proposed method has similar performance to existing methods.

A comparison of four existing methods with the proposed method is presented in figures 6.14 and 6.15, for two musical styles.

Example 1 is a electronic dance song. The beat locations are clearly visible in the waveform itself. It can be observed that the proposed method and the existing methods have similar performance for music of these types.

Example 2 is a classical music piece. The existing methods do not yield a very good performance for music of genres of the likes of example 2. But it can clearly be seen that the proposed method would impose a significant advantage for music in the likes of example 2.

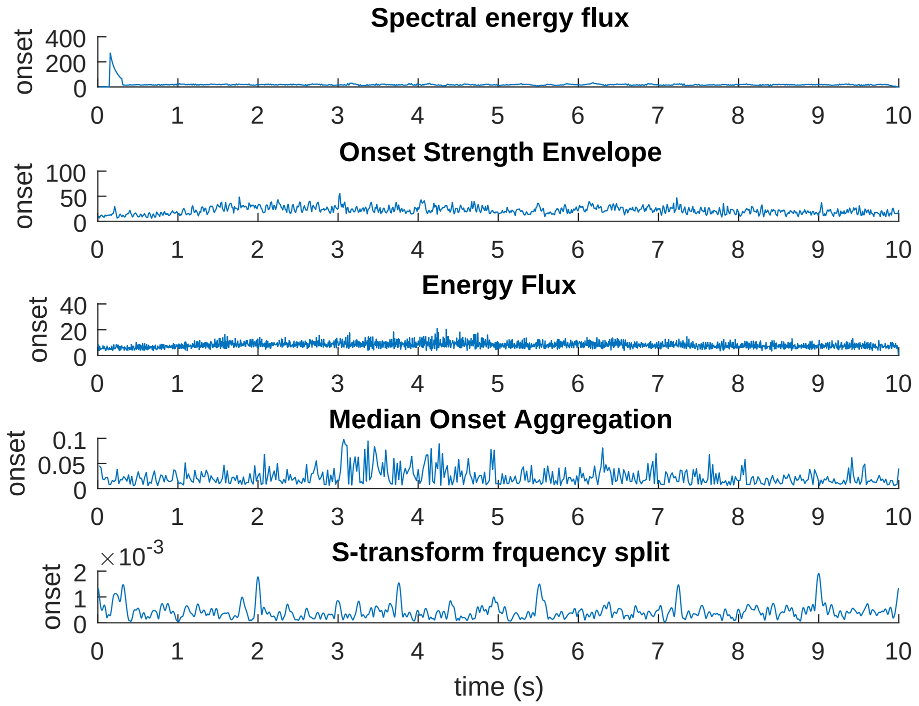

FIGURE 6.15: Example 2 - A comparison of existing methods and the proposed method

7 Conclusions and Future Work

This chapter serves as a summary of the conclusions of this thesis, while providing a brief highlight of the advantages of the proposed method over existing methods. Following which, several possible modifications are highlighted to improve the proposed method.

7.1 Conclusions

While existing methods excel at identifying beat causing onset locations in musical genres such as rock music dance music, they fail in identification of beat causing onsets in genres such as classical music and opera music.

There are two main cases when an onset detection method would not yield accurate results as highlighted in section 1;

1. When the rhythm of the music is less pronounced,
2. During rapid tempo changes [30].

The work done in this thesis proposes a method that successfully overcomes point 1 in the above list. The rhythm may be less pronounced die to strong amplitude modulations, as found in classical music and when neighboring frequency components may mask off beat causing onsets.

The method proposed by this thesis introduces the s-transform to replace the STFT which is used in many of the existing onset detection methods. The proposed method also introduces a frequency band splitting in the s-transform matrix to localize musical onsets in neighboring frequency bands. Chapter 4 provides a basis for this operation, as beats are usually caused by a single, lower frequency instrument.

Onset components may be gracefully detected using the proposed method for songs for which other methods do not yield promising results. The test data is of diverse musical genres and is presented in section 6.1. A comparison between onsets detected using existing mehtods, and onsets detected by the proposed method is presented in chapter 6.

The s-transform, provides a better capture of spectral information due to its frequency dependent analysis window. Onset components which may be masked by those in neighboring frequencies are isolated by employing frequency band splitting. The beat causing onsets of a piece of music can be gracefully located through the proposed method by means of frequency band splitting, thresholding and periodicity checking.

7.2 Recommendations

The FFT shines in the fact that it takes an extremely small time to compute even a large dataset due to its efficient factoring algorithm. The primary reason for many systems to employ the STFT may be its computational speed.

