Real Time Tempo Analysis of Drum Beats

Author: **Philip Hannant**, Supervisor: **Professor Steve Maybank**

Birkbeck, University of London
Department of Computer Science and Information Systems

Project Report
MSc Computer Science

August, 2016

Contents

Abbreviations

BPM	Beats Per Minute
DWT	Discrete Wavelet Transform
FT	Fourier Transform
JSON	Javacript Object Notification
IDE	Integrated Development Environment
STFT	Short-Time Fourier Transform
TDD	Test Driven Development

Definitions

Acoustic Drum Kit A collection of drums and cymbals which do not have electronic amplification. Typically made up of a bass drum, snare drum, tom-toms, hi-hat and 1 or more cymbals.

Beat For the purpose of this project a beat will be defined as the sequence of equally spaced pulses used to calculate the tempo being played by the drummer.

Downbeat Refers to beat one of a measure of music, called a downbeat to correspond to the motion a conductor's arm [1].

Drum Module The device which serves as a central processing unit for an electronic drum kit, responsible for producing the sounds of the drum kit.

Electronic Drum Kit An electrical device which is played like an acoustic drum kit, producing sounds from a stored library of instruments and samples.

Measure/Bar A measure or bar is a segment of time made up by a predetermined number of beats, for example a piece of music with a 4/4 time signature will have 4 beats in every measure/bar [2].

MIDI Musical Instrument Digital Interface is a protocol developed in the 1980's to allow electronic instruments and other digital musical tools to communicate with each other [3].

Time Signature Used in musical notation to represent the number of beats in a measure or bar of music [4].

1 Introduction

This project report presents my aim to develop a real-time drum beat tempo analysis system using different beat detection algorithms which is able to record the perfomance of each method concurrently when an extensive set of drum samples, representing a real drummer's perfomance, is processed through the system.

1.1 Drumming Training Tools Background

Timing is the fundamental skill any good drummer should possess and is the staple by which they will be judged. For many years the only training tool available to a drummer to improve their timing was the metronome. An instrument used to mark musical tempo, erroenously attributed to Johann Nepomuk Maelzel in 1815 but was actually invented by a Dutchman, Dietrich Nikolaus Winkel a year earlier. The traditional metronome, based on Winkel's original design is a hand-wound clockwork instrument that uses a pendulum swung on a pivot to generate the ticking which depicts the desired tempo [1] is still used today by musicians.

For drummers however the electronic versions of the metronome are much more widely used, to the point that metronomes are now developed with functionality specifically tailored to a drummers training requirements. The Tama Rhytham Watch was the first metronome designed specifically for drummers, providing enough volume to be used with real drums as well as allowing for the use of different time signatures and preset set rhythm patterns to help improve perfomance. (Tama rhythm watch image)

Following the development of MIDI driven electronic drum kit came the development of more advanced training tools that now were able to provide live feedback to drummer during any given perfomance. Today the leaders in this field are Roland, their v-drums line provide a variety of tutition packages including the SCOPE and more recently the COACH system provided in the v-drum modules, the v-drum Rhymthm Coach line is an advanced version of the traditional drummers practice pad and the extensive DT-1 V-Drums tutor software package. Roland have even now gamified this field with their latest release, the V-Drums Friend Jam app. The application itself provides the player with live feedback and evaluates each performance in order to provide the player with a score which they can share over social media.

The aim of this project is to investigate whether some of the current beat detection algorithms available would be accurate enough to provide the basis for a training tool for dummers using an acoustic drum kit as opposed to an electronic drum kit.

1.2 Drum Musical Theory

The theory set out in this section provides only a basic review of some of the key characteristics used in drum music alone.

1.2.1 Notation

In order to understand the fundamentals of musical timing some theory needs to be examined. Music is written on staff that is made up of five individual lines, the clef is found on the far left of the staff which indicates the pitch of the notes [2], as percussion instruments are non-pitched they use the percussion-clef. On traditional musical notation the lines and spcaes between represent a tonal where as for drum notation notes written on lines or spaces indicate a certain drum or cymbal. The staff is seperated into individual measures which are known as bars [3] and it is these bars that are the basis of musical time.

1.2.2 Notes

The notes dep

1.2.3 Time Signatures

Time signatures appear on the staff just after the clef and are written as a fraction where the top number indicates the number of beats that there are in a bar. With the bottom number representing the size of the note that makes up the duration of one beat. For example the 4/4 or common time signature indicates 4 beats in each bar or measure where each beat is made up of 1 quarter note.

