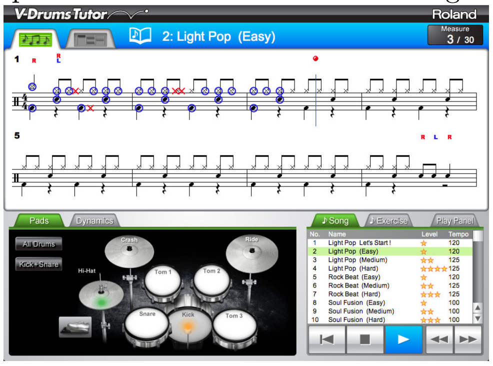
Proposal

Philip Hannant

March 28, 2016

1 Introduction & Background

Having played the drums for nearly twenty years I have experienced a number of different training tools in order to improve my timing, these tools have always been exclusive to midi driven electric drum kits. These drum kits are able to include training tools within their drum modules which utilise the midi events being triggered by the player and give live feedback to the exact timing and instrument being played. An example of such a system is the Roland DT-1 V-Drums tutor which is a software package that can be connected to their V-Drum modules in order to provide the player with an interactive experience in order to help a drummer improve their rhythm, coordination sight reading. Such an extensive and accurate range of tutoring functions is only made possible by the midi events that are intrinsic to an electronic drum kits function. (an example of the GUI can be found in figure 1) [roland ref]. These tutoring tools are not however available to a drummer who does not own an electronic drum kit, they are restricted to using a metronome and using their ear to determine their timing and rhythm. To address this, I intend to investigate the current audio tempo analysis and beat tracking algorithms available solely for the purpose of being implemented within an drum tutoring software package for the use with acoustic drums. 6, 3, 7, 2,

1 

No.	Name	Level	Tempo
1	Light Pop Let's Start !	★	120
2	Light Pop (Easy)	★	120
3	Light Pop (Medium)	★★	125
4	Light Pop (Hard)	★★★★	125
5	Rock Beat (Easy)	★	120
6	Rock Beat (Medium)	★★	125
7	Rock Beat (Hard)	★★★	125
8	Soul Fusion (Easy)	★	100
9	Soul Fusion (Medium)	★★	100
10	Soul Fusion (Hard)	★★★	100

1.1 Current Systems

Detecting musical time is a skill which is not only a fundamental musical skill [1] but also something that can seemingly come naturally to humans, the majority being able to analyse and reproduce discrete metrical stuctures of a piece of music [2]. Producing algorithms to replicate this nature human ability is probably first attempted by Longuet-Higgins [1], where he began to consider that rhythm as a binary tree with each node representing a note or rest. This theory developed into a system which would use a static tolerence limit on how much a the downbeats varied and enabling the perceived tempo to be adjusted accordingly [add longuet references]. Since Longuet-Higgins first work there have been a number of different approaches to beat detection in audio, M. Goto and Y. Muraoka created a system which would learn the frequencies of the bass drum and snare drum, in order to then detect events triggered by these instruments during a piece of music [Goto & muroaka]. In 2001 Simon Dixon presented a system which uses processed a piece of audio in order to generate a tempo hypotheses at various metrical levels, multiple agents were then employed to find the sequence of beat times which best matched the salient rhythmic events which were originally detected.

The majority of the audio beat detection systems developed to date rely on an onset detection function, where an audio onset is considered to be the beginning of each music note within a piece of music[wiki onset] and therefore can be used to find time-locations within all sonic events in a piece of music [MIREX].

systems developed to perform beat tracking and temporal analysis with the majority using the "surfboard" method first described by schloss which observed the peaks of sound energy within a piece of music in order to discern the beat locations and temporal information [schloss].

This method however was not considered accurate enough to fully replicate the skill of a trained musician at beat detection, it was soon recognised that onset detection would play a fundamental part in any future beat detection algorithms. Where the an indicator of a new onset is seen as "an increase in energy (or amplitude) within some frequency band(s)" [4].

Need to include the different types of

Due to the rapidly expanding research being carried out on beat detection, in 2005 the 1st annual Music Information Retrieval Evaluation eXchange (MIREX) was held. MIREX has been set up as a contest with the goal of comparing state-of-the-art algorithms and music information retrieval [MIREX website]. The topics to be evaluated were proposed by the participants and in the first year, three of the nine topics concerned beat detection (Audio Drum Detection, Audio Onset Detection and Audio Tempo Extraction).

Find the reference regarding how much work has been done in this field

Light version of beatroot and DWT

Needs to add history of all the work carried out on audio analysis with an emphasis on tempo detection

Beat detection and tempo analysis is a fundamental skill honed by musicians and in particularly drummers, their ability to perform at a consistent tempo determines a piece of musics rhythm.

