Musical Tempo Estimation from Audio using Sub-Band Synchrony

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Master of Technology

by

Shreyan Chowdhury

to the

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY,KANPUR

June 2015

CERTIFICATE

It is certified that the work contained in the thesis entitled "*Musical Tempo Estimation from Audio using Sub-Band Synchrony*" by *Shreyan Chowdhury (10327697)* has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

June 2015

Dr. Rajesh M. Hegde
Associate Professor,
Department of Electrical Engineering,
Indian Institute of Technology,
Kanpur-208016.

Abstract

Tempo estimation and onset detection are two important aspects of music information retrieval. Onset detection aims to locate instances in a music audio where there are note onsets or percussive hits, while tempo estimation uses inter-onset intervals and other features to estimate the pace of the musical piece, measured in BPM (beats per minute). Tempo estimation has applications in music production and mixing, music classification, automatic playlist generation, and audio-visual synchronization, among other music technology tasks. Numerous methods have been proposed in literature for tempo estimation with varying accuracies, however, most are error prone and tend to fail for musical styles that do not have a strong, distinct and steady percussive beat going on. This thesis proposes three different approaches to address this issue. The first proposed method uses the fluctuation strength feature to detect the dominant amplitude modulation frequency in the audio and determines tempo based on the same. The remaining two methods detect onsets first, followed by estimating the tempo based on the onset curve. The spectral centroid method detects onsets by calculating the "center of gravity" of the spectrum at each time frame, and the sub-band synchrony method detects onsets by locating frames at which there are coherent changes in the envelopes of different auditory frequency bands. Sub-band synchrony has been shown to provide the best results for tempo estimation among the algorithms tested.

Dedicated to

my mother

who introduced me to music,

my father

who introduced me to engineering,

and my brother

who has always kept me motivated.

Acknowledgements

I would first like to express my sincere gratitude towards my thesis supervisor, Dr. Rajesh M Hegde, for his guidance and support. He has been a motivating mentor, always eager to show me a direction or a new insight at times when I got stuck, and helping me develop my interests in music signal processing. I am grateful for his patience and his belief in me to pursue a research topic of my liking. I thank him for being my guide throughout and giving my thesis the required direction. I am also grateful to Dr. Preeti Rao for giving me the opportunity to visit her lab at IIT-Bombay and learn about the fascinating field of music and audio processing, that eventually helped me gain a bird's-eye-view of the field.

My 5 year long stay at IIT Kanpur would not have been the same without the support and companionship of my friends, especially Nitica, Venkat, Vipul, Sankalp, Pratul, Manav, and Puneet with whom I have shared some of the best years of my life. I would also like to thank Devanshu for being a helping hand in times of trouble, and the members of MiPS lab who helped me in one way or the other in completing this work.

Finally, I express my cordial honour to my caring parents for their unconditional support and encouragement and for bringing me to this stage of my life. I am indebted to them for their endless love, inspiration and care.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Without music, life would be a mistake.

Friedrich Nietzsche

Music as an art form has been a part of human existence for thousands of years. Based on paleolithic archaeological findings of ancient musical instruments, music can be theorized to pervade human life even in prehistoric periods, before civilization began. From those roots till the current time, it has stood its ground, and it now finds itself in the midst of the twenty-first century population with an exponentially increasing trend of consumption of music.

Listening to music today has become a more personal experience than ever before. With the advent of portable music players, and personalised music recommendation software, we are becoming used to listening to music on demand. We no longer listen to whatever comes up on the radio; we feel more comfortable when we can choose our music according to our mood or the ambience or the activity we are doing. This creates a necessity for music service providers to analyze, classify, modify and play music in an unprecedented way. Additionally, the explosive expansion of music production due to affordability of personal computers is creating a massive amount of music-related data, and new technologies are needed to store all of that in a manageable fashion, and leverage the enormous data for the benefit of people by making their listening experience better and better.

Music Information Retrieval (MIR) is the interdisciplinary science that has emerged to

address this issue. It is a growing field of research which spans people who have background in musicology, psychology, academic music study, signal processing, machine learning or some combination of these. MIR involves applications like chord recognition, music classification, genre detection, automatic music transcription, music generation, and beat tracking, among many others [3].

1.1 Motivation

When we listen to music, we experience an auditory perception which can be said to comprise broadly of what we call melody, harmony, rhythm and timbre. Rhythm, in its most generic sense, is used to refer to all of the temporal aspects of a musical work, whether it is represented in a score, measured from a performance, or existing only in the perception of the listener, whereas melody and harmony are more pitch-related. Timbre, also known as tone color or tone quality from psychoacoustics, is the quality of a musical note, sound, or tone that distinguishes different types of sound production, such as voices and musical instruments, string instruments, wind instruments etc.

Among these, rhythm has played an interesting role in the evolution of music in humans. There are theories that human rhythm recalls the regularity with which we walk and the heartbeat. Other research suggests that it does not relate to the heartbeat directly, but rather the speed of emotional affect, which also influences heartbeat. Yet other researchers suggest that since certain features of human music are widespread, it is reasonable to suspect that beat-based rhythmic processing has ancient evolutionary roots. Justin London in [4] writes that musical metre "involves our initial perception as well as subsequent anticipation of a series of beats that we abstract from the rhythm surface of the music as it unfolds in time". The "perception" and "abstraction" of rhythmic measure is the foundation of human instinctive musical participation, as when we divide a series of identical clock-ticks into "tick-tock-tick-tock".

As music evolved and differentiated into various genres and styles, their rhythms also evolved, and thus rhythmic structure has become a good metric to distinguish between them. For example, the tabla rhythm patterns in Hindustani music are in stark contrast to drums

in rock or jazz genres, and the Clave rhythm pattern immediately reminds one of Latin music and associated tango and salsa dance forms. Thus it becomes instructive to include the study of rhythms as one of the cornerstones of Music Technology.

In this regard, two of the long standing problems in MIR are rhythm pattern extraction and tempo induction from audio. Music typically involves multiple and concurrent layers of rhythmic activity, and as such simultaneously affords different modes of temporal engagement by listeners and performers [4]. Among those layers, one layer is often thought to be perceptually dominant, serving as the temporal referent [5]. Musicians refer to this layer as the beat or tactus, and musical tempo is typically expressed in beats per minute (BPM); as such, BPM is often regarded as a transparent measure of musical speed [6] [7]. Likewise, tapping behaviours where one taps at the tactus rate are often regarded as reliable indicators of tempo perception [8] [9] [10].

Thus, tempo being one of the basic components of rhythm, it is important to develop a methodology for accurate estimation of it before delving into deeper levels of rhythmic pattern extraction and metrical rhythm transcription.

1.2 Literature Survey

The essence of beat tracking is that tapping to a music is a quite automatic and subconscious task for most humans. However, the same is not true for computers; replicating this process algorithmically has been an active area of research for well over twenty years, with reasonable success achieved only recently. This section provides a quick glance of the different works in this area. Before proceeding, it might be prudent to define some terms that will be encountered frequently in this thesis.

- **Rhythm:** A strong, regular repeated pattern of movement or sound. In music, rhythm is often expressed by a periodic variation in instantaneous sound energy, or a periodic variation in pitch.

- **Tempo:** In musical terminology, tempo ("time" in Italian; plural: tempi) is the speed or pace of a given piece or subsection thereof. A piece of music's tempo is typically written at the start of the score, and in modern Western music is usually indicated

in beats per minute (BPM). This means that a particular note value is specified as the beat, and that the amount of time between successive beats is a specified fraction of a minute. The greater the number of beats per minute, the smaller the amount of time between successive beats, and thus faster a piece must be played. For example, a tempo of 60 beats per minute signifies one beat per second, while a tempo of 120 beats per minute is twice as rapid, signifying one beat every 0.5 seconds. In Hindustani and Carnatic music, tempo is referred to as *Laya*.

- **Tactus:** Same as *beat*. Klapuri [2] describes the beat or tactus as the preferred (trained) human tapping tempo and is what most of the beat-tracking algorithms attempt to extract at a minimum. This usually corresponds to the 1/4 note or crotchet when written out in common notation, though this is not always the case.

- **Tatum:** At a lower level than the beat is the tatum, which is defined to be the shortest commonly occurring time interval. This is often defined by the l/8th notes (quavers) or l/16th notes (semiquavers) [2].

- **Measure or bar:** Conversely, the main metrical level above the beat is that of the bar or measure. This is related to the rate of harmonic change within the piece, usually to a repeated pattern of emphasis and also notational convention. Figure 1.1 gives a diagrammatic representation of the above discussion.

- **Harmonic Sounds:** Those sounds in the musical piece that are played and heard for a relatively prolonged period of musical time, and are perceived as consonant or dissonant depending on the pitch of that and accompanying sounds. Pitch is a characteristic feature of these kind of sounds. Instruments that are harmonic in nature are violin, sarangi, trumpet, accordian, guitar, piano, etc. However, it may be noted that the transients of plucked or hammered instruments (like guitar and piano) are considered percussive in nature. In other words, the sound heard just after plucking a string or hitting a key is a high-energy one, and is considered percussive in nature.

- **Percussive Sounds:** Those sounds in the musical piece that are played and heard as short, high-energy hits. Instruments that produce these sounds are drums, tabla,

congo, transients of plucked or hammered instruments like guitar and piano. Percussive sounds play a major role in the sensation of rhythm in a musical piece.

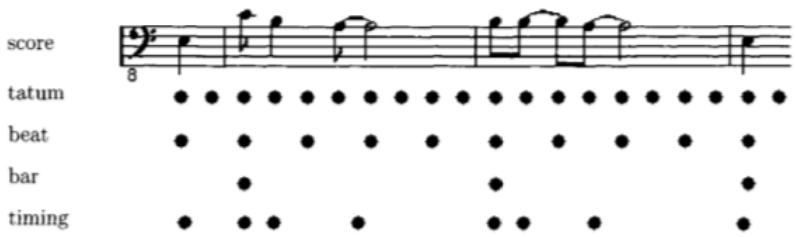

Figure 1.1: Image courtesy Klapuri [2]. Diagram of relationships between metrical levels.

1.2.1 Onset Detection

While the metre and tempo of a piece of music can be thought of as constantly evolving signals, the musical events which underpin this are the starts of notes, and these are discrete events. It is highly possible, and indeed common, to simply attach an onset detector to find the note starts in an audio signal and then track the resulting set of discrete impulses. Percussive sounds are usually characterized by significant increases in signal energy (a 'transient') and methods for detecting this type of musical sound are relatively well developed. Harmonic change with little associated energy variation is much harder to reliably detect and has received less attention in the literature. Two recent studies of onset detection are those of Bello [11] and Dixon [12].

1.2.1.1 Transient Event Detection

Transient events, such as start of notes with significant energy change (e.g. piano, guitar), or drum sounds, can be detected by analyzing the signal amplitude envelope. This basic approach can be described as follows. The amplitude envelope is extracted from the audio, and then peak picking is performed to give the onset positions as outputs.

A different approach is taking the energy envelope function $E(t)$ formed by summing the

power of frequency components in the spectrogram for each time slice over the range required:

$$E_j(n) = \sum_{k \in \kappa_j} |STFT_x^w(n,k)|^2 \tag{1.1}$$

where $STFT_x^w(n,k)$ is the short-time Fourier transform (STFT) of the signal $x(n)$ with rectangular window w centred at time n; k is the frequency index.

1.2.1.2 Pitched Event Detection

Detection of note starts where there is no associated energy transient (e.g. violins, choral music) has received less attention than the easier problem addressed above. Notable recent exceptions are Laurent [13], who used wavelets; Davy [14], who took a support vector machine approach; Desobry [15], who furthered Davy's research and also used kernel methods; and Abdallah [16], who used independent component analysis (ICA) to generate a 'surprise' measure followed by an HMM to perform reliable detection.

1.2.2 Auto-correlation Methods

Auto-correlation is a method for finding periodicities in data and has hence been used in several studies. Without subsequent processing, it can only find tempo and not the beat phase. The basic approach is to define an energy function $E(n)$ to which local auto-correlation is then applied (in frames of length T_w, centred at time n). The value of i which maximizes $r(n,i)$ should correspond to the period-length of a metrical level [17].

1.2.3 Oscillating Filter Approaches

There are two distinct approaches using oscillating filters: In the first, an adaptive oscillator is excited by an input signal and, hopefully, the oscillator will resonate at the frequency of the beat. The second method uses a bank of resonators at fixed frequencies which are exposed to the signal and the filter with the maximum response is picked for the tempo. Beat location can be calculated by examining the phase of the oscillator [18]. The observed signal is a set of impulses $s(n) = 1$ when there is an onset event and $s(n) = 0$ otherwise. The oscillator is given by

$$o(n) = 1 + \tanh \alpha(\cos 2\pi\phi(n) - 1) \tag{1.2}$$

where $o(n)$ defines an output waveform with pulses at beat locations with width tuned by α. The phase is given by

$$\phi(n) = \frac{n - n_i}{p} \tag{1.3}$$

where n_i is the location of the previous beat and p is the period of oscillation (inverse of tempo).

1.2.4 Energy Flux

Frequency-domain processing allows detecting onsets of much lower energy than other continuous signals in the audio [19]. The signal is analyzed using a short-time Fourier transform. Denoting by $x(n)$ the signal, t_i the frame time in seconds, F_s the sampling frequency, and N the size in samples of the analysis window $w(n)$, the short-time Fourier transform $X(f, t_i)$ at the normalized frequency f and frame i is

$$X(f, t_i) = \sum_{n=0}^{N-1} w(n)x(n + F_s t_i)e^{-2\pi j f n} \tag{1.4}$$

A suitable value for the window size is about 10 ms, and the hop size (the interval between two successive FFT analyses $t_{i+1} - t_i$) can be set to 10 ms (no overlap). In the following steps only the magnitude of the FFT is used. Since our goal is to locate areas in time and frequency where there is a sudden energy increase, a first-order difference from frame to frame is then calculated on the result. The results for all the bins are summed together, and the result is half-wave rectified to obtain a positive energy flux signal $E(i)$, which exhibits sharp maxima at transients and note onsets.

$$\hat{E}(i) = \sum_{f=f_{min}}^{f_{max}} G(|X(f, t_i)|) - G(|X(f, t_{i-1})|) \tag{1.5}$$

where $G(x)$ is a non-linear monotonic compression function to emphasize high frequencies.

$$E(i) = \begin{cases} \hat{E}(i) & \hat{E}(i) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{1.6}$$

where f_{min} and f_{max} control the range of frequencies over which the summation is carried out (typically from 100 Hz to 10 kHz). This signal is used in the subsequent stage to select tempo.

1.2.5 Median Filtering

Fitzgerald, in [20] uses median filtering to separate percussive events from non-percussive ones and thus detecting percussive or transient onsets. The motivation comes from looking at harmonic events as outliers in the frequency spectrum at a given time frame, and to regard percussive events as outliers across time in a given frequency bin. This brings us to the concept of using median filters individually in the horizontal and vertical directions to separate harmonic and percussive events. For odd l, median filter can be defined as

$$y(n) = \text{median}\,\{x(n-k:n+k),\ k=(l-1)/2\} \tag{1.7}$$

As opposed to moving average filters, median filters are effective in removing impulse noise because they do not depend on values which are outliers from the typical values in the region around the original sample.

Given an input magnitude spectrogram S,and denoting the i-th time frame as S_i, and the h-th frequency slice as S_h a percussion enhanced spectrogram frame P_i can be generated by performing median filtering on S_i:

$$P_i = \mathfrak{M}(S_i, l_{\text{perc}}) \tag{1.8}$$

where $\mathfrak{M}$ denotes median filtering and l_{perc} is the filter length of the percussion-enhancing median filter. The individual percussion enhanced frames P_i are then combined to yield a percussion-enhanced spectrogram $\mathbf{P}$. Standard envelope-based onset detection is then applied to $\mathbf{P}$ to obtain the onset curve and calculate tempo.

1.3 Contribution of Thesis

- A detailed study and analysis of different methods for onset detection for rhythm extraction and tempo estimation purposes has been done.
- Onset detection through temporal methods like amplitude modulation and spectral methods like MFCCs, spectral centroid, and finally sub-band synchrony are carried out.

- Three new tempo estimation methods are proposed: using fluctuation, using spectral centroid, and using sub-band synchrony. Spectral centroid and sub-band synchrony are novel techniques for onset detection as well.

- Experiments and detailed analyses are done for tempo estimation using the three proposed and two existing methods of tempo estimation on various datasets containing songs of different genre each. The efficacy of each algorithm on different musical styles is studied in this fashion.

1.4 Organization of the Thesis

The organization of the thesis is as follows.

Chapter 2 discusses in detail various different existing methods of onset detection. A general scheme of onset detection is presented along with recent developments and algorithms for achieving more accurate onset detection curves. Examples of onset detection curves for the algorithms are also presented.

Chapter 3 presents tempo estimation methods using fluctuation and spectral centroid. The fluctuation and spectral centroid features have been explained in detail and the motivation and intuition behind using these features for tempo estimation have been elaborated.

Chapter 4 presents tempo estimation using sub-band synchrony. The procedure is explained along with plots to demonstrate graphically the process of onset detection using sub-band synchrony and subsequently tempo calculation.

Chapter 5 discusses experiments conducted to test the efficacy of the tempo estimation algorithms. Dataset generation process, ground truth establishment process and genre wise testing of algorithms have been presented.

Chapter 6 contains conclusions of this thesis and future scope.