Although the results have proven that it is more favorable, a significant disadvantage in the s-transform based approach is the computational cost. The s-transform is computationally costly and requires a significant time to compute, even on a high-end computer. This drawback limits the presented method from being used in real time. Modified s-transform functions may be applied to increase computational speed.

The proposed method takes into consideration the periodicity of onsets - i.e., the period between two successive onsets need to be similar. Due to this hypothesis, the proposed system will only work for songs with a constant time signature - i.e., If the song goes from a $\frac{4}{4}$ to a $\frac{6}{8}$, the system will fail as the periodicity between onsets is lost. Due to the periodicity factor, abrupt and drastic tempo changes may also not yield accurate results.

8 Appendix A

8.1 Bandwise Splitting of S-Transform matrix

The s-transform matrix is split into frequency bands of 100*Hz* as elaborated in section 6.4. Following are the resultant split frequency bands for all eight example tracks.

8.1.1 Example 1

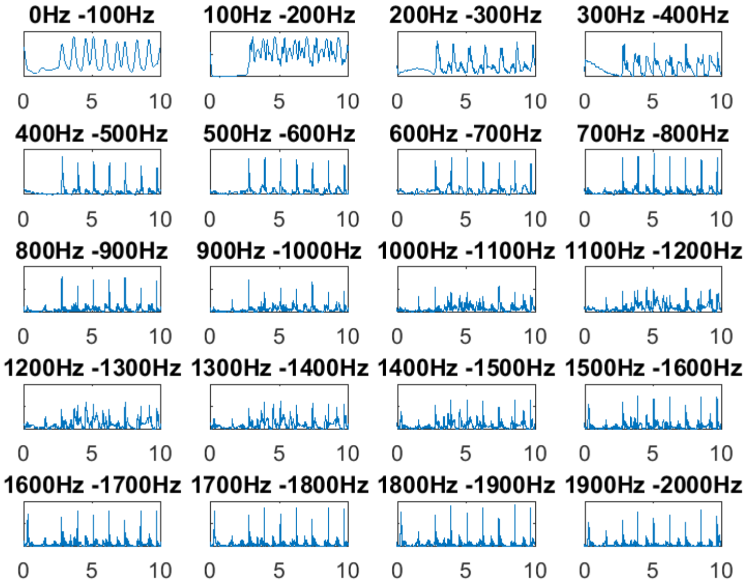

FIGURE 8.1: Example 1, split into 20 frequency bands of 100*Hz*

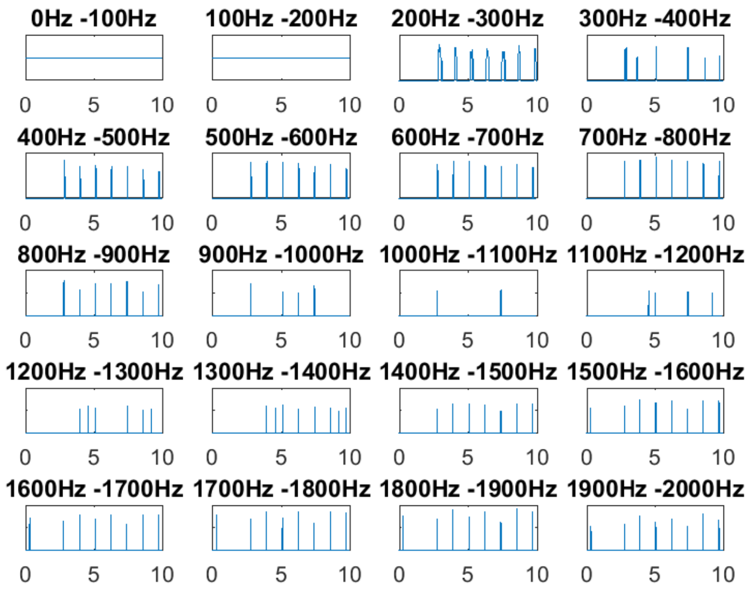

FIGURE 8.2: Example 1, split into 20 frequency bands of 100Hz, and thresholded

8.1.2 Example 2

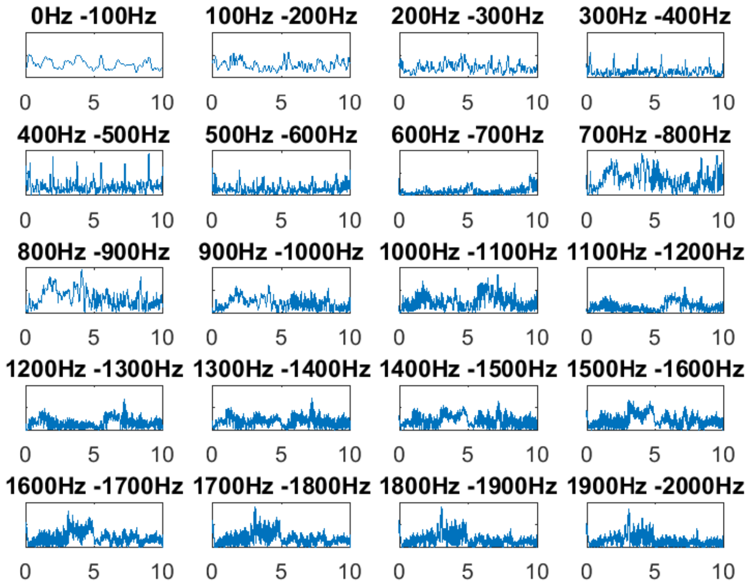

FIGURE 8.3: Example 2, split into 20 frequency bands of 100Hz