To date, the majority beat detection software has focused mainly on the dj industry where tempo and beat matching are popular aspects to be included in modern dj programs. There are however other sectors of the music industry would find beat detection software a useful, in particular, drummers could find it an enormously useful training tool. There are currently many systems available to a drummer,

1.2 Beatroot and JWave

2 Aims and Objectives

As part of my project I propose to build a real time drumbeat tempo analyser, this will use a combination of existing libraries in order to provide a basis for the creation of a live drumming tempo training tool. I will elaborate on the key features of this work in the following sections.

which wiill implement a variety of different wave analysis algorithms. sound energy analysis and discrete wavelet transform technologies. I will provide more details regarding the system architecture in the following sections.

Investigate with a sole focus on drums how accurate/reliable the beatroot and discrete wavelet transform algorithms are in order to ascertain how viable a future mobile metronome and drum tempo training application would be.

2.1 Core Project Features

- Live audio capturing and processing

- Log detailed comparative information regarding the efficiency and accuracy of the Beatroot and DWT implemntation using JWave
- GUI to provide the user with current tempo played and notification of beat detection
- Build and test an extensive sample set of drum beats at varying tempos and velocities in order to ensure the system is tested against as close to a real drummer as possible

2.2 Non-Core Project Features

- Live tempo status bar created and tested with a live player to demostrate the systems ability to handle varying tempos
- Create a calibration mode within the system in order to allow for the system to learn the relevant frequencies of all of the drums being played
- Extend GUI to provide the user with real time notification on when a certain drum is played by using the stored calibrated frequencies

3 Development Plan for the Solution

The first part of this project involves the setting

3.1 Live Audio Processing

In order to capture and process live audio the Javax Sound package will be ulitlised, where the audio will be captured using the Java Line hierachy; specifically the TargetDataLine class, which requires an instance of the Javax Sound AudioFormat class to be parsed in to work. The AudioFormat class is made up of a number of constructed fields which are defined as follows:

- Encoding - This will be set to “”PCM.signed“”, representing audio encoded to the native linear pulse code modulation, where quantization levels are linearly uniform [wiki].
- Sample Rate - 44,100, set to match CD quality for the number of analogs samples which will be analysed per second.
- Sample Size in Bits - 24, based on a sound card with a 24 bit sample depth.
- Channels - 2, audio will be captured using a stereo microphone.
- Frame Size - 6, where frame size equals the number of bytes in a sample multiplied by the number of channels.
- Frame Rate - 44,100, same as sample rate.
- Big Endian (boolean) - false, as the project will be developed on an Intel core which uses a little endian architecture. Where endianess refers to the order of bytes which make up a digital word, with big endianess storing the most significant byte at a certain memory address and the remaining bytes being stored in the following higher memory addresses. The little-endian formate reverses the order storing the least significant at the lowest and most siginicant at the highest memory address [https://en.wikipedia.org/wiki/Endianness].

3.2 Comparative analysis of Beatroot and JWave

A key part of this project is to ascertain whether the technologies currently available are accurate enough to create a training tool similar to the midi based software described in section 1. To achieve this the 2 libraries will need to be run concurrently with the same captured audio

3.3 Develop GUI & logging system

3.4 Build Extensive drum sample set

The success of this project will rely heavily on the amount of data that the system can produce and in order to do this a large set of drum beat samples will need to be created. The drum beats themselves will be created using the installed midi drumbeats with the Apple software package, Garageband. The sample set will need to contains beats from a variety of styles as well as samples with fluctuating tempos. The main aspects of the sample set can be found below:

- The tempo range will be 60 - 180 bpm, although some samples will be be created with varying tempos in order to closely reflect a real human player
- The time signatures will be mainly in common time (4/4) to reflect playing styles, however some beats will use swing time which will either be created by feel within the sample or by using triplets with a duple meter in 12/8
- Velocities will be varied across the sample set, accent and ghost notes will be used 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 & beats with large solos and cymbal work

4 Project Schedule

5 References

http://www.roland.co.uk/blog/exploring-roland-dt-1-v-drums-tutor-software/ https://en.wikipedia.org/wiki/Pu
code$_m$*odulationhttp : //www.jsresources.org/faq*$_a$*audio.htmlframe*$_r$*atehttps : //en.wikipedia.org/wiki/Onse*
*//www.music − ir.org/mirex/wiki/*2016 : *Audio*$_O$*nset*$_D$*etection*

Table 1: Project Timeline

Dates	**Task**
w/c 13th June	Develop live audio capturing system
w/c 20th June	Set up Beatroot library in development environment
w/c 27th June	Implement DWT tempo and beat detection algorithm in Scala/Java
w/c 4th July	Test systems with single live audio source
w/c 11th July	Develop system to store data generated during tempo analysis
w/c 18th July	Research and build appropriate framework to process captured audio in parallel e.g. akka actors
w/c 1st August	Create sample set of drum beats
w/c 8th August	Test system using full sample set
w/c 15th August	Develop UI
w/c 22nd August	Analyse logged data
w/c 29th August	Write up report
w/c 5th September	Present findings to project supervisor
w/c 12th September	Finalise report
w/c 19th September	Submit report