Chapter 2

Overview of Existing Onset Detection Algorithms

Tempo induction requires the accurate detection of onsets of musical tones, and several different methods can be found in literature for onset detection. Onset detection is a well-defined task at first sight with the aim being to find the starting time of each musical note (where a musical note is not restricted to those having a clear pitch or harmonic partials). However, in polyphonic music, where nominally simultaneous notes (chords) might be spread over tens of milliseconds, the definition of onsets starts to become blurred. Likewise, instruments with long attack times (e.g. flute) produce notes for which it is difficult to define an unambiguous and precise onset time [12].

2.1 General Scheme of Onset Detection

In the more realistic case of a polyphonic signal with added noise, where multiple instruments and sound sources may be present at a given time, the above distinctions become less precise. Audio signals are both additive (musical objects in polyphonic music superimpose and not conceal each other) and oscillatory. Therefore, it is not possible to look for changes simply by differentiating the original signal in the time domain; this has to be done on an intermediate signal that reflects, in a simplified form, the local structure of the original. In this thesis, such a signal is referred to as a detection function; in the literature, the term novelty function

is sometimes used instead [21]. Figure 2.1 illustrates the procedure employed in the majority of onset detection algorithms: from the original audio signal, which can be pre-processed to improve the performance of subsequent stages, a detection function is derived at a lower sampling rate, to which a peak-picking algorithm is applied to locate the onsets. Whereas peak-picking algorithms are well documented in the literature, the diversity of existing approaches for the construction of the detection function makes the comparison between onset detection algorithms difficult for audio engineers and researchers [11].

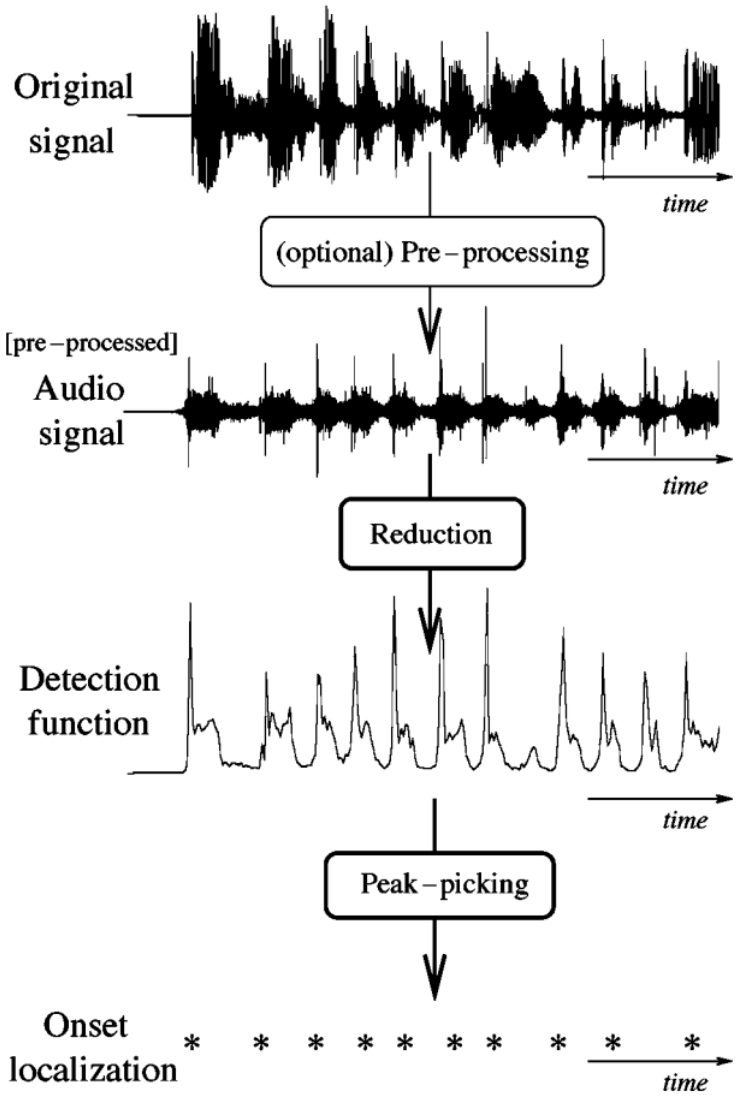

Figure 2.1: Image courtesy: Bello [11]. Flowchart of standard onset detection algorithm.

2.1.1 Pre-processing Step for Onset Detection

There are a number of different treatments that can be applied to a musical signal in order to facilitate the task of onset detection. However, two processes that are consistently mentioned in the literature, and that appear to be of particular relevance to onset detection schemes are described here.

1. **Multiple Bands:** Several onset detection studies have found it useful to independently analyze information across different frequency bands. In some cases this pre-processing is needed to satisfy the needs of specific applications that require detection in individual sub-bands to complement global estimates; in others, such an approach can be justified as a way of increasing the robustness of a given onset detection method.

2. **Transient/Steady-State Separation:** The onset detection problem can be treated as a source-separation problem when transients are considered as a different source than steady state. This is called Harmonic-Percussive Source Separation (HPSS). Fitzgerald [20] tackled this problem by considering the overall power spectrum to be distributed into the harmonic and the percussive components and estimating each component by a median filtering approach. Other methods include sinusoidal modeling approaches. Amongst these methods, spectral modeling synthesis (SMS) [22] explicitly considers the residual of the synthesis method as a Gaussian white noise filtered with a slowly varying low-order filter. Levine [23] calculates the residual between the original signal and a multi-resolution SMS model. Significant increases in the energy of the residual show a mismatch between the model and the original, thus effectively marking onsets.

2.1.2 Reduction Step for Onset Detection

In the context of onset detection, the concept of reduction refers to the process of transforming the audio signal into a highly subsampled detection function which manifests the occurrence of transients in the original signal.

1. Reduction Based on Signal Features

   (a) *Temporal Features:* By observing the temporal evolution of simple musical signals,

it is noticed that an increase of the signals amplitude accompanies the occurrence of an onset. Early methods of onset detection capitalized on this by using a detection function which follows the amplitude envelope of the signal [24]. Such an envelope follower can be easily constructed by rectifying and smoothing (i.e., low-pass filtering) the signal:

$$E_o(n) = \frac{1}{N} \sum_{m=-\frac{N}{2}}^{\frac{N}{2}-1} |x(n+m)|\, w(m) \tag{2.1}$$

where $w(n)$ is an n-point window or smoothing kernel, centered at $m = 0$. This gives satisfactory results for some applications where strong percussive transients exist against a relatively quiet background. A variant of this is to follow the local energy, rather than the amplitude, by squaring, instead of rectifying, each sample.

$$E_o(n) = \frac{1}{N} \sum_{m=-\frac{N}{2}}^{\frac{N}{2}} [x(n+m)]^2\, w(m) \tag{2.2}$$

(b) *Spectral Features:* A number of techniques have been proposed that use the frequency domain representation of the signal to produce more reliable detection functions. In the spectral domain, energy increases linked to transients tend to appear as a broadband event. Since the energy of the signal is usually concentrated at low frequencies, changes due to transients are more noticeable at high frequencies [25]. To emphasize this, the spectrum can be weighted preferentially toward high frequencies before summing to obtain a weighted energy measure.

$$\tilde{E}(n) = \frac{1}{N} \sum_{m=-\frac{N}{2}}^{\frac{N}{2}} W_k(n)\, |X_k(n)|^2 \tag{2.3}$$

where W_k is the frequency dependent weighting. Masri [26] proposes a high frequency content (HFC) function with $W_k = |k|$, linearly weighting each bins contribution in with the weights being proportional to its frequency. The HFC function produces sharp peaks during attack transients and is notably successful when faced with percussive onsets, where transients are well modeled as bursts of white noise. A more general approach based on changes in the spectrum measure the distance

between successive short-term Fourier spectra, treating them as points in an N-dimensional space, and formulate this as a detection function. Depending on the metric chosen to calculate this distance, spectral difference or spectral flux detection functions can be constructed. As an example, the L_2-norm metric is taken as

$$SD(n) = \sum_{k=1}^{\frac{N}{2}-1} \{H(|X_k(n)| - |X_k(n-1)|)\}^2 \tag{2.4}$$

where $H(x) = \frac{x+|x|}{2}$, i.e., zero for negative arguments. The rectification has the effect of counting only those frequencies where there is an increase in energy, and is intended to emphasize onsets rather than offsets.

(c) *Spectral Features using Phase:* Using phase spectra for onset detection instead of magnitude is relevant since much of the temporal structure of a signal is encoded in the phase spectrum. Let us define $\phi_k(n)$ the 2π-unwrapped phase of a given STFT coefficient $X_k(n)$. The instantaneous frequency $f_k(n)$, an estimate of the actual frequency of the STFT component within a window, is calculated as

$$f_k(n) = \left(\frac{\phi_k(n) - \phi_k(n-1)}{2\pi h} \right) f_s \tag{2.5}$$

where h is the hop size between windows and f_s is the sampling frequency. It is expected that, for a locally stationary sinusoid, the instantaneous frequency should be approximately constant over adjacent windows. Thus, according to 2.5, this is equivalent to the phase increment from window to window remaining approximately constant.

$$\phi_k(n) - \phi_k(n-1) \simeq \phi_k(n-1) - \phi_k(n-2) \tag{2.6}$$

Equivalently, the phase deviation can be defined as the second difference of the phase.

$$\Delta\phi_k(n) = \phi_k(n) - 2\phi_k(n-1) + \phi_k(n-2) \simeq 0 \tag{2.7}$$

During a transient region, the instantaneous frequency is not usually well defined, and hence $\Delta\phi_k(n)$ will tend to be large. This is used as an onset detection function. However, this method is susceptible to phase distortion and to noise introduced by the phases of components with no significant energy.

(d) *Complex Domain:* Amplitude and phase can be considered jointly to search for departures from steady-state behaviour by calculating the expected amplitude and phase of the current bin $X(n,k)$, based on the previous two bins $X(n1,k)$ and $X(n2,k)$. The target value $X_T(n,k)$ is estimated by assuming constant amplitude and rate of phase change [12]

$$X_T(n,k) = |X(n-1,k)|e^{\psi(n-1,k)-\psi'(n-1,k)} \tag{2.8}$$

and therefore a complex domain onset detection function CD can be defined as the sum of absolute deviations from the target values:

$$CD = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} |X(n,k) - X_T(n,k)| \tag{2.9}$$

(e) *Mel-Frequency Cepstral Coefficients:* Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC. They are derived from a type of cepstral representation of the audio clip (a nonlinear "spectrum-of-a-spectrum"). The difference between the cepstrum and the mel-frequency cepstrum is that in the MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum. This frequency warping can allow for better representation of sound, for example, in audio compression.

MFCCs are commonly derived as follows:

- Take the Fourier transform of (a windowed excerpt of) a signal.
- Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows.
- Take the logs of the powers at each of the mel frequencies.
- Take the discrete cosine transform of the list of mel log powers, as if it were a signal.
- The MFCCs are the amplitudes of the resulting spectrum.

Let $\mathbf{M}(k,n)$ denote the MFCC matrix where $k = 1,2,...,40$ are the 40 MFCC coefficients and $n = 1,2,..,N$ are the number of windowed frames in the music

signal. A threshold function is applied to this matrix that extracts coefficients greater than the threshold value, in this case, 80 dB. This is done to suppress spurious peaks in $\mathbf{M}$. Next, the derivative of the matrix $\mathbf{M}$ is taken across time for all coefficients.

$$\mathbf{M}'(k,n) = \frac{\partial \mathbf{M}(k,n)}{\partial n}, \quad k = 1, 2, ..., 40 \tag{2.10}$$

This gives high values at locations where there are sudden changes in the MFCC domain. The mean of all the coefficients for each time frame is taken to give the final onset curve:

$$m(n) = \frac{1}{40} \sum_{k=1}^{40} \mathbf{M}'(k,n) \tag{2.11}$$

2. Reduction Based on Probability Models

   (a) *Calculation of Surprise Signals:* This method looks for "surprising" events relative to a single global model. To this end, a detection function is defined as the moment-by-moment trace of the negative log-probability of the signal given its recent history, according to a global model. A dynamically evolving measure of surprise, or novelty, is used as a detection function [11]. Let us consider the signal as a multivariate random process where each vector $\mathbf{x}(n) \in \mathbb{R}^N$ is a frame of audio samples. At time n , an observers expectations about $\mathbf{x}(n)$ will be summarized by the conditional probability according to that observers model: $p(\mathbf{x}(n)|\mathbf{x}(n-1), \mathbf{x}(n-2)...)$. When $\mathbf{x}(n)$ is actually observed, the observer will be surprised to a certain degree, which we will define as

$$S(n) \equiv S(\mathbf{x}(n)) = -\log p(\mathbf{x}(n)|\{\mathbf{x}(j) : j < n\}) \tag{2.12}$$

This is closely related to the entropy rate of the random process, which is simply the expected surprise according to the true model.

2.2 Examples of Onset Detection

2.2.1 Onset Detection using Temporal Features

Percussive and transient sounds, which yield the perception of rhythm, have a short-time high-energy content. This is the intuition behind using amplitude envelope as a marker of onsets in an audio stream. The signal's envelope is equivalent to its outline and an envelope detector connects all the peaks in this signal. An example of an audio signal and its envelope is shown in Figure 2.2.

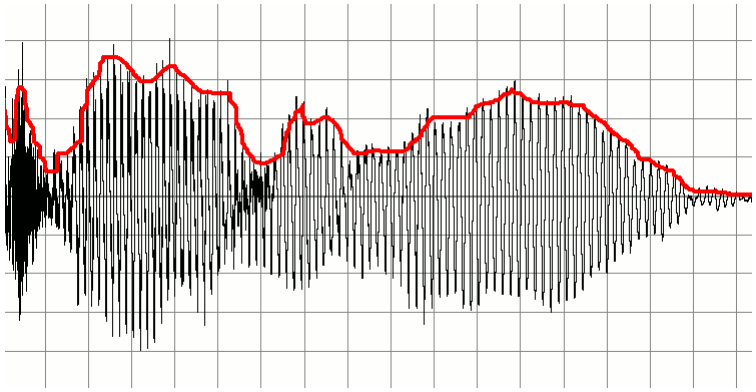

Figure 2.2: Envelope (in red) of an audio signal.

Percussive sounds such as drum sounds have high energy content. In audio signals containing clear and distinct percussive sounds, onset detection is an easy task using envelope detection and peak picking. Figure 2.3 shows onsets detected (in red dots) and ground truth values (in black dots) of an audio signal with a distinct drum track (excerpt of *Stayin Alive* by the Bee Gees). It can be seen that the detected onsets and the ground truth onsets are very close to each other and there are no false positives or false negatives in this short extract.

Amplitude envelope is computed by squaring the signal and low pass filtering the output. The general methodology is given in Figure 2.4. However, it must be noted that this scheme fails when the onsets are not high-energy percussive sounds. For example, when the audio signal is that of an organ, the onsets as annotated by human subjects are at the note change times. However, using the present scheme of onset detection, that cannot be reliably extracted. This is shown in Figure 2.5.

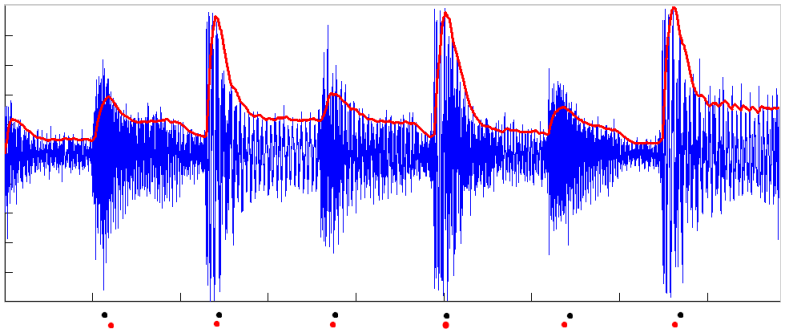

Figure 2.3: Envelope (in red) of an audio signal with distinct drum sounds. The peaks of the onset curve are shown as red dots and the actual onsets are marked with black dots.

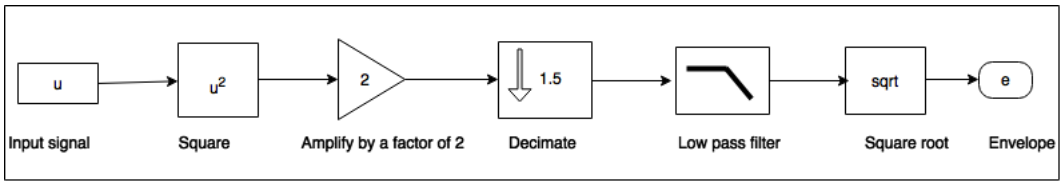

Figure 2.4: Amplitude envelope detection scheme.

2.2.2 Onset Detection using Spectral Features

Section 1b described five spectral features that are typically used for onset detection. Here, examples of onset detection using the MFCC features and using spectral flux are elaborated. Figure 2.6 shows 40 MFCC coefficients with time for an input music signal. Its derivative shown in Figure 2.7. Next, the mean of the coefficient values at each time instant is taken to obtain the final onset curve in Figure 2.8.