1.3 Beat Detection Background

Detecting musical time is a skill which is not only fundamental to musicians [6] but also something that seemingly comes naturally to humans. The majority are able to analyse and reproduce the metre[1], tempo and rhythmic aspects of a piece of music [7]. Longuet-Higgins [6] was one of the first to produce an algorithm to replicate this human ability. He constructed a binary tree with each node representing a note or rest [8]. He then developed his theory into a system that was able to measure the variations in the downbeats of a piece of music and adjust the perceived tempo according to whether the note was later or earlier than expected [6]. Since Longuet-Higgins' first work there have been a number of different approaches to beat detection. M. Goto and Y. Muraoka developed a system which learned the frequencies of the bass drum and snare drum, in order to detect events triggered by these instruments during a piece of music [9]. In 2001, Simon Dixon presented Beatroot, an interactive beat tracking and visualisation system which is able to estimate the tempo and times of musical beats in performed music [10]. In the same year Tzanetakis *et al* [11] described an algorithm based on the Discrete Wavelet Transform (DWT) which is capable of detecting the beat attributes of music.

1.3.1 Beatroot

Beatroot is an audio beat detection system presented by Simon Dixon in 2001 and is described as a "beat tracking system which finds the times of musical beats and tracks changes in tempo throughout a performance" [13]. Beatroot works by first processing digital audio data to produce a list of onset (see Figure 2) or event times. The time intervals between these events are then analysed to generate tempo hypotheses concerning the rate and location of beats. Using the tempo hypotheses, searches are carried out to test the different hypotheses about the rate and timing of beats. The results of these searches are ranked and the beat times found in the highest ranked search are returned [10].

1.3.2 Short-Time Fourier Transform

To detect the onsets of a piece of digital audio Beatroot uses an onset detection function based on the Short-Time Fourier Transform (STFT). The STFT is a form of Fourier transform (FT) which was developed by Joseph Fourier in 1822 [16], which can be used to find out how much of each frequency exists in a signal. A drawback of the FT is that it is unable to provide any details of when a frequency component occurs in time for non-stationary signals[2]. A solution to this was to split a non-stationary signal up into a number of smaller segments using a window function, which effectively created a series of stationary[3] signals which the FT could then be applied to. However,

[1] Metre is the repeating pattern that provides the pulse of beat of a piece of music [12]

[2] Non-stationary signals are signals whose frequency contents changes over time [17].

[3] The frequency contents of a stationary signal does not change over time

this did not fully solve the problem as the size of window function affects the quality of frequency resolution and time resolution:

- Narrow Window Function $\longrightarrow$ Good Time Resolution, Bad Frequency Resolution
- Wide Window Function $\longrightarrow$ Bad Time Resolution, Good Frequency Resolution [17]

1.3.3 Discrete Wavelet Transform

The first literature regarding the wavelet was provided by the mathematician Albert Haar in 1909 [18]. The wavelet transform is a technique for analysing signals which was developed as an alternative to the STFT [11]. Like the STFT, the DWT is able to provide time and frequency information, however, unlike the STFT the DWT is able to do this without the need for a window function.

1.3.4 Tzanetakis *et al* Beat Detection Method

In 2001, Tzanetakis *et al* described how the Discrete Wavelet Transform (DWT) could be used to extract information from non-speech audio [11]. Their beat detection algorithm was based on detecting the most prominent signals which are repeated over a period of time within the analysed audio and was is split into the following steps:

1. Signal decomposed into a number of octave frequency bands using the DWT
2. The subsequent time domain information is extracted for each frequency band
3. The data from each band are then summed together and a function to find repeating patterns is applied

As there is no current open source Java implementation of this algorithm. I will attempt to implement a version of this myself, which will be the DWT based beat detection component of this project.

1.4 Proposed Architecture

1.5 Live Audio Processing

The live audio will be processed using the Javax Sound package. The audio will be captured using a stereo microphone and processed to match CD quality with the Javax Sound AudioFormat class. The Beatroot system was not originally intended to be used as a real time system [19] so currently only works with prerecorded audio. It will therefore be will need to be modified in order for it to work with live audio.

1.6 Implementation of Tzanetakis *et al* Beat Detection Algorithm

The beat detection algorithm described by Tzanetakis *et al* [11] is based on detecting the most prominent periodicities of a signal and is made up of the following stages:[4]

- DWT - First the signal is processed by the DWT into a number of frequency bands
- Low Pass Filtering - a low pass filter is then applied to the signal in order to allow the lower frequencies of the signal to be analysed

[4] A full description of the theory regarding Tzanetakis *et al* beat detection algorithm will be provided in the Project Report.

- Full Wave Rectification - each frequency band is then converted to one constant polarity (positive or negative)[wiki]. A visual representation can be seen in Figure 4
- Downsampling - the sampling rate of the signal is decreased by an integer factor [20]
- Normalisation - each band is then normalised using mean removal
- AutoCorrelation - an autocorrelation function is then applied to the frequency bands, the first five peaks of this function are detected and their periodicities are calculated in beats per minute

If the implementation of this algorithm takes longer than described in the schedule in section 5. The Matlab implementation created by Eng Eder de Souza [21] will be adapted for use to be used in this system.

1.6.1 JWave

The implementation of Tzanetakis *et al* [11] beat detection algorithm will require the use of a library which provides DWT functionality. This will be provided by JWave. JWave is a Java library created by Christian Scheiblich which provides a number of different transform packages [22], including a wavelet package. For the purpose of this project the Daubechies wavelets classes within JWave will be used. The Daubechies wavelets are a form of discrete wavelet transform which were developed by Ingrid Daubechies in the late 1980's and are frequently used in applications [14].