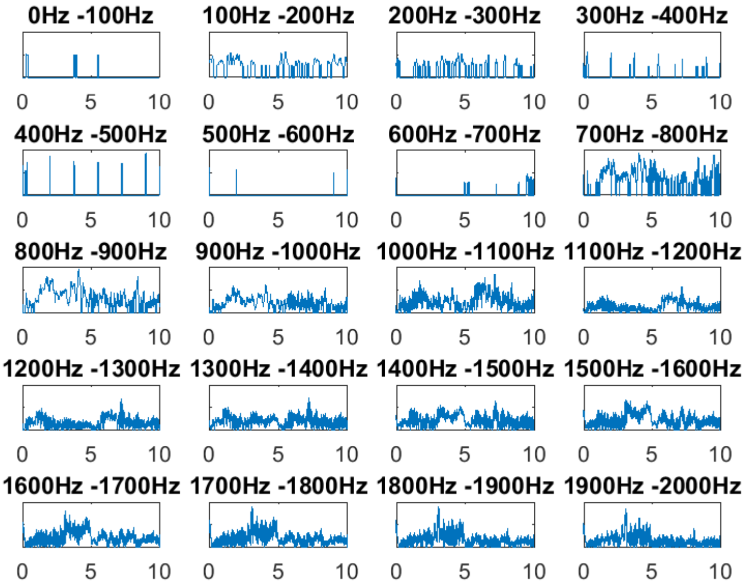

FIGURE 8.4: Example 2, split into 20 frequency bands of 100Hz, and thresholded

8.1.3 Example 3

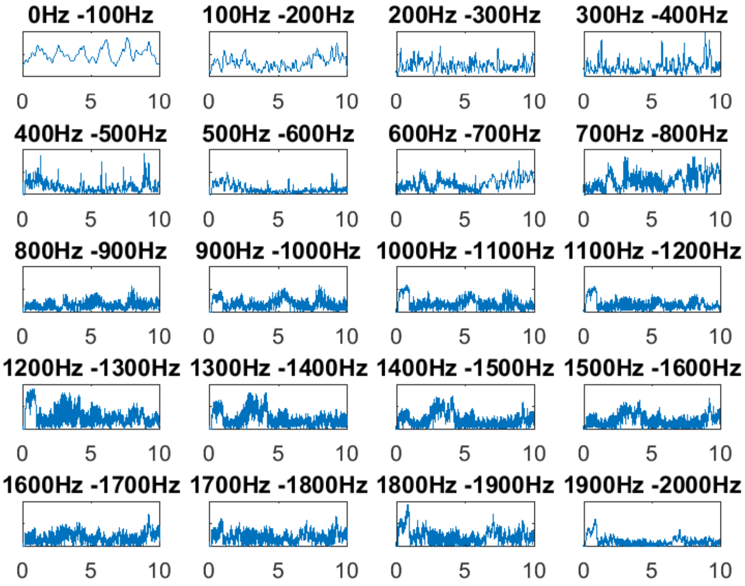

FIGURE 8.5: Example 3, split into 20 frequency bands of 100Hz

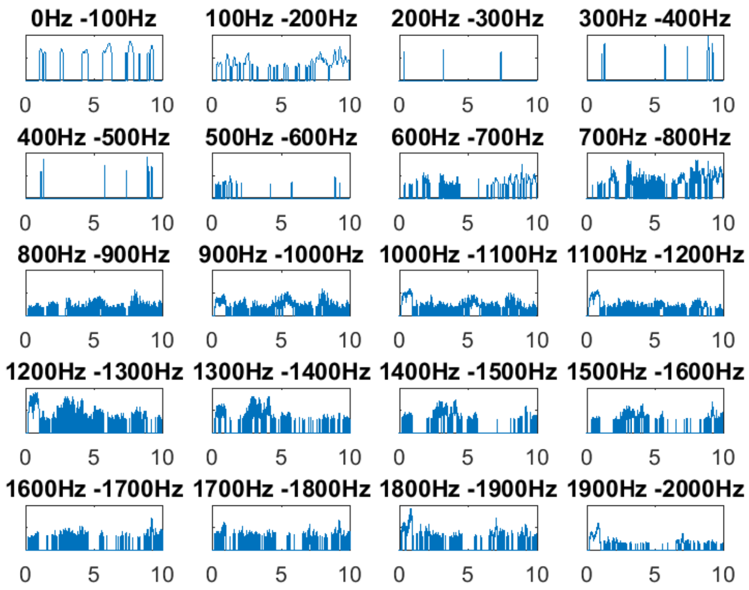

FIGURE 8.6: Example 3, split into 20 frequency bands of 100Hz, and thresholded

8.1.4 Example 4

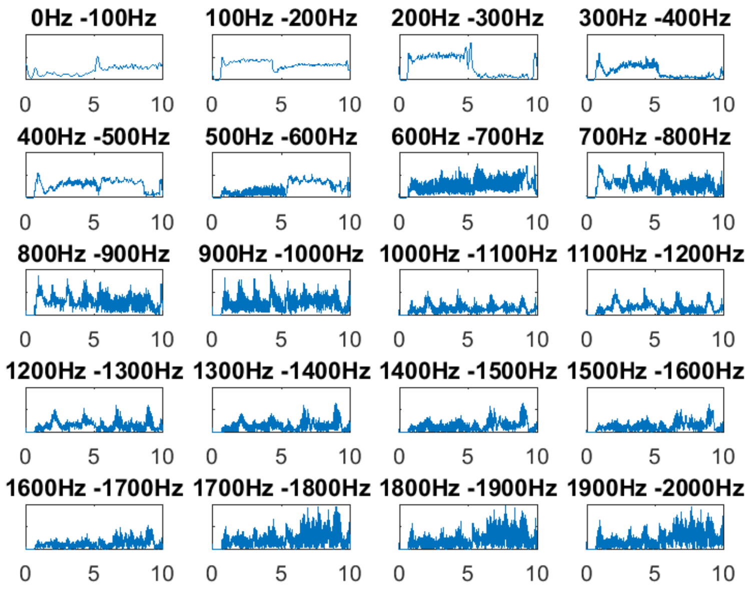

FIGURE 8.7: Example 4, split into 20 frequency bands of 100Hz

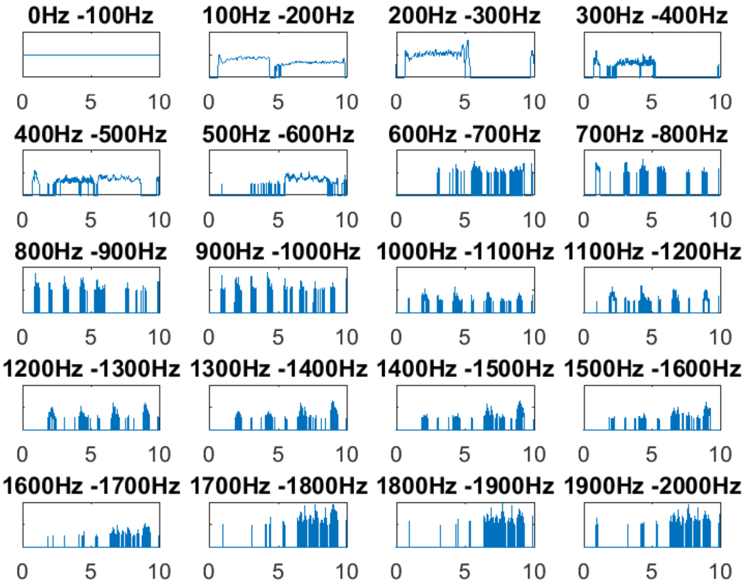

FIGURE 8.8: Example 4, split into 20 frequency bands of 100Hz, and thresholded

8.1.5 Example 5

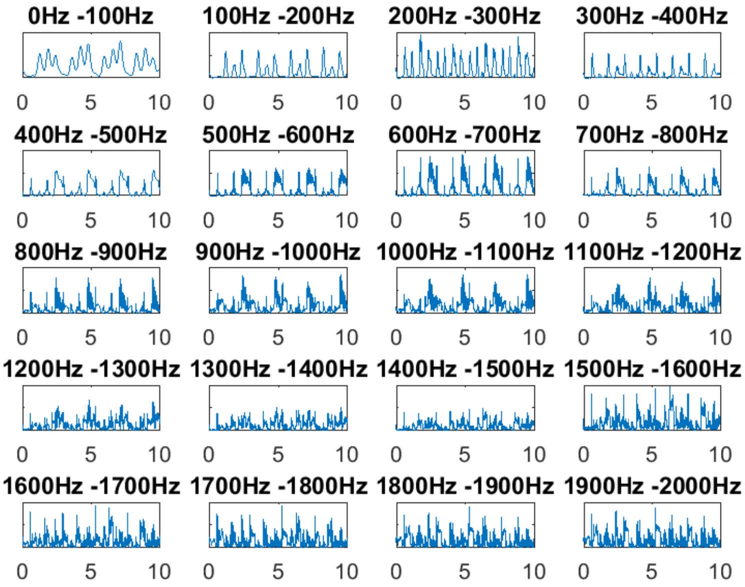

FIGURE 8.9: Example 5, split into 20 frequency bands of 100Hz