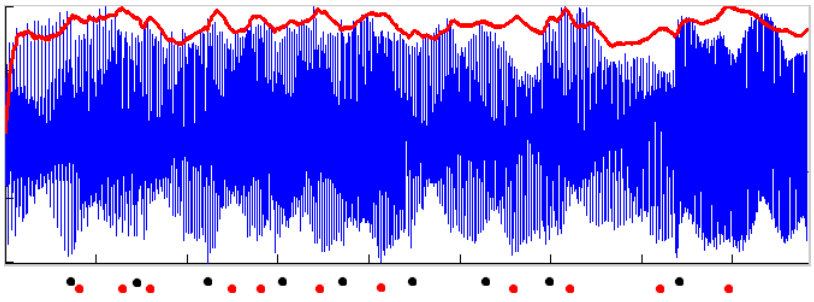

Figure 2.5: Spurious results in audio with note changes but no percussive sounds. Red dots indicate detected onset times, black dots show annotations by human subjects.

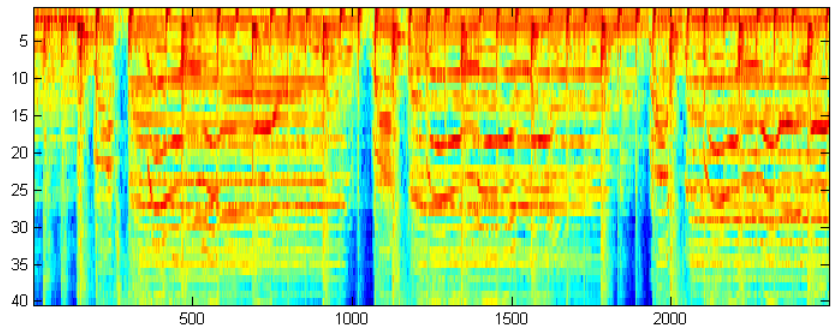

Figure 2.6: MFCC coefficient magnitude with respect to time

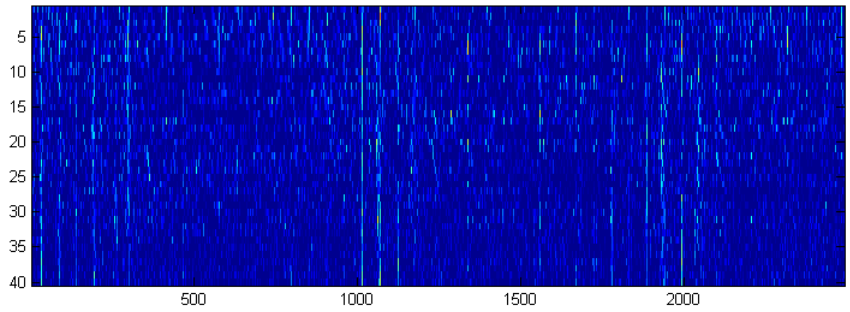

Figure 2.7: Derivative of MFCC coefficients as function of time

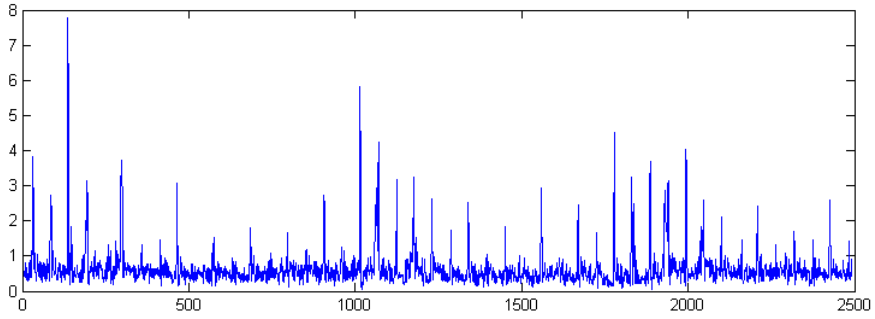

Figure 2.8: Onset curve generated by taking the mean of the derivative of MFCC matrix across all coefficients for each time frame

Figure 2.9 shows the onset curve obtained from calculation of spectral flux.

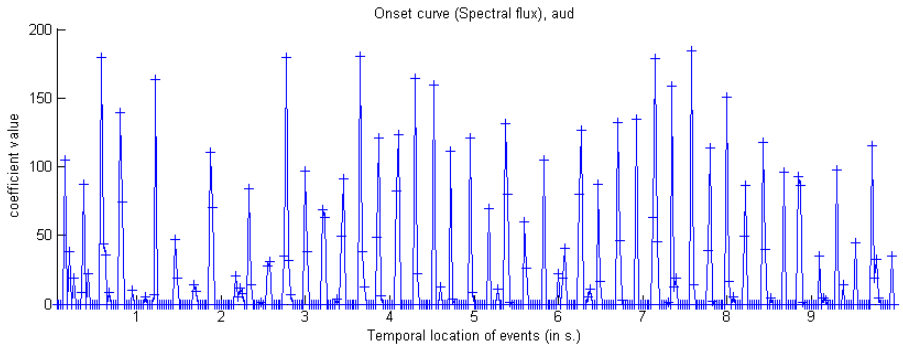

Figure 2.9: Onset curve generated using spectral flux

2.3 Summary

In this chapter, a detailed introduction to methodologies involved in onset detection was presented. The procedure was sequentially described in two stages: preprocessing and reduction. Preprocessing is done to facilitate onset detection in the subsequent steps. Separating into multiple bands and transient/steady state separation techniques were explained in this section. Reduction is the actual onset detection step. Various features, both temporal and spectral were described that are used to detect novelties in the audio, resulting in onsets.

Chapter 3

Tempo Estimation using Fluctuation Strength Feature and Spectral Centroid Feature

In this chapter, two different novel algorithms for tempo estimation are presented. The first of these is the fluctuation method, which computes the overall *fluctuation pattern* for the audio input and estimates the frequency at which the envelope resonates with maximum amplitude after scaling by the prior probability function of tempo frequencies. The second algorithm is based on an onset detection approach. Onsets are detected using a spectral feature called spectral centroid, which is a weighted average of frequency bins weighted by their magnitudes in the spectrogram.

3.1 Computation Scheme for Fluctuation Strength Feature

The loudness of a critical-band usually rises and falls several times. Often there is a periodical pattern, providing the sense of rhythm. At every beat the loudness sensation rises. The loudness values of a critical-band over a certain time period can be regarded as a signal that has been sampled at discrete points in time. The periodical patterns of this signal can then be assumed to originate from a mixture of sinusoids. These sinusoids modulate the amplitude

of the loudness, and can be calculated by a Fourier transform.

The amplitude modulation of the loudness has different effects on our sensation depending on the frequency. The sensation of fluctuation strength is most intense around 4Hz and gradually decreases up to a modulation frequency of 15Hz (cf. Figure 3.4) [27] [28]. At 15Hz the sensation of roughness starts to increase, reaches its maximum at about 70Hz, and starts to decreases at about 150Hz.

Before describing fluctuation in detail, it is necessary to define a few terms.

3.1.1 Definitions

3.1.1.1 Critical Bands

The inner ear can be regarded as a complex system of a series of band-pass filters with an asymmetrical shape of frequency response. Through several psychoacoustic experiments, the center frequencies and the bandwidths of these bands have been ascertained. While we can distinguish low frequencies of up to about 500Hz well, our ability decreases above 500Hz with approximately a factor of $0.2f$, where f is the frequency. This is shown in experiments using a loud tone to mask a more quiet one. At high frequencies these two tones need to be rather far apart regarding their frequencies, while at lower frequencies the quiet tone will still be noticeable at smaller distances. In addition to these masking effects the critical-bandwidth is also very closely related to just noticeable frequency variations. Within a critical-band it is difficult to notice any variations. This can be tested by presenting two tones to a listener and asking which of the two has a higher or lower frequency. Critical bands have been assigned the unit *Bark*.

A critical-band value is calculated by summing up the values of the power spectrum within the respective lower $f_a(i)$ and upper $f_b(i)$ frequency limits of the i-th critical-band. This can be formulated as

$$\mathbf{B}(i,t) = \sum_{n \in I(i)} \mathbf{P}(n,t), \quad I(i) = \{n | f_a(i) < f_{res}(n-1, 256, 1/f_s) \leq f_b(i)\} \tag{3.1}$$

where i, t, n are indexes and $\mathbf{B}$ is a matrix containing the power within the i-th critical-band at a specific time interval t. $\mathbf{P}$ is the matrix representing the power per frequency and time

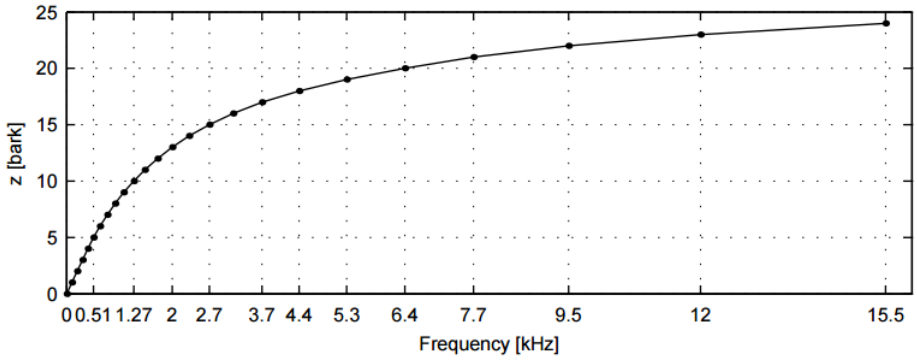

Figure 3.1: Image courtesy: Pampalk [29]. The basic characteristics of the critical-band rate scale. Two adjoining markers on the plotted line indicate the upper and lower frequency borders for a critical band. For example, the 24th band starts at 12kHz and ends at 15.5kHz.

interval t. The relation between actual frequency and the row index n of $\mathbf{P}$ can be obtained using $f_{res}(n1, 256, 1/f_s)$, where

$$f_{res}(n, N, \Delta t) \equiv \frac{n}{N\Delta t}, \quad n = 0, ..., \frac{N}{2} \tag{3.2}$$

3.1.1.2 Phon

The relationship between the sound pressure level in decibel and our hearing sensation measured in sone is not linear. The perceived loudness depends on the frequency of the tone. Figure 3.2 shows so-called loudness levels for pure tones, which are measured in phon. The phon is defined using the 1kHz tone and the decibel scale. For example, a pure tone at any frequency with 40 phon is as loud as a pure tone with 40dB at 1kHz.

The loudness matrix in phon, $\mathbf{L}_{phon}$, can be calculated using the equal loudness contour matrix $\mathbf{C}_{elc}$ and the corresponding phone values to each contour $\mathbf{c}_{phon} = [3, 20, 40, 60, 80, 100]$. $\mathbf{C}_{elc}(i, j)$ contains the decibel values of the j-th loudness contour at the i-th critical-band.

z	f_a, f_b	f_c	z	f_Δ	z	f_a, f_b	f_c	z	f_Δ
Bark	Hz	Hz	Bark	Hz	Bark	Hz	Hz	Bark	Hz
0	0				12	1720			
		50	0.5	100			1850	12.5	280
1	100				13	2000			
		150	1.5	100			2150	13.5	320
2	200				14	2320			
		250	2.5	100			2500	14.5	380
3	300				15	2700			
		350	3.5	100			2900	15.5	450
4	400				16	3150			
		570	4.5	110			3400	16.5	550
5	510				17	3700			
		570	5.5	120			4000	17.5	700
6	630				18	4400			
		700	6.5	140			4800	18.5	900
7	770				19	5300			
		840	7.5	150			5800	19.5	1100
8	920				20	6400			
		1000	8.5	160			7000	20.5	1300
9	1080				21	7700			
		1170	9.5	190			8500	21.5	1800
10	1270				22	9500			
		1370	10.5	210			10500	22.5	2500
11	1480				23	12000			
		1600	11.5	240			13500	23.5	3500
12	1720				24	15500			
		1850	12.5	280					

Table 3.1: Critical-band rate z, lower (f_a) and upper (f_b) frequency limits of the critical bandwidths, f_Δ, centered at f_c.

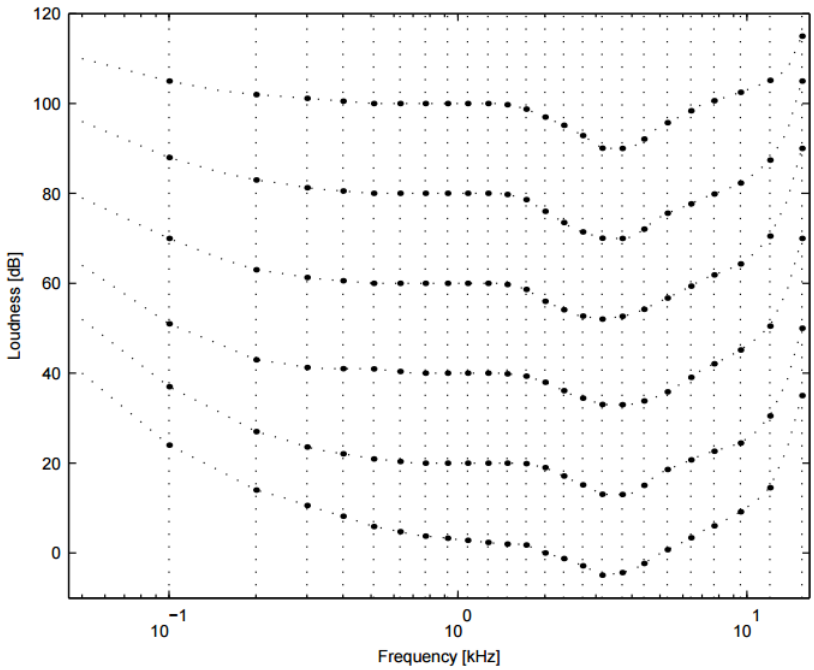

Figure 3.2: Image courtesy: Pampalk [29]. Equal loudness contours for 3, 20, 40, 60, 80 and 100 phon. The respective sone values are 0, 0.15, 1, 4, 16 and 64 sone. The dotted vertical lines indicate the positions of the center frequencies of the critical-bands. The dip around 2kHz to 5kHz corresponds to the frequency spectrum we are most sensitive to.

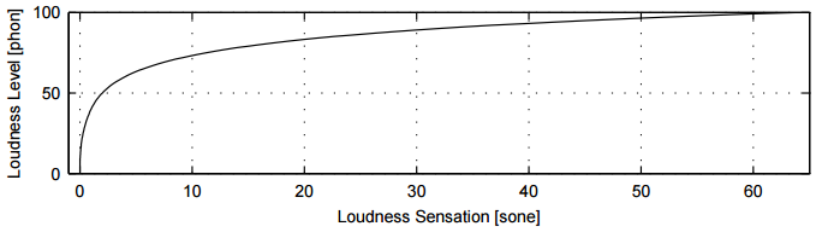

Figure 3.3: Image courtesy: Pampalk [29]. The relationship between the loudness level and the loudness sensation.

3.1.1.3 Sone

Finally, from the loudness level $\mathbf{L}_{phon}$ the specific loudness sensation $\mathbf{L}_{sone}$ per critical band is calculated as,

$$\mathbf{L}_{sone}(i,t) = \begin{cases} 2^{\frac{1}{10}(\mathbf{L}_{phon}(i,t)-40)} & \text{if } \mathbf{L}_{phon}(i,t) > 40 \\ (\frac{1}{40}\mathbf{L}_{phon}(i,t))^{2.642} & \text{otherwise.} \end{cases} \tag{3.3}$$

The relationship between phon and sone can be seen in Figure 3.3. For low values up to 40 phon the sensation rises slowly until it reaches 1 sone at 40 phon. Beyond 40 phon the sensation increases at a faster rate.

The fluctuation strength of a tone with the loudness ΔL, which is 100 percent amplitude modulated with the frequency f_{mod} can be expressed by [29],

$$f_{flux}(\Delta L, f_{mod}) \propto \frac{\Delta L}{(f_{mod}/4\text{Hz}) + (4\text{Hz}/f_{mod})} \tag{3.4}$$

The modulation amplitudes $F(i,n)$ of the i-th critical-band and the modulation frequency f_{res} are weighted according to the fluctuation strength sensation as follows,

$$\mathbf{F}(i,n) = f_{flux}(|\Delta L_i(n+1)|, f_{res}(n)) \tag{3.5}$$

The fluctuation pattern is computed in a similar way by calculating the rhythmic periodicity along different auditory channels. One way of estimating the rhythmic fluctuation is based on spectrogram computation transformed by auditory modeling and then a spectrum estimation in each band [29]. As the output, a matrix is generated showing the rhythmic periodicities for each different Bark band.

The weighing function of amplitude modulation coefficients based on the psychoacoustic model of the fluctuation strength is given in Figure 3.4.

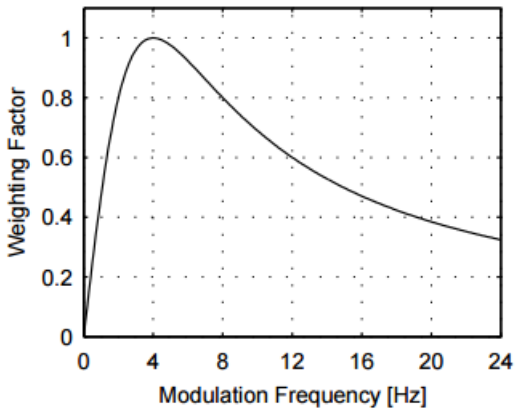

Figure 3.4: Image courtesy: Pampalk [29]. The relationship between fluctuation strength and the modulation frequency.

3.1.2 Algorithm for computing fluctuation [1]

- Compute power spectrogram of the input audio
  - The Terhardt outer ear modeling is computed.
  - A multi-band redistribution of the energy is performed along the 'Bark' bands decomposition.
  - Masking effects are estimated on the multi-band distribution.
  - Finally the amplitudes are represented in dB scale.
- Compute FFT on each band
  - Frequencies range from 0 to m Hz, where m is set by default to 10 Hz can be controlled.
  - The frequency resolution of the FFT is set by default to .01 Hz and can also be controlled.
  - The amplitude modulation coefficients are weighted based on the psychoacoustic model of the fluctuation strength [30].