1.7 Comparative Analysis of Beat Detection Algorithms

A key part of this project is to ascertain whether the two beat detection methods described in section 1 are accurate enough to create a training tool similar to the MIDI based software seen in figure 1. To achieve this the two systems will be run in parallel using real time audio and return details on the calculated tempo in beats per minute (bpm), the number of beats detected and the time it took to calculate the results.

The data collected during this process will be recorded and stored in an appropriate format to allow it to be queried and analysed. Due to the relatively small amount of data which will be collected, it is not deemed necessary to create a database to store this. Instead the data will be stored in a JSON file. JSON was chosen over XML as it has a much simpler grammar and is able to map directly onto the data structures used in modern programming languages [23].

1.8 Build Extensive Drum Sample Set

The success of this project will rely heavily on the amount of live audio data that the system processes. A large set of drum beat samples will be required. The drum beats will be created using the Apple software package, Garageband. The sample set will contain beats from a variety of styles as well as samples with fluctuating tempos. The main aspects of the sample set are as follows:

- The tempo range will be 60 - 160 beats per minute. Some samples will be be created with varying tempos in order to model human playing more closely.
- The time signatures will be mainly in common time (4/4) to reflect normal playing styles, however some drum beats will use swing notes[5] which will be created by feel[6].

[5] A swing note is style where notes with equal time are played with an unequal durations, usually as alternating long and short durations [wiki]

[6] By adapting the strength a drum or cymbal is being played at it is possible to infer a different style of music using feel, e.g. swing time can be created this way.

- Accenting[7], ghost notes[8] and missed beats[9] will be added in line with a real playing style.
- The sample set will contain a wide range of drum beats, including; beats using one drum, those only with a back beat, beats incorporating single rests within a bar or for the whole duration of a bar and beats with large solos and loud cymbal hits[10].

The main musical styles will be rock and pop, blues and jazz. Approximately ten unique drum beats will be created for each style, which will then be sampled at a range of bpms. There will also be a final set of simple single and multiple drum beats included. The aim will be to generate a total set of approximately seven hundred and fifty samples.

1.9 User Interface

The user interface will be developed using the JavaFX framework, enabling the system will to be cross-platform compatible. Its features will be kept simple initially, where it will only be required to provide a real time output of the current two tempo calculations and accumulative count of beats detected.

1.10 Non-Core Project Features

The non-core project features will be included only if the project is developed ahead of the schedule described in section 5.

1.10.1 Frequency Learning

This non-core feature will investigate if it is possible to create a similar system to the one developed by Goto and Muraoka, which learnt the frequencies of the snare and kick drum in order to detect events triggered when these instruments were played [6]. The goal will be to enable the system to learn the frequencies of all of the separate parts of a drum kit (snare drum, bass drum, tom-tom, hi-hat and cymbals) being played and use this event to extend the GUI.

1.10.2 Extended User Interface

Developing this additional feature will only be possible if the previous non-core feature (3.7.1) is successfully developed. If so, the user interface will be extended to look similar to the drum kit visual seen in Figure 1, where a colour indicator will be displayed to represent which instrument was last played. The development of this final feature depends heavily on the accuracy and efficiency of the tempo analysis algorithms used.

1.11 Project Methodology

This project will be developed using the Test Driven Development (TDD) framework. TDD is a development process that combines the test-first development and refactoring. Test-first development involves writing a test before just enough production code is written to fulfill that test [25]. Refactoring is the process of making small changes to code in order to improve its design, making it easier to understand and modify [26]. I intend to use TDD due to the relatively short development time of this project. TDD will enable small steps to be taken during the process which is considered to be a more productive approach to writing software [25].

[7] Accents are an emphasis placed on a specific note

[8] Ghost notes are a note which is played at a very low volume

[9] Missed beats will be used to represent the drummer missing a drum or cymbal

[10] When a cymbal is played at high amplitudes (loudness) the vibrations it create become chaotic and its clearly identifiable signals disappear and it effectively becomes noise [24]. It will therefore be interesting to see if this noise affects the accuracy of the algorithms in any way.

1.12 Development Languages and Testing

The project will be developed in a combination of Java and Scala. Java is required as both the Beatroot and JWave libraries are in this language, however any functional aspects of the project will be developed in Scala, as it offers a superior functional tool set when compared to those offered in Java 8. As previously discussed JSON will be used to store the collected data.

I will use the Intellij IDE to develop this system. Git will be used for version control, testing will be completed using JUnit for Java, and Scala Test for any functionality developed in Scala. In order to ensure that the parsed JSON is well formed it will be tested using the JSON validator found at - http://jsonlint.com/.

References

[1] *https://www.britannica.com/art/metronome*

[2] Alison Latham *The Oxford Companion to Music, 2002, Oxford University Press*

[3] *http://www.drummagazine.com/lessons/post/drumkey/*