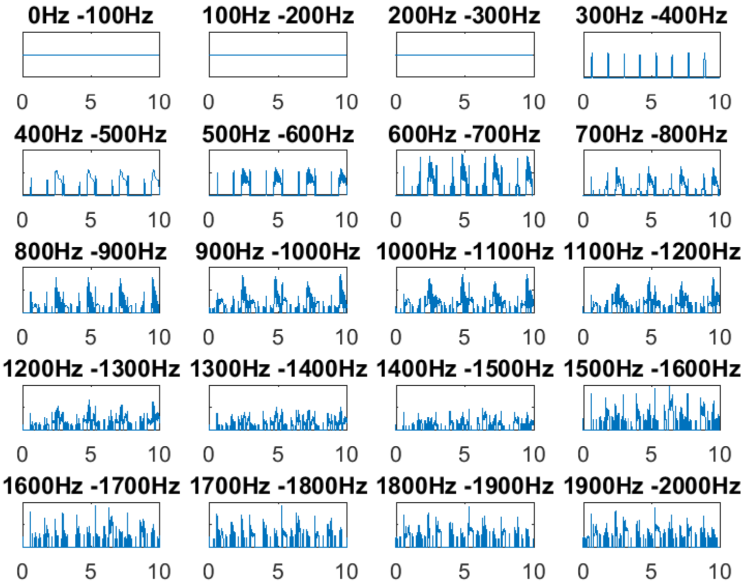

FIGURE 8.10: Example 5, split into 20 frequency bands of 100Hz, and thresholded

8.1.6 Example 6

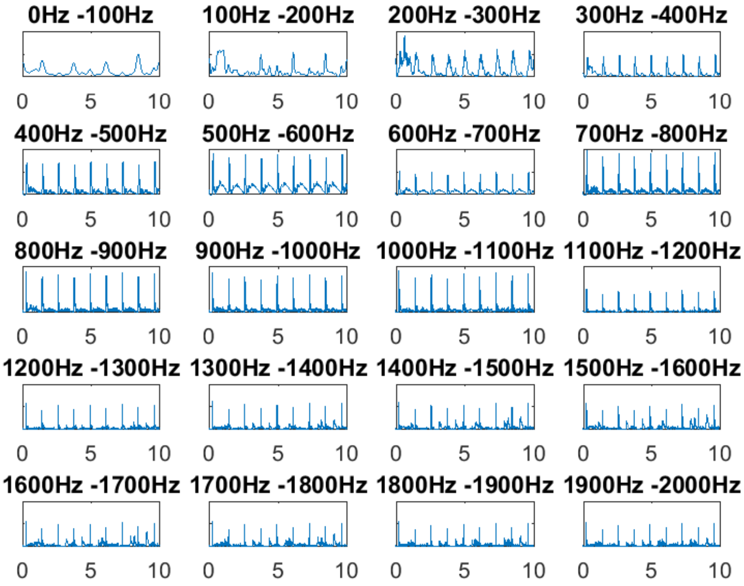

FIGURE 8.11: Example 6, split into 20 frequency bands of 100Hz

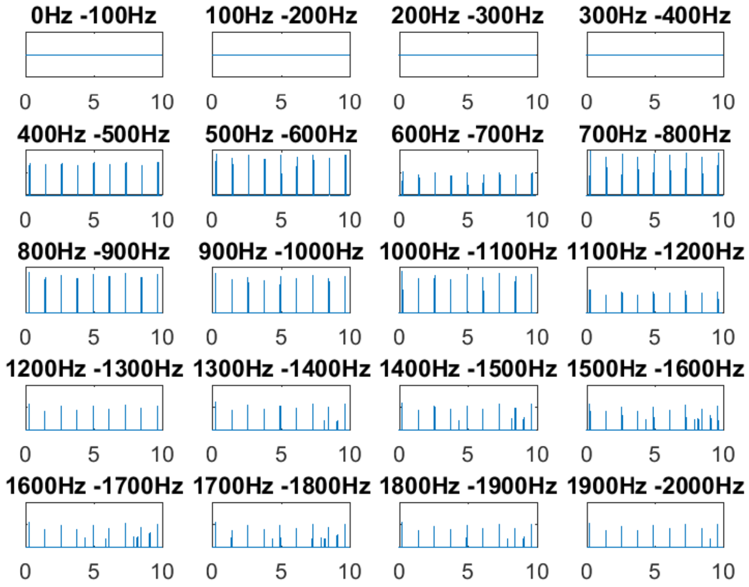

FIGURE 8.12: Example 6, split into 20 frequency bands of 100Hz, and thresholded

8.1.7 Example 7

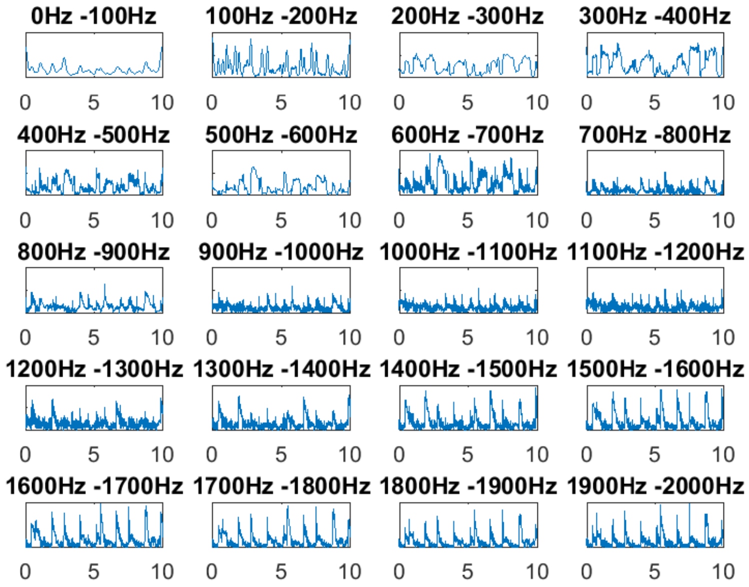

FIGURE 8.13: Example 7, split into 20 frequency bands of 100Hz

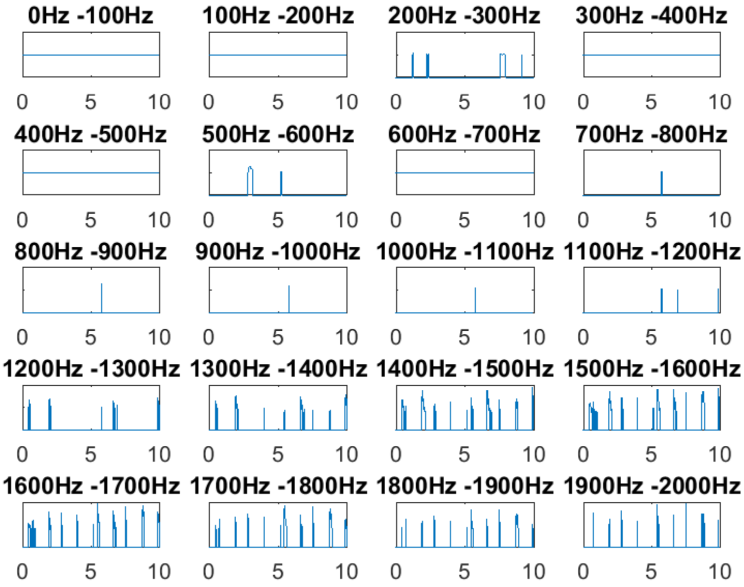

FIGURE 8.14: Example 7, split into 20 frequency bands of $100Hz$, and thresholded

Bibliography

[1] Miguel Alonso, Bertrand David, and Gaël Richard. *Tempo and Beat Estimation of Musical Signals*. Oct. 2004.

[2] Acoustical Society of America Standards Secretariat. *Acoustical Terminology ANSI S1.1-1994 (ASA 111-1994)*. American National Standard. ANSI / Acoustical Society of America, 1994.

[3] Willi Apel. *The Notation of Polyphonic Music, 900–1600*. 5824900M. Cambridge, Mass., Mediaeval Academy of America, 1961.

[4] Jose Soares Augusto. *What is the difference between decimation in time and decimation in frequency?*

[5] Jerry Avins. *Converting Stereo to Mono taking into account Phase modulation*.

[6] J. P. Bello et al. "A Tutorial on Onset Detection in Music Signals". In: *IEEE Transactions on Speech and Audio Processing* 13.5 (2005), pp. 1035–1047. ISSN: 1063-6676. DOI: 10.1109/TSA.2005.851998.