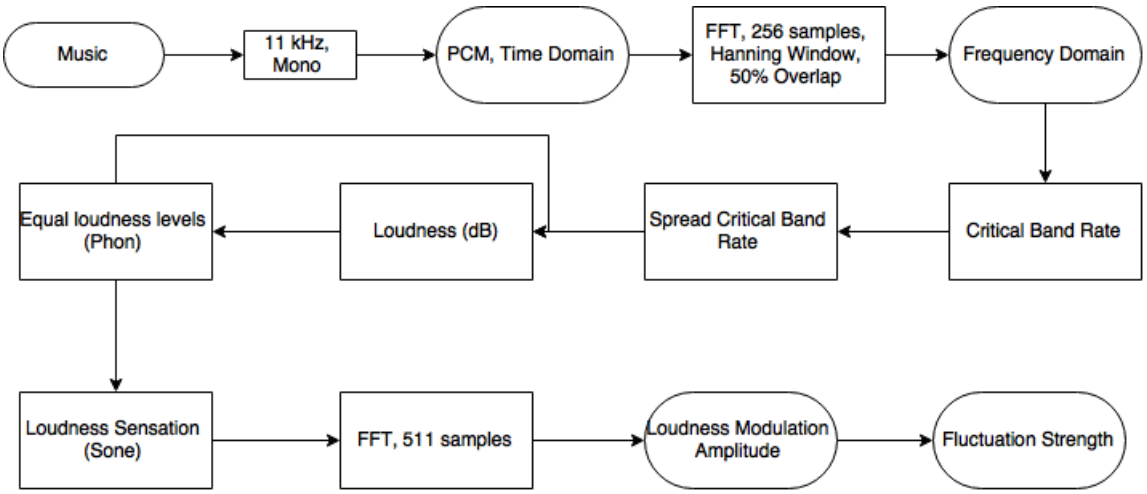

Figure 3.5: Flowchart describing the algorithm of computing band wise fluctuation strength

3.1.3 Tempo Estimation from Fluctuation Pattern

Tempo estimation after computing fluctuation pattern is straightforward. As the output of the fluctuation pattern algorithm, for an audio, we get a matrix $\mathbf{M}(i,j)$ of size 1025×25 where $1 < i < 1025$ and $1 < j < 25$. Here i corresponds to the modulation frequency, multiplied by 100, and j corresponds to Bark bands.

The matrix is summed across all bands to get a one dimensional overall fluctuation strength with respect to modulation frequency:

$$m(i) = \sum_{j=1}^{25} \mathbf{M}(i,j), \quad i = 1, ..., 1025 \tag{3.6}$$

This signal $m(i)$ is multiplied by the tempo prior probability function (described in Appendix A), and the maximum is picked and converted to BPM units. For example, a peak at 2 Hz will correspond to a tempo of 120 BPM.

$$\tilde{m}(i) = m(i) \times P_0(i) \tag{3.7}$$

$$\text{Tempo} = \text{argmax}\ \tilde{m}(i) \tag{3.8}$$

Figure 3.7 helps explain it graphically.

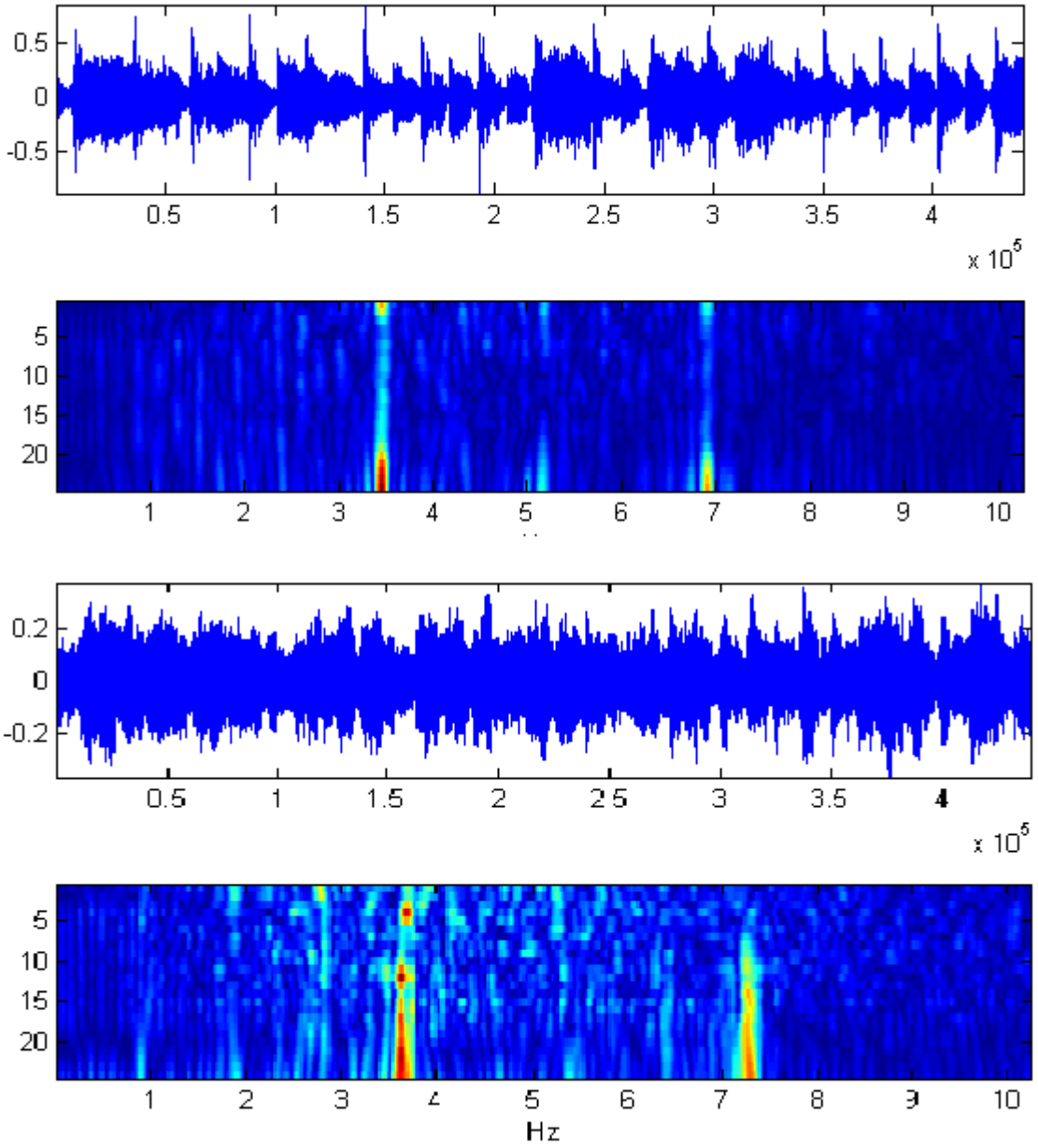

Figure 3.6: Fluctuation patterns for two songs with similar rhythm structure and similar tempo.

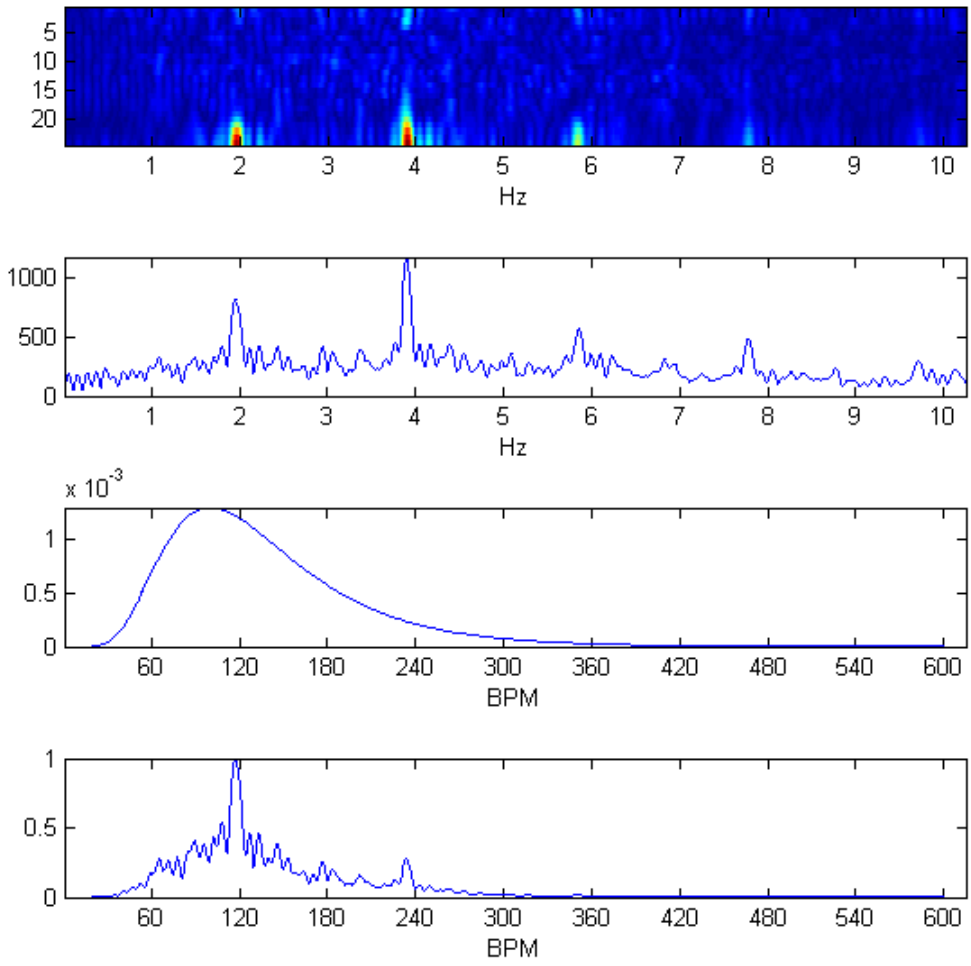

Figure 3.7: Process of tempo estimation from fluctuation pattern

3.2 Computation Scheme of Spectral Centroid Feature

The spectral centroid is a measure used to characterise a spectrum. It indicates where the "center of mass" of the spectrum is. Perceptually, it has a robust connection with the impression of "brightness" of a sound. [31]. It is calculated as the weighted mean of the frequencies present in the signal, determined using a Fourier transform, with their magnitudes as the weights [32]:

$$S(n) = \frac{\sum_{n=0}^{N-1} f(n)x(n)}{\sum_{n=0}^{N-1} x(n)} \tag{3.9}$$

where $x(n)$ represents the weighted frequency value, or magnitude, of bin number n, and $f(n)$ represents the center frequency of that bin.

When used as an onset detection feature spectral centroid effectively picks up the points in the audio having high-frequency content. This is based on the assumption that note transients and percussive hits tend to have more high-frequency along with high energy. Moreover, the sensation of rhythm is also carried by a fluctuating sound in terms of pitch, and not just amplitude.

To differentiate between transient sounds that contain high-frequency content from steady sounds at high frequencies, the derivative of the spectral centroid signal is taken, since onsets are primarily temporal features, and one would like to detect transients for onset detection purposes.

Figure 3.8 shows the process of onset detection by using the spectral centroid. Onsets are detected where the derivative of the spectral centroid signal peaks.

Let $\dot{S}(n)$ denote the derivative of spectral centroid signal.

$$\dot{S}(n) = \frac{\mathrm{d}}{\mathrm{d}n} S(n) \tag{3.10}$$

3.2.1 Tempo Estimation from Spectral Centroid

In order to compute the tempo, the auto-correlation approach is taken. From $\dot{S}(n)$, the auto-correlation is computed. Given a signal $f(t)$, the continuous auto-correlation $R_{ff}(\tau)$ is most often defined as the continuous cross-correlation integral of $f(t)$ with itself, at lag τ.

$$R_{ff}(\tau) = (f * g_{-1}(\overline{f}))(\tau) = \int_{-\infty}^{\infty} f(u+\tau)\overline{f}(u)\,\mathrm{d}u = \int_{-\infty}^{\infty} f(u)\overline{f}(u-\tau)\,\mathrm{d}u \tag{3.11}$$

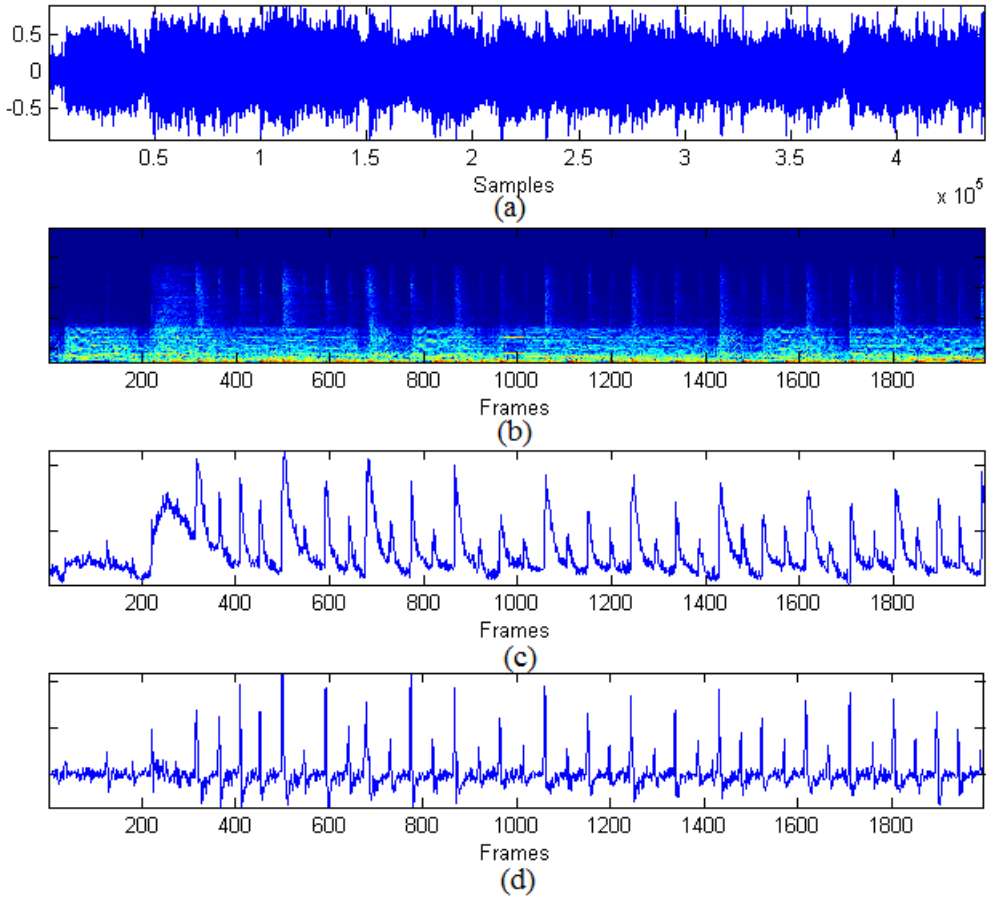

Figure 3.8: Process of onset detection using spectral centroid. (a) is the audio signal (b) is its spectrogram (c) is the spectral centroid signal, frame-wise (d) is the derivative of the spectral centroid function

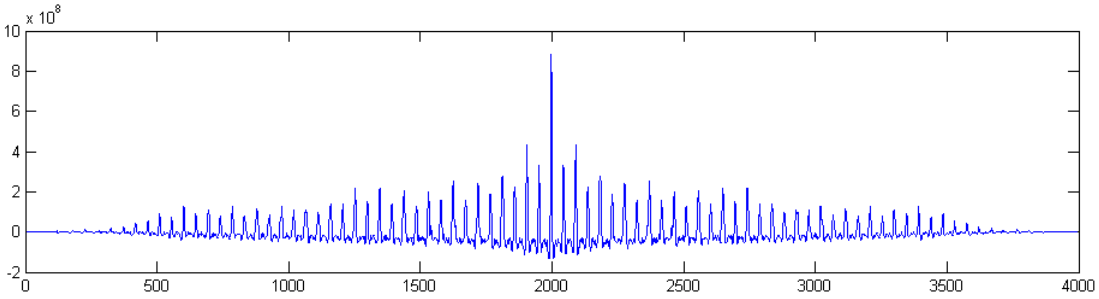

Figure 3.9: Auto-correlation of the derivative of spectral centroid signal

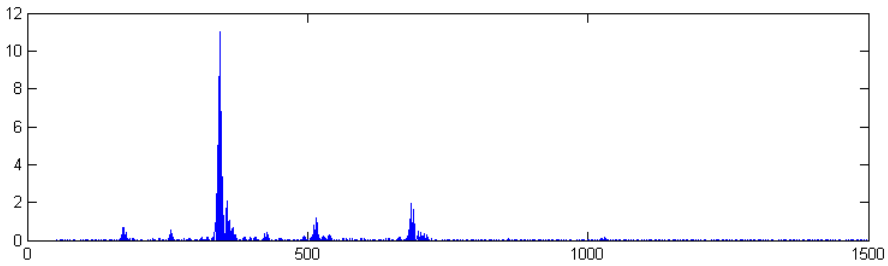

Figure 3.10: Fourier transform of auto-correlation of derivative of spectral centroid signal

where $\overline{f}$ represents the complex conjugate, g_{-1} is a function which manipulates the function f and is defined as $g_{-1}(f)(u) = f(-u)$ and $*$ represents convolution. f and $\overline{f}$ are equal for a real signal.

Therefore, the auto-correlation function of $\dot{S}(n)$, denoted by $R(n)$, is given as:

$$R(n) = \sum_{i=-N}^{N} \dot{S}(i)\dot{S}(i-n) \tag{3.12}$$

and shown in Figure 3.9

A Fourier transform of $R(n)$ is then taken to find the frequency distribution of oscillation of the auto-correlation signal. The Fourier transform of the auto-correlation function in Figure 3.9 is given in Figure 3.10

The Fourier spectrum so obtained is then scaled by the tempo prior probability distribution function as given in Appendix A, and the peak of the spectrum is extracted. The frequency at which the peak is present is converted to tempo value in BPM by

$$\text{tempo} = f_p \times \frac{f_{max}}{N_{\text{FFT}}} \times 60 \tag{3.13}$$

where f_p is the FFT point at which the peak occurs, f_{max} is the Nyquist frequency of the audio (i.e. for an audio with sampling rate of 44.1 kHz, $f_{max} = 22.05$ kHz), and N_{FFT} is the number of points in the FFT.

3.3 Summary

In this chapter, tempo estimation procedure using two different novel techniques were proposed: fluctuation and spectral centroid. While fluctuation uses amplitude modulation frequencies in different Bark bands to estimate the overall modulation frequency, later converted to BPM, to compute tempo, the spectral centroid method first detects onsets in the music signal, takes its derivative, computes auto-correlation and finds the dominant frequency of oscillation of the auto-correlation signal. This is then converted to BPM values.

Chapter 4

Tempo Estimation using Sub-Band Synchrony

In the previous chapter, we saw that many rhythmic onsets have distinctive spectral shape - onsets show as vertical lines in the spectrogram. This means that the envelope in different bands in the spectrum see an increase in a coherent fashion. This property is what was sought to be extracted by sub-band synchrony.

An overview of the entire process of tempo induction using sub-band synchrony is given in the flowchart on the next page.

4.1 Splitting into sub-bands

The input audio is first split into a number of sub-bands. Splitting into 40 sub-bands gave the best results and is thus used as the number of bands henceforth. A filter bank with 40 filters is applied to the input audio and the sub-band audio outputs are extracted. The filter bank uses gammatone filters are used as they are widely used as approximations of auditory filters in the human auditory system. A gammatone filter is a linear filter described by an impulse response that is the product of a gamma distribution and sinusoidal tone. The impulse response is given by [33]

$$g(t) = at^{n-1}e^{-2\pi bt}\cos(2\pi ft + \phi) \tag{4.1}$$

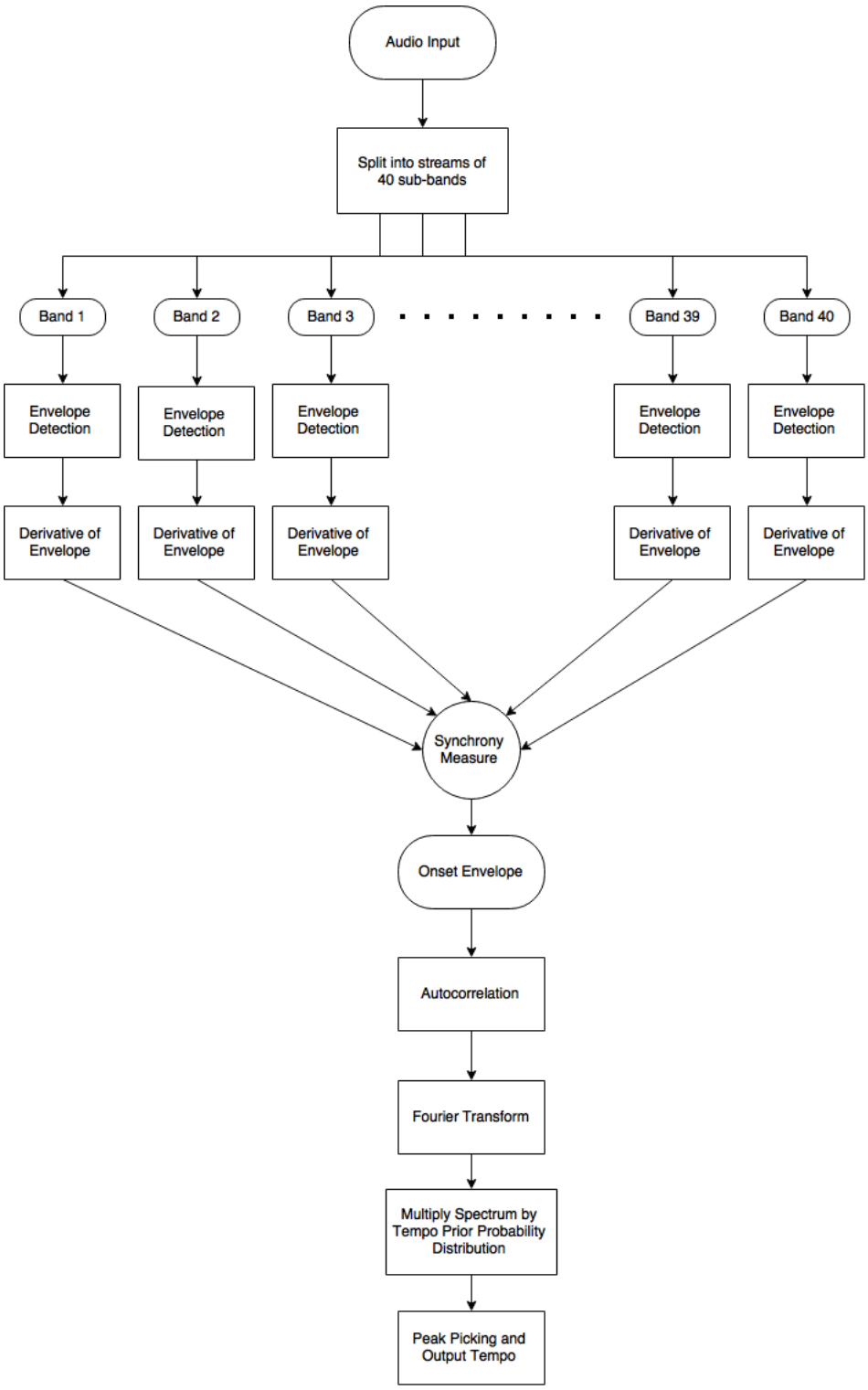

Figure 4.1: Flowchart for the proposed Sub-Band Synchrony Algorithm for Tempo Induction

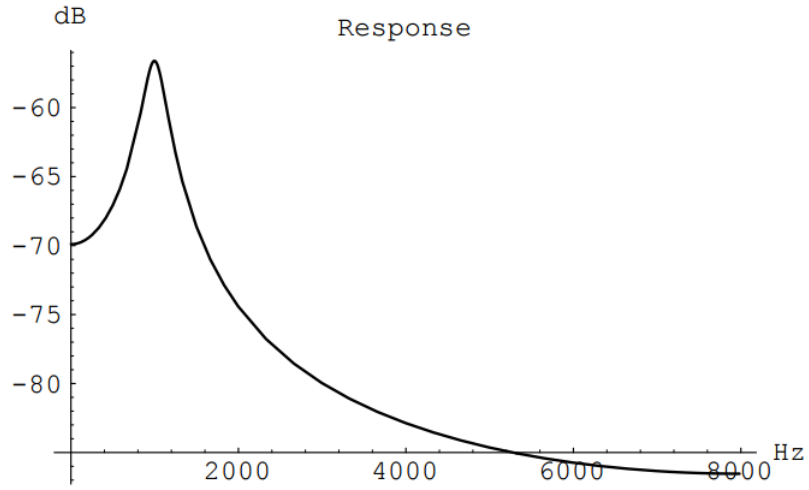

Figure 4.2: Frequency response of a gammatone filter centered at 1000 Hz

and its frequency response looks like Figure 4.2.

4.2 Band-wise Processing

An envelope detection function is applied to each of the sub-band outputs. The scheme remains the same as shown in chapter 1. When the combined envelopes of all the sub-bands are visualised as a heat map, we get an image as shown in Figure 4.3 (b). In order to detect onsets determined by either an increase in energy in a majority of the sub-bands simultaneously or a change in pitch of the sound, it is necessary to detect transients in the envelopes of the sub-bands. This is done by taking the derivative of each the 40 envelopes. This is shown in Figure 4.3 (c). This entire process can be summarized by the following equations:

$$s_k(n) = g_k(n) * s(n), \qquad k = 1, 2, ..., 40 \tag{4.2}$$

where $s(n)$ is the input audio signal, $g_k(n)$ is the k-th gammatone filter, and $*$ denotes convolution:

$$x(n) * y(n) = \int_{-\infty}^{\infty} x(\tau)y(t-\tau) \tag{4.3}$$

This is the output of the filter bank, with 40 filters. Next the signal is squared. Squaring the signal demodulates the input by using the input as its own carrier wave. This means that half the energy of the signal is pushed up to higher frequencies and half is shifted down toward DC.

$$\tilde{s}_k(n) = 2 \times [s_k(n)]^2 \tag{4.4}$$

To maintain the correct scale, two additional operations must be performed. First, the signal must be amplified by a factor of two. Since we are keeping only the lower half of the signal energy, this gain matches the final energy to its original energy. Second, we must take the square root of the signal to reverse the scaling distortion that resulted from squaring the signal, which is done later in Equation 4.6. The signal is then downsampled to reduce the sampling frequency. The downsampling can be done if the signal does not have any high frequencies which could cause aliasing. Otherwise an FIR decimation should be used which applies a low pass filter before downsampling the signal. Here, the downsampling is done by a factor of 4.

$$\tilde{s}_{k,\mathrm{dec}}(n) = \mathrm{dec}(\tilde{s}_k(n), 4) \tag{4.5}$$

where $\mathrm{dec}(x, n)$ denotes the decimation function decimating x by a factor of n. The next step computes the envelope by passing through a low pass filter and taking the square root of the resulting signal.

$$E_k(n) = \sqrt{h(n) * \tilde{s}_{k,\mathrm{dec}}(n)} \tag{4.6}$$

where $E_k(n)$ denotes the final sub-band envelope for band k, and $h(n)$ is the impulse response of a two degree low pass filter used here as a smoothing function. In order to detect changes in the spectrum that correspond to rhythmic onsets, it is necessary to compute a measure of temporal change of the envelopes. We take the derivative of all the envelopes to achieve this.

$$D_k(n) = \frac{\mathrm{d}}{\mathrm{d}n} E_k(n) \tag{4.7}$$

where $D_k(n)$ is the derivative of the envelope $E_k(n)$.

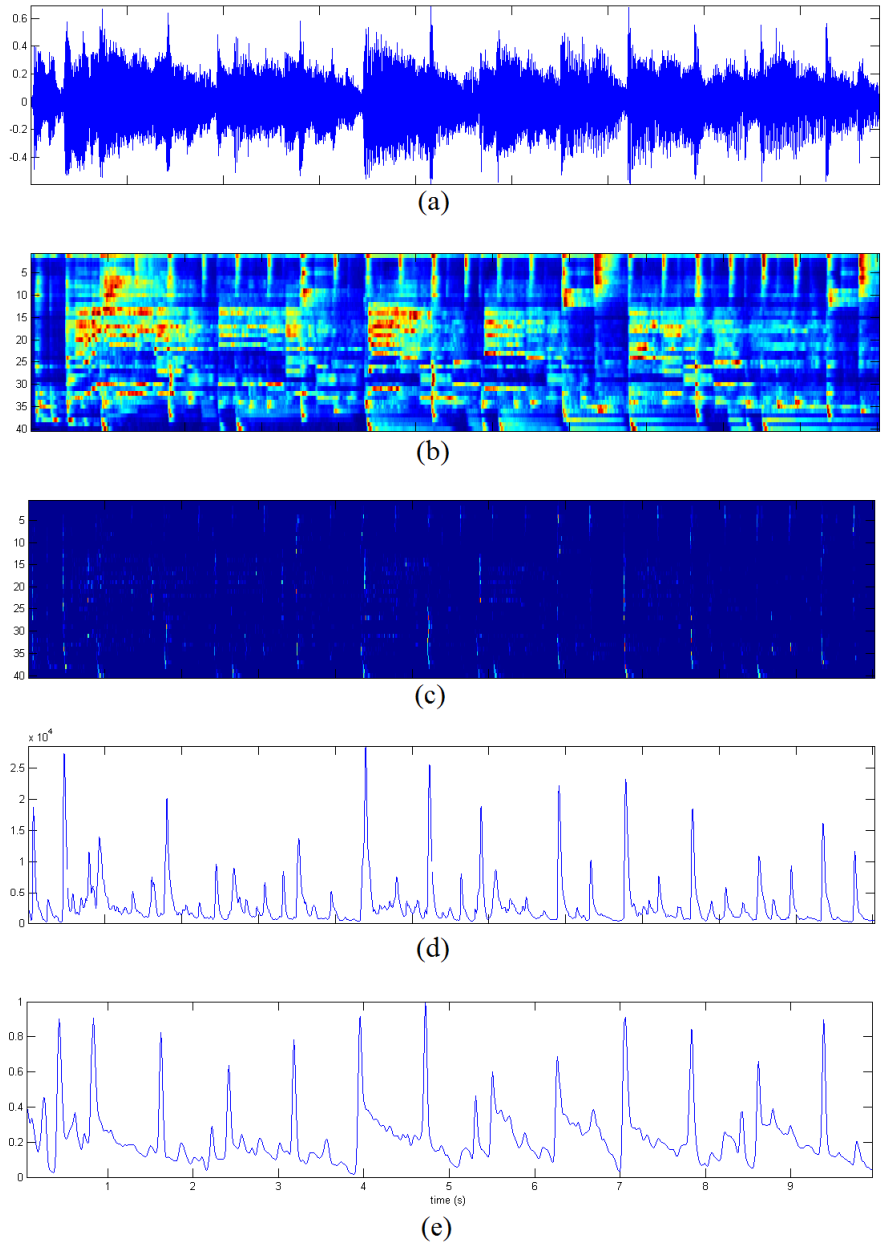

Figure 4.3: Onset Detection process using sub-band synchrony for an excerpt of the song *Careless Whisper*. (a) Visualization of the audio, (b) 40 sub-band envelopes as heat-map, (c) Derivative of each band, (d) Onset envelope obtained after taking mean of the derivative envelopes for each frame (e) Onset envelope from amplitude envelope

4.3 Computing Synchrony and Onset Curve

After we obtain the derivative of the sub-band envelopes, we are interested in locating the times at which there is a coherent motion in many of the bands, as this is an observed feature of onsets. Two different methodologies were explored to compute this "synchrony" between sub-bands. First of these is the variance of derivatives at each frame across bands. The second is the mean over all the bands, which is what has been used finally because of its simplicity and effective output.

4.3.1 Variance based synchrony measure

As can be seen in Figure 4.3 (c), derivatives of the sub-band envelopes at non-onset times are mostly close to zero across all bands, whereas those at onset positions have a greater variability. Calculation of variance across bands reflect this by giving maxima at these locations. This has been utilised for onset detection. For each time frame, the variance across sub-bands is computed and plotted. An example is shown in Figure 4.4.

$$\sigma(n) = \frac{1}{40}\sum_{k=1}^{40}[D_k(n) - \mu(n)]^2 \tag{4.8}$$

where $D_k(n)$ is as defined in Equation 3.4 and $\mu(n)$ is the mean across bands

$$\mu(n) = \frac{1}{40}\sum_{k=1}^{40} D_k(n) \tag{4.9}$$

Figure 4.4 clearly shows onsets detected by computing the variance as described above. As compared to the amplitude envelope onset curve in Figure 4.3, this curve has more distinct peaks and less background noise.

4.3.2 Mean based synchrony measure

From Figure 4.3, it can also be observed that the points of onset have overall greater values of derivatives of sub-band envelopes. The mean, in such case, is expected to be higher at onset points. For each time frame, the mean of the derivative of sub-band envelopes across bands is computed and plotted. This gives our onset curve.

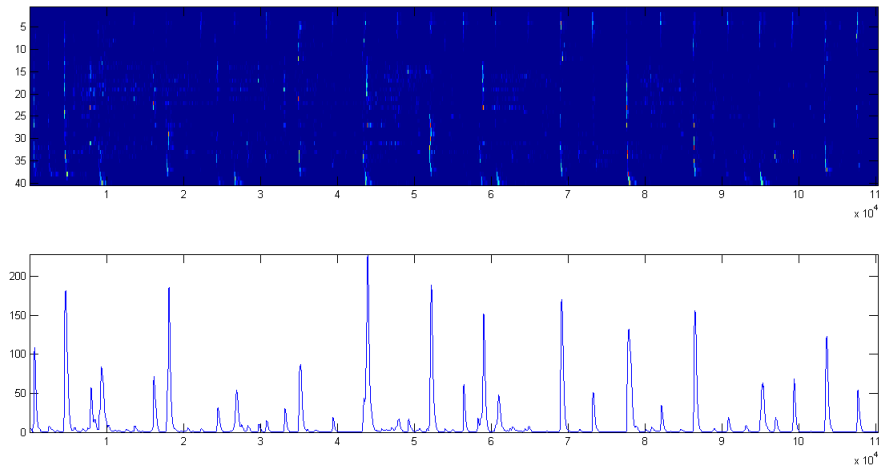

Figure 4.4: Onset detection using sub-band synchrony (variance) for song *Careless Whisper*

$$\mu(n) = \frac{1}{40}\sum_{k=1}^{40} D_k(n) \tag{4.10}$$

The onset curve calculated in this way is given in Figure 4.3 (d).