[7] Ben Bowers. *Breaking Down the Art of Beatmatching*. Gear Patrol. URL: https://gearpatrol.com/2015/07/10/how-to-beatmatch-dj/.

[8] Andrew G. Klein C. Richard Johnson Jr William A. Sethares. *Software Receiver Design: Build Your Own Digital Communication System in Five Easy Steps*. Cambridge University Press, 2011, p. 417. ISBN: 0521189446.

[9] Nick Collins. "A Comparison of Sound Onset Detection Algorithms with Emphasis on Psychoacoustically Motivated Detection Functions". In: (Jan. 2012).

[10] *Concert Grand 290 Imperial*. 2017. URL: https://www.boesendorfer.com/en/pianos/pianos/Concert-Grand-290-Imperial.

[11] Kyle Coughlin and SkyLeap Music. *Fundamentals of music*. URL: http://www.fundamentalsofmusic.com/tone-color.html.

[12] Marion Bröer Sabine Pfeifer Cristina Bachmann Heiko Bischoff. *Cubase 4 - Operation Manual*. Steinberg Media Technologies GmbH.

[13] M. E. P. Davies and M. D. Plumbley. "Context-Dependent Beat Tracking of Musical Audio". In: *IEEE Transactions on Audio, Speech, and Language Processing* 15.3 (2007), pp. 1009–1020. ISSN: 1558-7916. DOI: 10.1109/TASL.2006.885257.

[14] Simon Dixon. "Automatic Extraction of Tempo and Beat From Expressive Performances". In: *Journal of New Music Research* 30.1 (2001), pp. 39–58. DOI: 10.1076/jnmr.30.1.39.7119. eprint: http://www.tandfonline.

com/doi/pdf/10.1076/jnmr.30.1.39.7119. URL: http://www.tandfonline.com/doi/abs/10.1076/jnmr.30.1.39.7119.

[15] Chris Duxbury, Mark Sandler, and Mike Davies. "A Hybrid Approach To Musical Note Onset Detection". In: (Nov. 2002).

[16] Sibyl Marcuse Edmund Addison Bowles. Percussion instrument. In: *MUSICAL INSTRUMENT*.

[17] Laurence Elliot Libin Edwin M. Ripin Cecil Clutton. Keyboard instrument. In: *MUSIC*.

[18] Blake Eiseman. *The Difference Between Mono and Stereo Files*.

[19] D. P. W. Ellis. *PLP and Rasta and MFCC and inversion in Matlab*. 2005. URL: http://labrosa.ee.columbia.edu/matlab/rastamat/.

[20] Daniel P. W. Ellis. "Beat Tracking by Dynamic Programming". In: *Journal of New Music Research* 36.1 (2007), pp. 51–60. DOI: 10.1080/09298210701653344. eprint: http://dx.doi.org/10.1080/09298210701653344. URL: http://dx.doi.org/10.1080/09298210701653344.

[21] Theodore C. Grame Eric Halfpenny. Stringed instrument. In: *MUSICAL INSTRUMENT*.

[22] Christopher L. Farrow et al. "Nyquist-Shannon sampling theorem applied to refinements of the atomic pair distribution function". In: *Phys. Rev. B* 84 (13 2011), p. 134105. DOI: 10.1103/PhysRevB.84.134105. URL: https://link.aps.org/doi/10.1103/PhysRevB.84.134105.

[23] Hans J. Weber George B. Arfken. *Mathematical Methods for Physicists*. Elsevier Academic Press, 2005. ISBN: 0-12-088584-0.

[24] Masataka Goto. "An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds". In: *Journal of New Music Research*. Vol. 30. Sept. 2002.

[25] F. Hlawatsch and F. Auger. *Time-frequency Analysis: Concepts and Methods*. Digital signal and image processing series. ISTE, 2008. ISBN: 9781905209149.

[26] Apple Inc. *Logic Pro X: Time stretch regions in the Tracks area*. URL: https://support.apple.com/kb/PH13042?locale=en_US.

[27] Theodore C. Grame Jack Allan Westrup. *Musical Instrument*. Encyclopædia Britannica, inc. URL: https://www.britannica.com/art/musical-instrument. December 11, 2013.

[28] Robert Austin Warner James M. Borders Eugene J. Enrico. Keyboard instrument. In: *MUSIC*.

[29] Tristan Jehan. "Creating Music by Listening". PhD thesis. MIT Media Lab, Cambridge, MA, 2005. URL: http://web.media.mit.edu/âĹijtristan/phd/.

[30] A. Klapuri. "Sound onset detection by applying psychoacoustic knowledge". In: *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*. Vol. 6. 1999, 3089–3092 vol.6. DOI: 10.1109/ICASSP.1999.757494.