4.4 Tempo Calculation

Tempo calculation is done by taking the auto-correlation approach like in Chapter 3 and defined in Equation 3.11. Let $O(n)$ be the onset curve obtained by using either the mean based synchrony method or using the variance based synchrony method. Then, its auto-correlation, $\tilde{O}(n)$ is given by

$$\tilde{O}(n) = \sum_{i=-N}^{N} \dot{S}(i)\dot{O}(i-n) \tag{4.11}$$

A Fourier Transform of $\tilde{O}(n)$ is calculated to find the frequency distribution of oscillation of the auto-correlation signal. The Fourier spectrum so obtained is then scaled by the tempo prior probability distribution function as given in Appendix A, and the peak of the spectrum is extracted. The frequency at which the peak is present is converted to tempo value in BPM by

$$\text{tempo} = f_p \times \frac{f_{max}}{N_{\text{FFT}}} \times 60 \tag{4.12}$$

where f_p is the FFT point at which the peak occurs, f_{max} is the Nyquist frequency of the audio (i.e. for an audio with sampling rate of 44.1 kHz, $f_{max} = 22.05$ kHz), and N_{FFT} is the number of points in the FFT.

4.5 Summary

In this chapter a tempo estimation scheme using sub-band synchrony has been described. The scheme involves computing the onset curve of a music signal by band-wise temporal analysis of amplitude envelope functions and then computing the synchrony measure fame-wise over all the bands. The frames at which there is high synchrony or coherence between the derivatives of the amplitude envelope signals in different bands are taken to be onset positions. After obtaining the onset curve, its correlation with itself is taken as a new signal and subsequently its Fourier spectrum is computed. This gives the amplitudes of the frequency components in the auto-correlation signal. Taking the frequency at which the spectrum peaks gives us the dominant oscillation frequency of the auto-correlation signal. This is then used to compute the tempo of the onset signal, and therefore, the music signal, after weighting with the prior tempo probability function.

Chapter 5

Performance Evaluation

This chapter presents accuracy measurements of various tempo estimation algorithms by comparing estimated tempos to those established by human subjects. The first section elaborates the database used for the experiments, explaining the types of music samples used and their musical and spectral features. The next section describes the tapping experiment, which is a method of establishing the ground truth perceptual tempo values for the songs in the database. Third section describes in detail the process of tempo estimation using the algorithms given in this thesis and the obtained results. This is followed by a results summary and a discussion on the observations.

5.1 Database Used for Performance Evaluation

The database was generated from music collections of 4 different genres, namely, Western Classical (denoted as "WC" henceforth), Indian Classical (denoted as "IC" henceforth), Rock music (denoted as "Rock" henceforth), and Popular music (denoted as "Pop" henceforth"). 10 second clips were extracted from the audio files and saved in a sorted fashion into different sets according to genre. Western Classical and Indian Classical datasets contain 30 clips of 10-second length each. Pop and Rock categories, on the other hand, contain 60 clips of 10-second length each.

Moreover, a test set was also generated using the FL Studio Digital Audio Workstation (denoted as "Test" henceforth). The test set consists of 30 clips of 10-second length, and each

clip has a monophonic melody but using different instruments. Also, the audio in the test set were passed through a dynamic range compression stage in order to make the envelope variation as little as possible. This was done in order to test the efficacy of the algorithms presented in this thesis on the cases where there are no sharp energy onsets associated with note changes, but only spectral changes are present.

Other than these, a free dataset was downloaded from the url: `http://labrosa.ee.columbia.edu/projects/beattrack/`, which is a dataset provided by Dan Ellis of LabROSA, and used for the development of 2006 MIREX Beat Tracking Evaluation (denoted as "Ellis" henceforth). It consists of 20 clips of 30 seconds each and each of the files are annotated with human-tapped onsets and ground truth tempo values. The summary of the dataset is provided as under.

	Datasets						
Attributes	WC	IC	Pop	Rock	Test	Ellis	Overall
Number of Clips	30	30	60	60	30	20	230
Duration of Each (s)	10	10	10	10	10	30	NA
Total Duration (s)	300	300	600	600	300	600	2700
Max Tempo (bpm)	163.5	334.0	191.0	278.0	220.5	130.0	334.0
Min Tempo (bpm)	81.5	90.0	72.0	70.0	54.0	90.0	54.0

Table 5.1: Database summary

The salient features of each dataset of the database have been provided in the following sections:

5.1.1 Western Classical Music Dataset

This dataset consists of recordings of classical Western compositions including music by Mozart, Beethoven, Pachelbel and Chopin. Songs were chosen in such a way that it will be possible to define a ground truth tempo value, or in other words, those songs were chosen that have a constant and uniform rhythm to make the readings from the tapping experiment

in Section 5.2 unambiguous. It is in general a difficult task to define characteristics of any genre, attempts at describing the style may help in shaping algorithms that are suited for that genre and in explaining why certain features are typical for that genre. Western classical music consists mostly of orchestral instruments, like violin, viola, cello, piano, and flutes, and percussion instruments are either absent or less pronounced. The attack time for instruments like violin are also usually long, and thus onset detection becomes an extremely difficult problem for this genre. Since percussion is less pronounced, the fluctuation patterns for western classical songs are usually smeared and lack distinct peaks. Different features for a typical song from this genre, and onset detection using temporal and spectral measures is given in Figure 5.1.

5.1.2 Indian Classical Music Dataset

This dataset consists of recordings of classical Indian compositions as well as contemporary compositions in the classical style. Recordings of Ustad Zakir Hussain, Pandit Hariprasad Chaurasia, Ustad Alla Rakha and Pandit Ravi Shankar have been included in this dataset. Most songs have a distinctive *tabla* beat as an accompaniment. Indian classical music usually contains the *tabla* as a prominent accompanying percussion instrument. The sense of rhythm in this genre comes from the sharp hits on the smaller tabla drum. Indian classical music is also characterised by fewer instruments playing at once, than western classical, i.e. lesser polyphony. Different features for a typical song from this genre is given in Figure 5.2. This genre, in general, has a more strongly peaked fluctuation pattern and thus a better estimation of tempo with this method. Sub-band synchrony also performs well as it picks up many onsets that may be missed by the typical onset detection algorithms. Features for a typical song from this genre are shown in Figure 5.2.

5.1.3 Pop Music Dataset

The pop music dataset was made from recordings of various popular songs through the decades, including artists like Elvis Presley, Michael Jackson, Bee Gees, Adele, Ed Sheeran and many more. Musicologists often identify the following as on of the characteristics typical of the pop music genre: much pop music is intended to encourage dancing, or it uses dance-

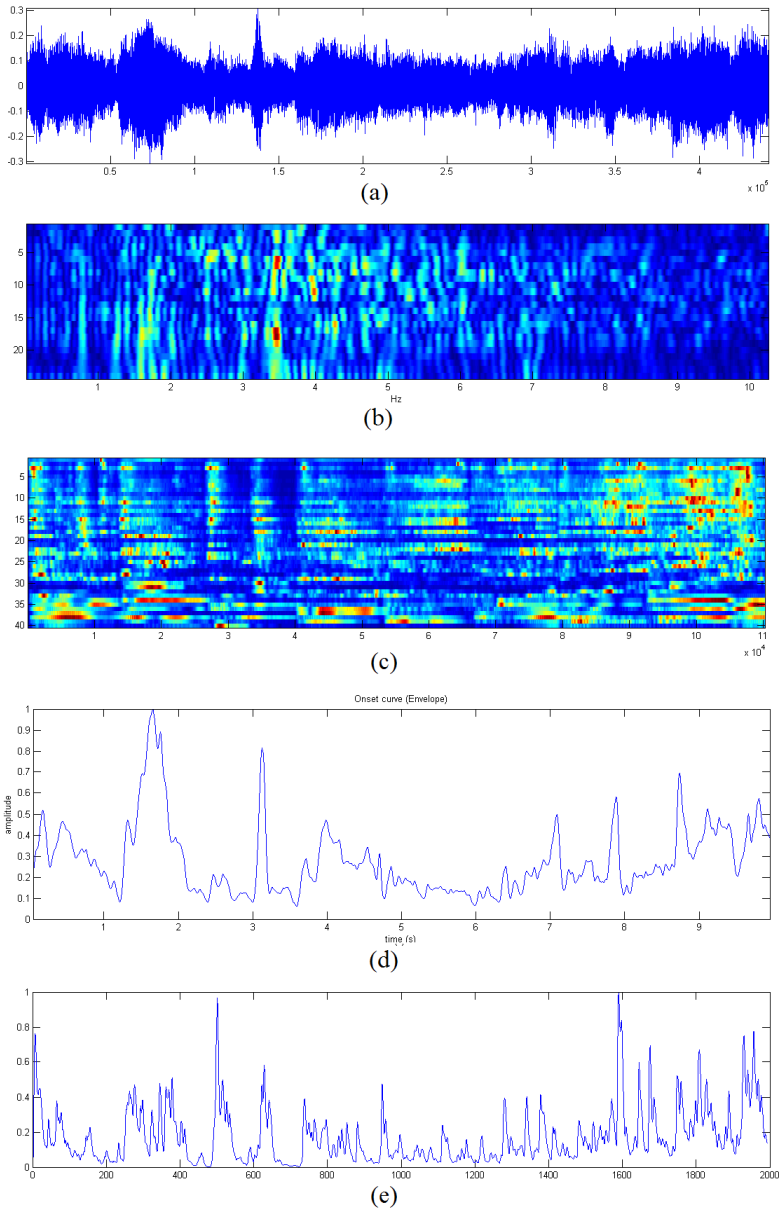

Figure 5.1: Features for a Western classical song *Pachelbel's Canon in D*. (a) Audio, (b) Fluctuation pattern, (c) Band-wise envelopes, (d) Onset curve based on amplitude, (e) Onset curve based on sub-band synchrony (mean)

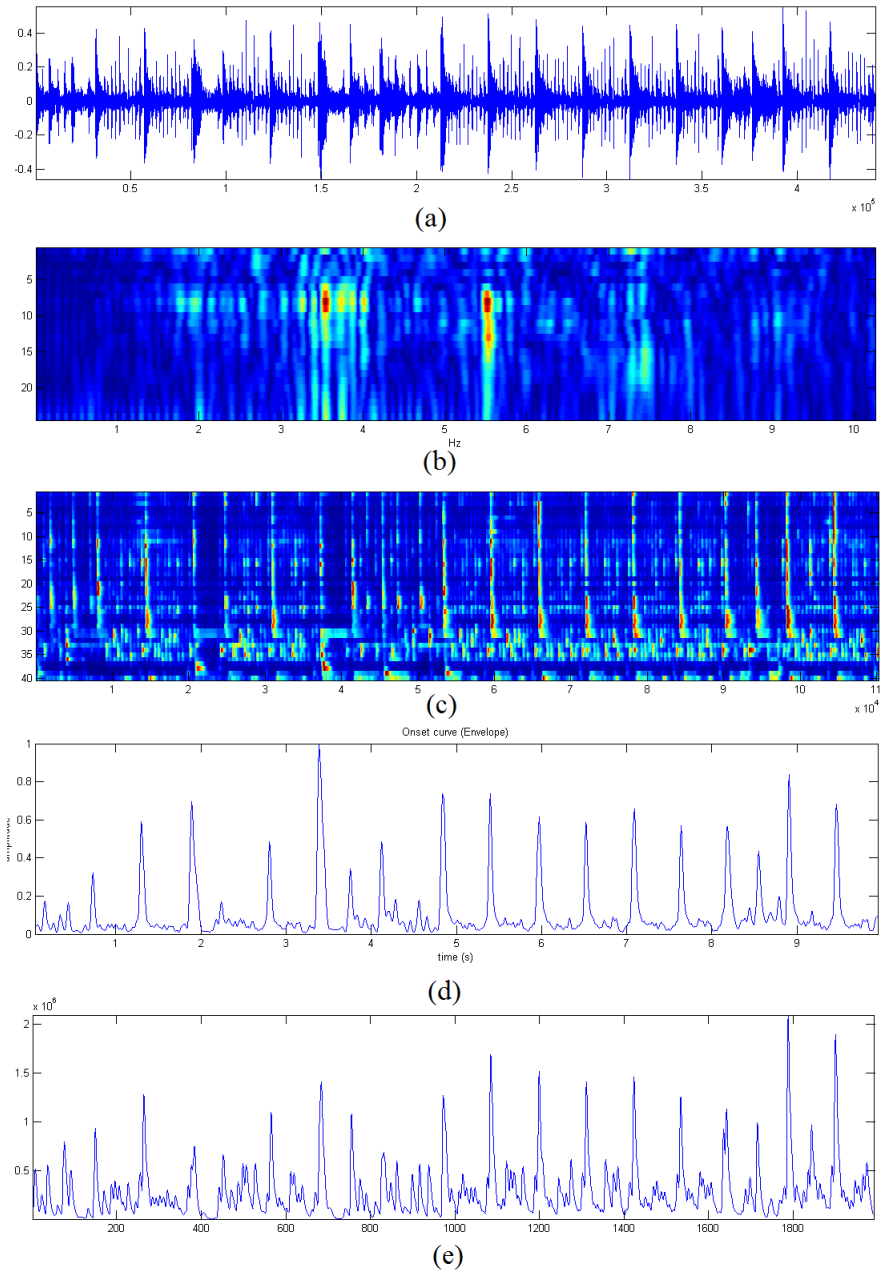

Figure 5.2: Features for an Indian classical song. (a) Audio, (b) Fluctuation pattern, (c) Band-wise envelopes, (d) Onset curve based on amplitude, (e) Onset curve based on sub-band synchrony (mean)

oriented beats or rhythms. This means that pop music has a clear and steady rhythm and tempo. The main medium of pop music is the song, often between two and a half and three and a half minutes in length, generally marked by a consistent and noticeable rhythmic element, a mainstream style and a simple traditional structure. Common variants include the verse-chorus form and the thirty-two-bar form, with a focus on melodies and catchy hooks, and a chorus that contrasts melodically, rhythmically and harmonically with the verse. The beat and the melodies tend to be simple, with limited harmonic accompaniment. These features of pop music make it easier for machines and humans to estimate the tempo. This is reflected by the fact that the lowest RMSE values for all algorithms of tempo estimation are obtained for this genre. In this genre, sub-band synchrony outperforms other algorithms. A sample of pop music and associated features is elaborated in Figure 5.3.

5.1.4 Rock Music Dataset

This dataset contains songs from the classic rock period, with artists like ACDC, Led Zeppelin, blues artists like B.B.King and Eric Clapton, and modern rock and metal artists like Lamb of God and Pantera. The sound of rock is traditionally centered on the electric guitar, which emerged in its modern form in the 1950s with the popularization of rock and roll. The sound of an electric guitar in rock music is typically supported by an electric bass guitar pioneered in jazz music in the same era, and percussion produced from a drum kit that combines drums and cymbals. Rock music is traditionally built on a foundation of simple unsyncopated rhythms in a 4/4 meter, with a repetitive snare drum back beat on beats two and four. Most songs in this genre have a distinct drum rhythm going on, and thus have clear onset points. Sub-band synchrony works well as the onset curve is accurate and does not miss masked onsets. Fluctuation patterns also have distinct peaks because of the steady rhythm.

5.1.5 Mixed Dataset from LabROSA

Dan Ellis's dataset downloaded from: `http://labrosa.ee.columbia.edu/projects/beattrack/`, consists of 20 songs from different genres of varying rhythmic styles and tempos. The files are annotated with human-tapped onsets and ground truth tempo values. Since this is a mixed

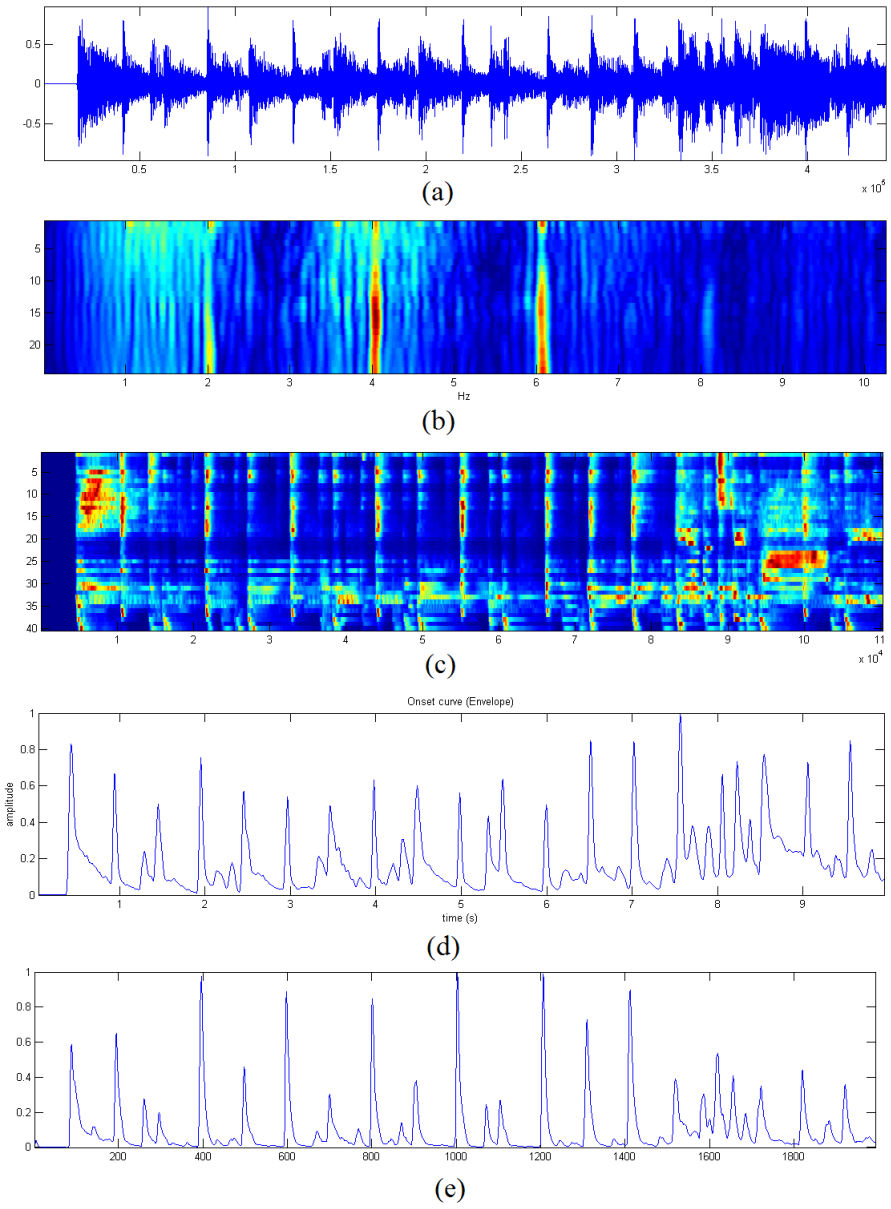

Figure 5.3: Features for a typical pop song. (a) Audio, (b) Fluctuation pattern, (c) Band-wise envelopes, (d) Onset curve based on amplitude, (e) Onset curve based on sub-band synchrony (mean)

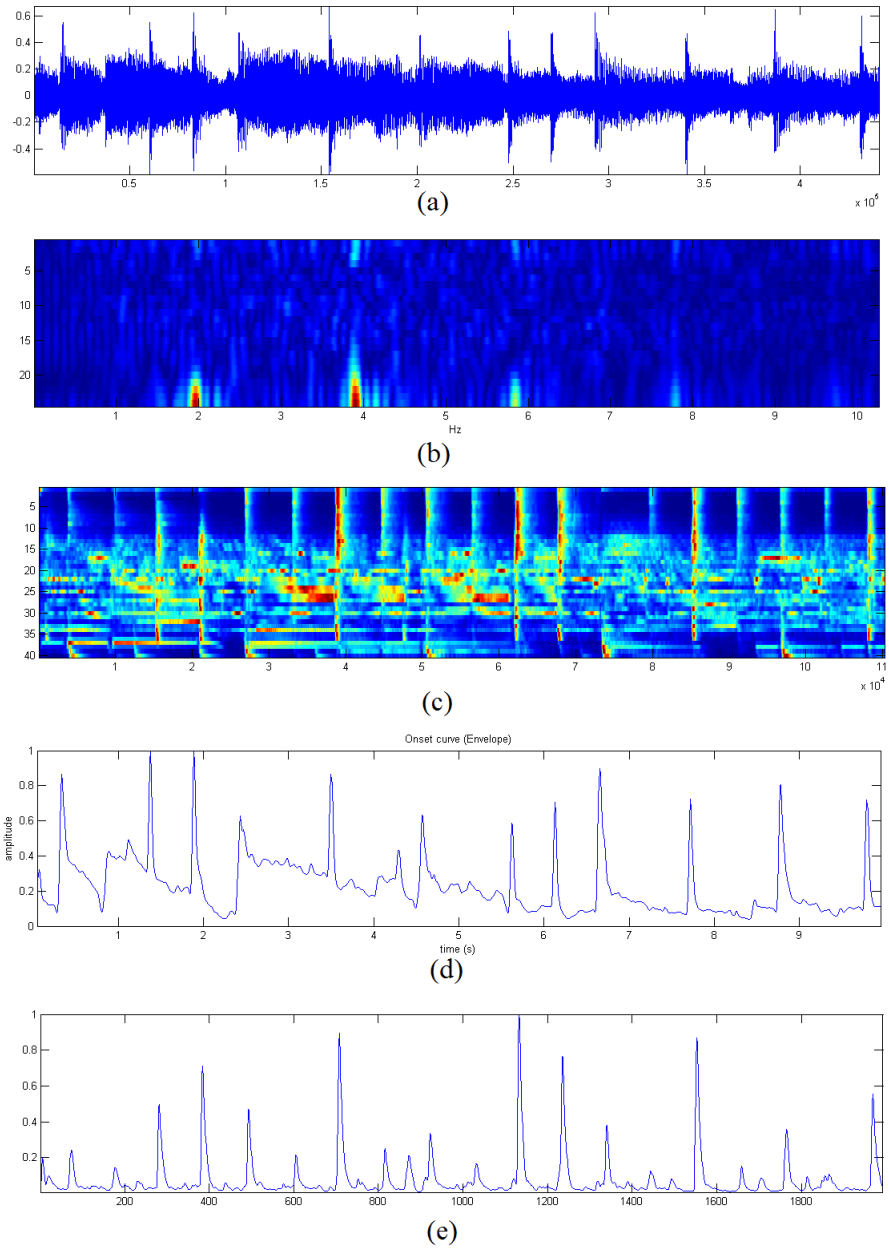

Figure 5.4: Features for a typical rock song. (a) Audio, (b) Fluctuation pattern, (c) Band-wise envelopes, (d) Onset curve based on amplitude, (e) Onset curve based on sub-band synchrony (mean)

dataset, the "typical" song feature figure is skipped for this dataset. The tempo estimation error analysis for this set is given in Table 5.6

5.1.6 Generated Test Dataset

The test dataset was generated in a way that suppresses onsets based on amplitude changes. The pieces are monophonic (only one instrument playing at a time) with no percussion instruments being played. Thus onset detection is expected to work well if it is based on spectral features, and hence we see a large error in the amplitude envelope onset detection method. Sub-band synchrony gives the best result in this case also. Features shown in Figure 5.5

5.2 Establishment of Ground Truth Tempo Values using Tapping Experiment

In order to determine the ground truth tempo values for the songs in the dataset (except LabROSA's annotated dataset), the standard approach of tapping was used. Tapping feet or fingers to the beat of a song is a tendency natural to most of us. Trained musicians can follow a beat consistently and hence are a good measure of the perceptual tempo of a song.

5.2.1 Conditions of the Tapping Experiment

The experiment specifics and conditions are elaborated in this section.

- A musically trained person was provided with the audio clips and asked to tap a key on the computer keyboard at constant rate, with each tap coinciding with what the subject perceived as the tactus of the music being played.
- The average time period between taps was noted and the tempo computed according to $T = 60/\overline{\Delta t}$ where T is the computed tempo, and $\overline{\Delta t}$ is the averaged time period of taps. This is considered to be one reading.
- The human subject was given the chance to listen to the clips as many times as they wished, and the tempo reading was only taken after achieving a sufficiently constant

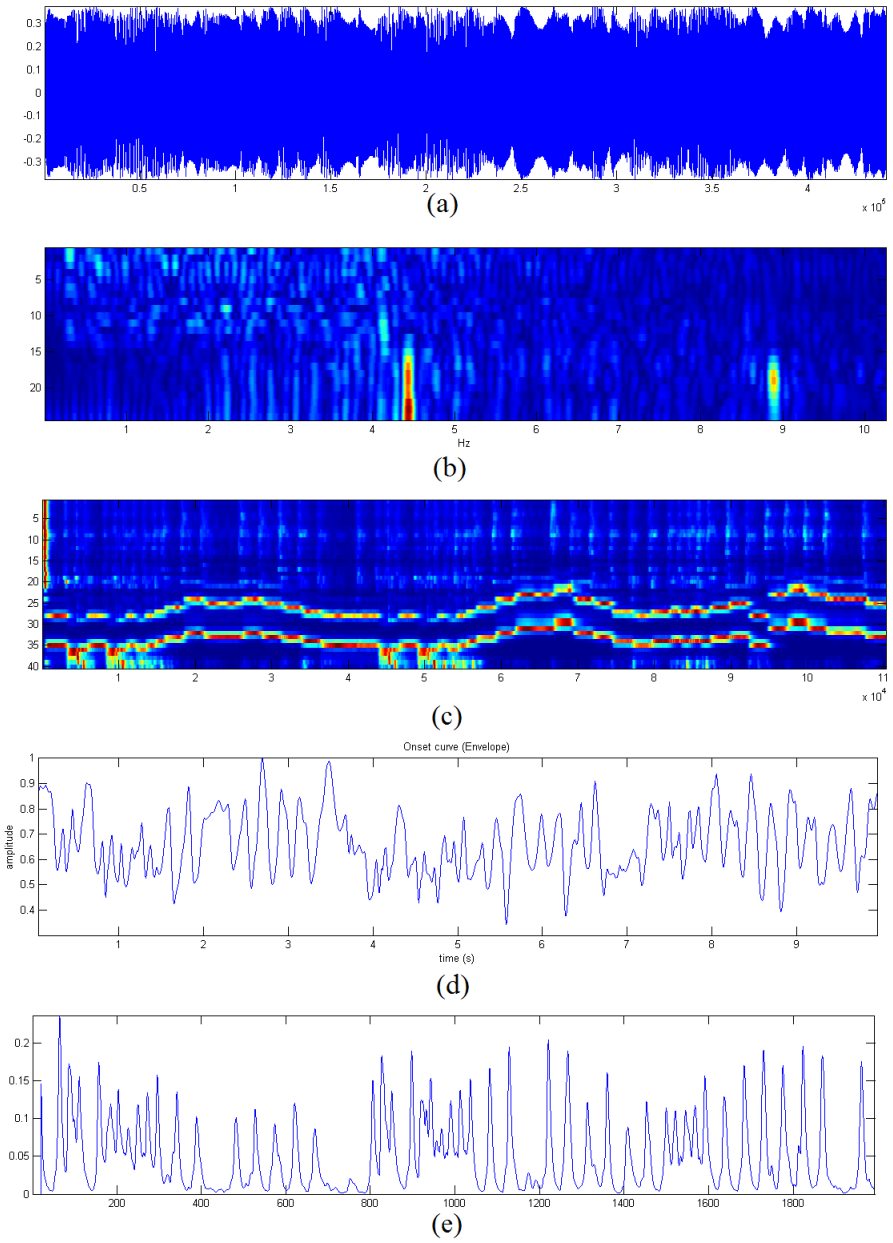

Figure 5.5: Features for a typical rock song. (a) Audio, (b) Fluctuation pattern, (c) Band-wise envelopes, (d) Onset curve based on amplitude, (e) Onset curve based on sub-band synchrony (mean)

tapping rate.

- Each song was played thrice to the listener but in a randomised order, with other songs being played and tempo measured in between. Thus for each song, there was a total of three readings.
- In case the subject differed by a factor of approximately 2 between the readings, then that tempo is chosen on which two of the three readings agree around, and the remaining reading is halved or doubled depending on whether this was double or half of the agreed value initially.
- If all three differed, then the median value was taken to be the approximate range of the tempo and the other two values halved or doubled accordingly.
- Finally, the average of all readings is taken to be the final ground truth value.
- The least count of the recorded tempos was 0.5 BPM.

5.3 Experiments on Musical Tempo Estimation using Different Estimation Algorithms

This section presents the experimental procedure, specifics and results done for testing the performance of six different tempo estimation algorithms.

5.3.1 Conditions and Specifications for the Experiments

5.3.1.1 Error Calculation

The algorithms were tested genre-wise with the average errors being calculated for each genre separately. Errors are taken as root-mean-squared values of deviations from the ground truth values. This RMSE represents the sample standard deviation of the differences between predicted values and observed values.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left[\hat{t}(n) - t(n)\right]^2} \tag{5.1}$$

where RMSE stands for Root Mean Squared Error, $\hat{t}(n)$ is the predicted tempo for song number n, $t(n)$ is the ground truth tempo for song number n, and N is the total number of songs being tested for. All octave errors have been ignored while calculating the RMSE.

5.3.1.2 Output Specifications

Dan Ellis's open source code, downloaded from the following url: `http://labrosa.ee.columbia.edu/projects/beattrack/`, estimates the tempo of the audio waveform and returns three values: two tempo estimates in BPM as t(1) (slower) and t(2) (faster), with the relative strength of t(1) given by the third value t(3) (i.e. t(1) is the preferred tempo if t(3) $>$ 0.5) [34]. The preferred tempo is taken for error calculations in this thesis (i.e., the output for which the strength is higher).

5.3.2 Experiment Results

5.3.2.1 Results for Tempo Estimation on Western Classical Music Dataset

It can be observed here that the MFCC method works best whereas the spectral centroid method gives the least favourable result. Spectral centroid works poorly as WC lacks in percussive sounds and thus spikes in spectral centroid at onset positions are less likely. The sense of rhythm in this type of music is mostly mediated by gradual rises in amplitude or gradual changes in pitch, which is effectively picked up by MFCCs and envelope methods. This is also why sub-band synchrony does not work well in this case: because of this slow attack rate of instruments in WC, the derivative of the envelopes are not peaky enough to give a clear onset envelope. The tempo estimation error analysis for this genre is given in Table 5.2.

5.3.2.2 Results for Tempo Estimation on Indian Classical Music Dataset

Indian classical music has an advanced rhythmic framework which revolves around the concept of tala, where the rhythmic structure is hierarchically described at multiple time-scales. A complete description of rhythm in Indian Classical music traditions - both Hindustani and Carnatic, would need a rhythm model which can analyze music at these different time-scales

Method	RMSE
Fluctuation Strength	10.1878
Amplitude Envelope Onsets	11.2212
MFCC Onsets - Dan Ellis	**9.5454**
Spectral Centroid	17.8779
Sub-Band Synchrony (Using Mean)	14.7091
Sub-Band Synchrony (Using Variance)	16.3786

Table 5.2: Error values for tempo estimation in Western Classical dataset

and provide a musically relevant description [35]. Because of these complex metrical levels, tempo estimation using only onsets is a tough task and we get high error values for this genre. The tempo estimation error analysis for this genre is given in Table 5.3.

Method	RMSE
Fluctuation Strength	**30.0704**
Amplitude Envelope Onsets	53.6486
MFCC Onsets - Dan Ellis	47.4959
Spectral Centroid	31.0531
Sub-Band Synchrony (Using Mean)	30.2827
Sub-Band Synchrony (Using Variance)	31.7292

Table 5.3: Error values for tempo estimation in Indian Classical dataset

5.3.2.3 Results for Tempo Estimation on Pop Music Dataset

Owing to the distinctive and simple beat patterns in the popular genre of music, tempo estimation using onsets is particularly suited for pop music, and we get the lowest error values for it. Sub-band synchrony (mean) gives the best results followed by spectral centroid and MFCC. Amplitude envelope mehod also works well in this method because of the high dynamic range in most of the songs and sharp amplitude peaks at onset locations. The tempo

estimation error analysis for this genre is given in Table 5.4.

Method	RMSE
Fluctuation Strength	16.4372
Amplitude Envelope Onsets	16.1237
MFCC Onsets - Dan Ellis	8.2318
Spectral Centroid	7.6097
Sub-Band Synchrony (Using Mean)	**7.5220**
Sub-Band Synchrony (Using Variance)	11.2408

Table 5.4: Error values for tempo estimation in Pop Music dataset

5.3.2.4 Results for Tempo Estimation on Rock Music Dataset

Similar to pop, rock also has a uniform rhythm in most cases, but may be considered slightly complex as some songs contain variations in the beat. An interesting thing to note here is that spectral centroid works best, followed by sub-band synchrony. In both pop and rock datasets, spectral centroid and synchrony are expected to work well because of the distinctive percussion hits that span a wide spectral bandwidth, thus giving good centroid peaks as well as synchronous onsets. The tempo estimation error analysis for this genre is given in Table 5.5.

Method	RMSE
Fluctuation Strength	23.9632
Amplitude Envelope Onsets	20.8582
MFCC Onsets - Dan Ellis	21.0034
Spectral Centroid	**9.3702**
Sub-Band Synchrony (Using Mean)	13.6702
Sub-Band Synchrony (Using Variance)	16.1414

Table 5.5: Error values for tempo estimation in Rock Music dataset

5.3.2.5 Results for Tempo Estimation on LabROSA Music Dataset

This dataset consisted of a mixed set of songs from different genres. Sub-band synchrony seems to perform the best in this set as well. The tempo estimation error analysis for this genre is given in Table 5.6.

Method	RMSE
Fluctuation Strength	24.0897
Amplitude Envelope Onsets	27.7308
MFCC Onsets - Dan Ellis	22.2798
Spectral Centroid	9.3069
Sub-Band Synchrony (Using Mean)	**7.1230**
Sub-Band Synchrony (Using Variance)	11.0442

Table 5.6: Error values for tempo estimation in LabROSA dataset

5.3.2.6 Results for Tempo Estimation on Test Dataset

Spectral methods perform better than the temporal method, according to our expectations, and the best RMSE value is achieved by sub-band synchrony, because of its effectiveness in picking non-percussive note onsets. The tempo estimation error analysis for this genre is given in Table 5.7.