[31] Anssi Klapuri. “Introduction to Music Transcription”. In: *Signal Processing Methods for Music Transcription*. Ed. by Anssi Klapuri. Ed. by Manuel Davy. 978-0-387-30667-4. New York: Springer, 2006.

[32] Anssi Klapuri and Anssi P. *Musical Meter Estimation and Music Transcription*. Jan. 2003.

[33] R L. Allen. *Signal Analysis : Time, Frequency, Scale, and Structure*. Feb. 2004. ISBN: 0471234419.

[34] Jean Laroche. “Efficient Tempo and Beat Tracking in Audio Recordings”. In: 51 (Apr. 2003), pp. 226–.

[35] Julie LaRoche. *Estimating tempo, swing and beat locations in audio recordings*. Feb. 2001.

[36] Zbigniew Leonowicz, Tadeusz Lobos, and Krzysztof Wozniak. “Analysis of non-stationary electric signals using the S-transform”. In: 28 (Jan. 2009).

[37] Jean Jiang Li Tan. *Digital Signal Processing*. Elsevier Inc, 2013. ISBN: 978-0-12-415893-1.

[38] J Mcauley. “Tempo and Rhythm”. In: *Music Perception*. 2010, pp. 165–199.

[39] B. McFee and D. P. W. Ellis. “Better beat tracking through robust onset aggregation”. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014, pp. 2154–2158. DOI: 10.1109/ICASSP.2014.6853980.

[40] M. F. McKinney et al. “Evaluation of Audio Beat Tracking and Music Tempo Extraction Algorithms”. In: *Journal of New Music Research* 36.1 (2007), pp. 1–16. DOI: 10.1080/09298210701653252. eprint: http://dx.doi.org/10.1080/09298210701653252. URL: http://dx.doi.org/10.1080/09298210701653252.

[41] Holly Day Michael Pilhofer. *Music Theory for Dummies*. 111 River St. Hoboken, NJ 07030-5774: Wiley Publishing, Inc, 2007. ISBN: 978-0-7645-7838-0.

[42] Carlton Gamer Robert A. Moog. Electronic instrument. In: *MUSIC*.

[43] Eric D. Scheirer. “Tempo and beat analysis of acoustic musical signals”. In: *The Journal of the Acoustical Society of America* 103.1 (1998), pp. 588–601. DOI: 10.1121/1.421129. eprint: http://dx.doi.org/10.1121/1.421129. URL: http://dx.doi.org/10.1121/1.421129.

[44] Douglas Self. *Audio Engineering Explained- for professional audio recording*. Focal Press, 2010. ISBN: 978-0-240-81273-1.

[45] C. E. Shannon. “Communication in the Presence of Noise”. In: *Proceedings of the IRE* 37.1 (1949), pp. 10–21. ISSN: 0096-8390. DOI: 10.1109/JRPROC.1949.232969.

[46] S. S. Stevens, J. Volkmann, and E. B. Newman. “A Scale for the Measurement of the Psychological Magnitude Pitch”. In: *The Journal of the Acoustical Society of America* 8.3 (1937), pp. 185–190. DOI: 10.1121/1.1915893. eprint: http://dx.doi.org/10.1121/1.1915893. URL: http://dx.doi.org/10.1121/1.1915893.

[47] R. G. Stockwell, L. Mansinha, and R. P. Lowe. "Localization of the complex spectrum: the S transform". In: *IEEE Transactions on Signal Processing* 44.4 (1996), pp. 998–1001. ISSN: 1053-587X. DOI: 10.1109/78.492555.

[48] Thomas Strohmer. *Local time-frequency analysis and short time Fourier transform*. URL: https://www.math.ucdavis.edu/~strohmer/research/gabor/gaborintro/node3.html.

[49] Jeff Strong. *PC Recording Studios for Dummies*. Wiley Publishing, Inc, 2005, p. 25.

[50] Neil P. McAngus Todd. "The auditory "Primal Sketch": A multiscale model of rhythmic grouping". In: *Journal of New Music Research* 23.1 (1994), pp. 25–70. DOI: 10.1080/09298219408570647. eprint: http://dx.doi.org/10.1080/09298219408570647. URL: http://dx.doi.org/10.1080/09298219408570647.

[51] Charles Van Loan, for Industrial, and Society Applied Mathematics. "Computational frameworks for the fast fourier transform / Charles Van Loan". In: (Jan. 1993).

[52] Berry Wallace. *Structural Functions in Music*. 0-486-25384-8.

[53] Yu-Hsiang Wang. "The Tutorial: S Transform". In: (Jan. 2010).