Method	RMSE
Fluctuation Strength	13.5756
Amplitude Envelope Onsets	27.4740
MFCC Onsets - Dan Ellis	16.6500
Spectral Centroid	15.3895
Sub-Band Synchrony (Using Mean)	**6.6822**
Sub-Band Synchrony (Using Variance)	10.0792

Table 5.7: Error values for tempo estimation in Test dataset

5.4 Results Summary

The overall error values for all the songs tested are given in table 5.8. Sub-band synchrony (using mean) performs the best followed by spectral centroid and then sub-band synchrony (using variance). A discussion on the results is provided in the next section.

	Datasets						
Methods	WC	IC	Pop	Rock	Ellis	Test	Overall
FS	10.1878	**30.0704**	16.4372	23.9632	24.0897	13.5756	20.6462
AE	11.2212	53.6486	16.1237	20.8582	27.7308	27.4740	27.1749
MFCC	**9.5454**	47.4959	8.2318	21.0034	22.2798	16.6500	22.7642
SC	17.8779	31.0531	7.6097	**9.3702**	9.3069	15.3895	15.6174
SBS(M)	14.7091	30.2827	**7.5220**	13.6702	**7.1230**	**6.6822**	**14.8856**
SBS(V)	16.3786	31.7292	11.2408	16.1414	11.0442	10.0792	17.0614

Table 5.8: Results Summary for methods Fluctuation Strength (FS), Amplitude Envelope (AE), MFCCs (MFCC), Spectral Centroid (SC), Sub-band Synchrony using mean (SBS(M)), Sub-band Synchrony using Variance (SBS(V)).

For better visual comparison, a chart of errors in all datasets is given in Figure 5.6.

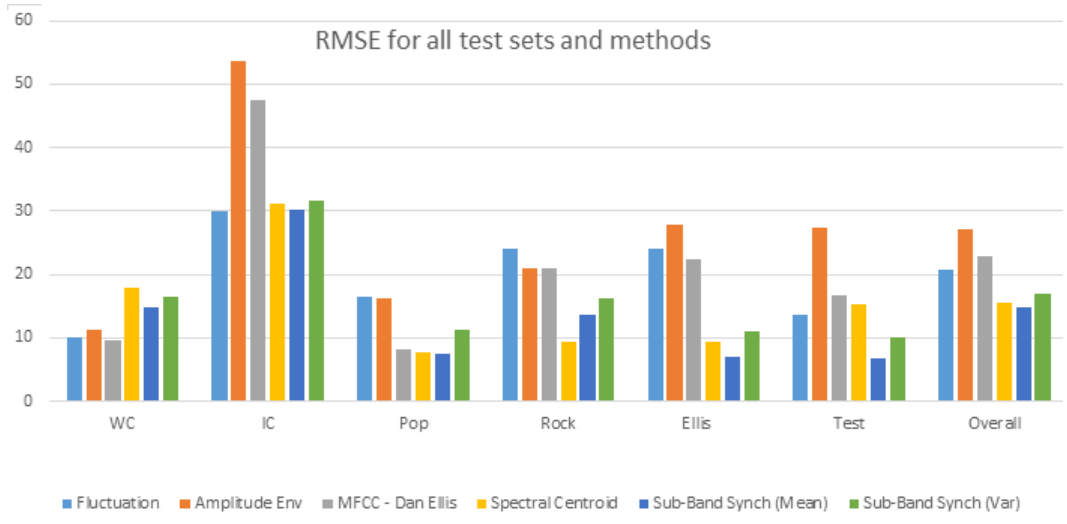

Figure 5.6: Graphical visualization of error in different datasets for different methods

5.5 Discussion

In this chapter a detailed study of the performance of six different tempo estimation algorithms was done on various different datasets. The datasets were generated keeping in mind the diversity of musical styles and the large variability of tempo values among them. Manually calculating tempo for songs is a time-consuming task, which restricted the size of dataset that could be used. Better tests would entail larger test sets and more number of annotated audio files. No such freely available human-annotated dataset was found upon search. Thus the need to generate a dataset of our own.

In the Tables 5.2 to 5.7 RMSE error values were shown for six methods on the six datasets. Although there appears to be no such algorithm that gives the best results in all the datasets, the overall error values suggest that the proposed method of sub-band synchrony gives the least error and is thus a superior method of tempo estimation than onset detection using spectral features (MFCCs) or using amplitude envelope in general.

There appears to be largest errors in the Indian Classical dataset. It can be attributed to the non-uniform style of tabla playing in this genre and complex metrical structures and varying tempo.

The test dataset clearly shows that when envelope information is not reliable, onset detection using pitch changes as the feature can also be done using sub-band synchrony. That is because pitch changes reflect as magnitude changes in different sub-bands. Sub-band synchrony was effective in detecting tempos from this dataset with a good accuracy and yield the lowest error values among other methods.

Sub-band synchrony also performs better on Dan Ellis's dataset than any other method, including Dan Ellis's algorithm. It must be noted that the Ellis dataset contained annotated audio with onsets and tempo marked. The tapping experiment was not performed for this dataset and the marked values taken directly as ground truth tempo values.

The overall results in Table 5.8 shows that sub-band synchrony (using mean) in fact does have the best values among other algorithms for tempo estimation. The average error using this method is around 7.88 BPM lesser than the best existing algorithm tested. The average tempo of the entire dataset used is 129.89 BPM. That amounts to an improvement of more

than 6%. Spectral centroid method also comes close in terms of error values with a 5.5% improvement.

5.6 Summary

In this chapter, a description of the database used to test the accuracy of tempo estimation algorithms is first given, followed by the tapping experiment used to determine ground truth perceptual tempo values. Next, error values obtained after tempo estimation for each method and for each dataset are provided and attempts are made to hypothesize why some methods work better in certain genre datasets and others do not. It is observed finally, that although there appears to be no single algorithm that works best in all datasets, the proposed method of sub-band synchrony (using mean) gives the overall least error values and can be concluded to perform better than existing tempo estimation methods based on onset detection using MFCCs or using amplitude envelope.

Chapter 6

Conclusions and Future Scope

6.1 Conclusions

The aim of this work was to investigate new methods of tempo estimation from only an audio input. Three new methods have been proposed in this thesis concerning the same: the fluctuation strength, the spectral centroid method, and the sub-band synchrony method, and are the primary contributions of it. The fluctuation strength estimates the tempo from a 2-dimensional rhythmic feature computed on a human auditory model (that divides the frequency range into Bark bands to mimic human auditory system) by calculating the strength of amplitude modulation of the sound at different Bark bands. The remaining two methods - spectral centroid, and sub-band synchrony - generate the onset curve of the test sound, and then computes the tempo based on the frequency content of occurrence of onsets. Among these, sub-band synchrony was found to give the best performance, on a comparative testing of the algorithms on a dataset containing different musical styles and varying tempos. The sub-band synchrony method also outperformed other traditional methods of tempo estimation on the same dataset (a 6% increase in accuracy was achieved from the best existing algorithm tested). Therefore it can be concluded that the method of sub-band synchrony introduced in this thesis is an effective method of tempo computation.

6.2 Future Scope

An overall error rate of 14.88 BPM that was obtained using sub-band synchrony algorithm is still not good enough to make effective use of the algorithm at applications that require precise tempo estimates like music production and remixing, and any application that requires synchronizing a given piece of music to another. A better tempo estimation algorithm might arise from machine learning methods. Using signal processing to extract relevant features, and using them as inputs to a supervised machine learning algorithm, like support vector machines, can give more accurate results for onset detection. Neural network also promises to be a future endeavour to undertake when the limits of signal processing for onset detection are reached and models that mimic human perception are required, since music is essentially a human perception. Schluter [36] uses convolutional neural networks for improved onset detection and is the state-of-the-art in this field. Machine learning approaches might outperform signal processing methods in the future.

A possible application of onset detection is rhythm extraction at different metrical levels. Currently, extracting a rhythmic pattern is a difficult task because rhythm consists of layers of metrical levels and that makes even human subjects differ from one another on what they think the rhythm pattern is. A scheme that can estimate the rhythmic structure at different metrical levels will cover the entire domain of rhythmic structures and can be subsequently used to, among other applications, calculate a perceptual complexity measure as a feature of comparing the similarity of musical pieces. An interesting investigation using this complexity feature can be to study the evolution of music in terms of complexity and determine trends that have led to the development of various musical styles that are seen in the present day.

Appendix A

Tempo Prior Probability Distribution

A priori probabilities of the occurrence of tactus periods has been measured by several authors and has been used to calculate posteriors by multiplying with likelihood funcions of tactus periods. As suggested by Paulus and Klapuri in [37], and Parncutt in [38], we apply log-normal distribution for tactus periods:

$$P_0(\tau) = \frac{1}{K} \exp\left\{ -\frac{1}{2\sigma^2} \left[\log_{10}\left(\frac{\tau}{\mu}\right) \right]^2 \right\} \tag{A.1}$$

Here μ denotes *moderate pulse period* and is typically around 600 ms; and σ is the standard deviation of the logarithm of the pulse period and typically has a value of about 0.2. $\frac{1}{K}$ is the normalization constant with

$$K = \int_0^\infty P_0(\tau) \mathrm{d}\tau \tag{A.2}$$

The plot of the distribution is given in Figure A.1.

This distribution can be converted into corresponding tempo values and that has been used as the scaling functions for various tempo likelihood functions. The plot of the distribution with respect to tempo in BPM is given in Figure A.2.

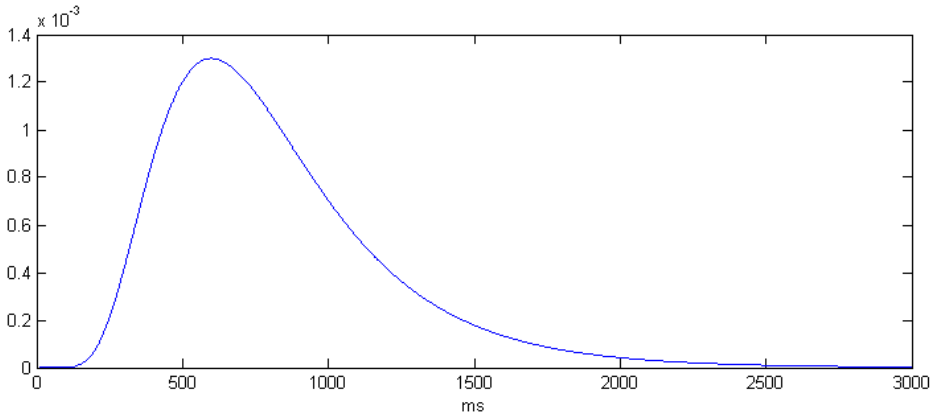

Figure A.1: *A priori* probability distribution of tactus periods

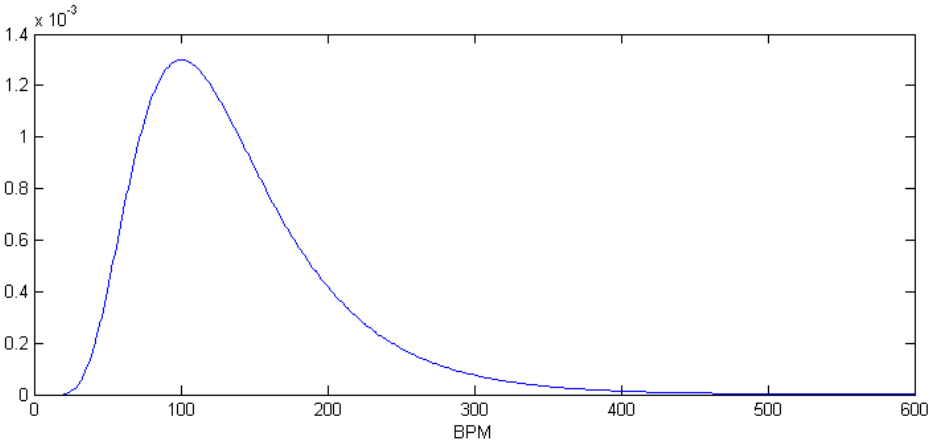

Figure A.2: *A priori* probability distribution of tempo

References

[1] O. Lartillot, "Mirtoolbox 1.3. 4 users manual," *Finnish Centre of Excellence in Interdisciplinary Music Research, University of Jyväskylä, Finland*, 2011.

[2] A. Klapuri and M. Davy, *Signal processing methods for music transcription.* Springer Science & Business Media, 2007.

[3] J. S. Downie, "Music information retrieval," *Annual review of information science and technology*, vol. 37, no. 1, pp. 295–340, 2003.

[4] J. London, *Hearing in time: Psychological aspects of musical meter.* Oxford University Press, 2012.

[5] M. R. Jones and M. Boltz, "Dynamic attending and responses to time.," *Psychological review*, vol. 96, no. 3, p. 459, 1989.

[6] S. Quinn and R. Watt, "The perception of tempo in music," *PERCEPTION-LONDON-*, vol. 35, no. 2, p. 267, 2006.

[7] D. Moelants, "Perception and performance of aksak metres," *Musicae Scientiae*, vol. 10, no. 2, pp. 147–172, 2006.

[8] J. Snyder and C. L. Krumhansl, "Tapping to ragtime: Cues to pulse finding," *Music Perception*, vol. 18, no. 4, pp. 455–489, 2001.

[9] D. Epstein, *Shaping time: Music, the brain, and performance.* Wadsworth Publishing Company, 1995.

[10] C. Drake, L. Gros, and A. Penel, "How fast is that music? the relation between physical and perceived tempo," in *Proc. Int. Conf. Music Percept. Cognit*, 1999.

[11] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A tutorial on onset detection in music signals," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 5, pp. 1035–1047, 2005.

[12] S. Dixon, "Onset detection revisited," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06)*, pp. 133–137, Citeseer, 2006.

[13] H. Laurent and C. Doncarli, "Stationarity index for abrupt changes detection in the time-frequency plane," *Signal Processing Letters, IEEE*, vol. 5, no. 2, pp. 43–45, 1998.

[14] M. Davy and S. Godsill, "Detection of abrupt spectral changes using support vector machines. an application to audio signal segmentation," in *ICASSP*, vol. 2, pp. 1313–1316, Citeseer, 2002.

[15] F. Desobry, M. Davy, and C. Doncarli, "An online kernel change detection algorithm," *Signal Processing, IEEE Transactions on*, vol. 53, no. 8, pp. 2961–2974, 2005.

[16] S. Abdallah and M. D. Plumbley, "Unsupervised onset detection: a probabilistic approach using ica and a hidden markov classifier," in *Cambridge Music Processing Colloquium*, 2003.

[17] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *Speech and Audio Processing, IEEE transactions on*, vol. 10, no. 5, pp. 293–302, 2002.

[18] E. W. Large, "Beat tracking with a nonlinear oscillator," in *Working Notes of the IJCAI-95 Workshop on Artificial Intelligence and Music*, vol. 24031, 1995.

[19] J. Laroche, "Efficient tempo and beat tracking in audio recordings," *Journal of the Audio Engineering Society*, vol. 51, no. 4, pp. 226–233, 2003.

[20] D. Fitzgerald, "Harmonic/percussive separation using median filtering," 2010.

[21] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, vol. 1, pp. 452–455, IEEE, 2000.

[22] X. Serra and J. Smith, "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, pp. 12–24, 1990.

[23] S. N. Levine, *Audio representations for data compression and compressed domain processing*. PhD thesis, Citeseer, 1998.

[24] W. A. Schloss, *On the automatic transcription of percussive music: from acoustic signal to high-level analysis*. No. 27, Stanford University, 1985.

[25] X. Rodet and F. Jaillet, "Detection and modeling of fast attack transients," in *Proceedings of the International Computer Music Conference*, pp. 30–33, 2001.

[26] P. Masri, *Computer modelling of sound for transformation and synthesis of musical signals*. PhD thesis, University of Bristol, 1996.

[27] E. Terhardt, "On the perception of periodic sound fluctuations (roughness)," *Acta Acustica united with Acustica*, vol. 30, no. 4, pp. 201–213, 1974.

[28] H. Fastl, "Fluctuation strength and temporal masking patterns of amplitude-modulated broadband noise," *Hearing Research*, vol. 8, no. 1, pp. 59–69, 1982.

[29] E. Pampalk, *Islands of music: Analysis, organization, and visualization of music archives*. na, 2001.

[30] H. Fastl, "Fluctuation strength of modulated tones and broadband noise," in *Hearing-Physiological Bases and Psychophysics*, pp. 282–288, Springer, 1983.

[31] J. M. Grey and J. W. Gordon, "Perceptual effects of spectral modifications on musical timbres," *The Journal of the Acoustical Society of America*, vol. 63, no. 5, pp. 1493–1500, 1978.

[32] G. Peeters, "{A large set of audio features for sound description (similarity and classification) in the CUIDADO project}," 2004.

[33] R. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice, "An efficient auditory filterbank based on the gammatone function," in *a meeting of the IOC Speech Group on Auditory Modelling at RSRE*, vol. 2, 1987.

[34] D. P. Ellis, "Beat tracking by dynamic programming," *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, 2007.

[35] A. Srinivasamurthy, G. Tronel, S. Subramanian, and P. Chordia, "A beat tracking approach to complete description of rhythm in indian classical music," in *Proc. of the 2nd CompMusic Workshop*, pp. 73–78, Citeseer, 2012.

[36] J. Schluter and S. Bock, "Improved musical onset detection with convolutional neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pp. 6979–6983, IEEE, 2014.

[37] J. Paulus and A. Klapuri, "Measuring the similarity of rhythmic patterns.,"

[38] R. Parncutt, "A perceptual model of pulse salience and metrical accent in musical rhythms," *Music perception*, pp. 409–464, 1994.