Biko Agozino

Inferring sequence specification from Drum Rhythms

Computer Science Tripos - Part II

St John's College

April 16, 2016

Proforma

Name:	**Biko Agozino**
College:	**St John's College**
Project Title:	
Examination:	**Computer Science Tripos - Part II, June 2016**
Word Count:	**TODO**
Project Originator:	Dr A. Blackwell & Dr S. Aaron
Supervisor:	Mr I. Herman

Original Aims of the Project

To build a system that is able to infer the intended sequence of pattern of strokes on a drum kit from those that are captured. The system then aims at matching this sequence against possible popular rock songs that were originally performed before the year 2002 regardless of the variations in pressure and timing of each individual stroke and the performance as a whole. The project aims to detail the possibility of using distance metrics on short drum performances as a basis for imitation based querying of drum notation.

Work Completed

The system has been built to infer and extract a candidate sample pattern from MIDI performance data by examining for repeated sequences in a repeated bar performance. This sample pattern is then compared with the database and returns the closest match within some bound. Multiple distance metrics have been implemented to assess the advantages and disadvantages of each in this domain.

A parser has been built to parse informal ASCII Drum Tablature that has been successful in parsing (TODO percent) of the notation to an appropriate format for use in the database that the system queries against.

Special Difficulties

None

Declaration

I, Biko Agozino of St John's College, being a candidate for Part II of the Computer Science Tripos, hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose.

Signed [signature]

Date [date]

Contents

Chapter 1

Introduction

This dissertation describes the implementation and evaluation of an query-by-example method for searching collaborative databases of drum rhythms, transcribed by users, found in pre-2002 contemporary music.

1.1 Motivation

The relationship between computers and music has been apparent since the 1950s when Trevor Pearcey and Maston Beard pioneered what was likely the first computer capable perform music, CSIRAC[1], which was able to broadcast music programmed on punched-paper data tape in a similar notation to standard musical notation[14]. At the time it took specialised, multimillion-dollar, computers hours or even days to generate a few minutes of simple tunes[6].

Since the early days of Computer Music research, computers have become both more accessible and more powerful. This has allowed computers to be more frequently used in the composition of music by both amateur and professional musicians. This coupled with the onset of MIDI[2] that coincided samplers and digital synthesisers and samplers - which opened possibilities to hobbyists to have studios in their homes and develop their own sounds. Many of the popular genres we hear today were spawned as a result of this musical revolution.

More recently a significant amount of research has been done into QBE[3] systems as they pertain to music. Most notably Shazam[1], which is a commercial service that aims at music recognition by using audio samples, that may be distorted, in order to query a database of over 3 million tracks.

[1] Council for Scientific and Industrial Research Automatic Computer
[2] Musical Instrument Digital Interface: Developed in the early 1980s[7]
[3] Query-by-Example

Due to this relation between computers and music, along with the commercial use and success of many musical platforms, it is clear that further research into computer music is a worthwhile pursuit. Further to this, the prevalence of using technology to produce music has shown that continued development of platforms that make it easier for artists to make music is important.

In contemporary musical ensembles the drummer usually serves the vital role of providing the tempo, pace, and rhythm of the performance. Due to this the drummer can often define the entire song as all other musicians typically use the drummer as the context in which to frame their timing. Therefore, many genres, can be defined by their use of drums.

When composing or practising a new song it is useful, from a drummers perspective, to know how similar rhythms were used in the wider context of the song. This has been the primary motivation behind developing a QBE system that relies only on the drum rhythms present in a song.

The main use case that has driven design is as follows. A drummer has a drum rhythm in mind, perhaps they have heard it in a song in the past and wants to either see where else it has been used in the past, or perhaps they have designed it themselves and want to see where it has been performed. They play the rhythm on a drumkit as they would usually, which then queries the database and returns all of the similar rhythms.

My goal in developing the system was to build a natural drum rhythm recognition system that:

- Returns both the inferred rhythm and all rhythms that are similar found in the database
- Does not require an extensive digital user interface;
- Extracts the rhythm from repeated play;
- Be trivially extended to work in real time

1.2 Challenges

With any information retrieval system it is important to consider the database available. At the start of this project there was no easily computer-readable database of drum rhythms. Therefore a significant proportion of the project was spent on how to best compile this resource.

Furthermore, as with any imitation-based QBE system it can be difficult to extract relevant features when the user is unable to provide a good example. In this case, each drummer has their own nuances to their performance that manifest in the form of a slower/faster tempo or softer/harder beats. This can make it difficult for users when attempting to reproduce the performance, as the intended patterns have been subject to their own personal transformations.

1.3 Summary of related work

Machine learning approaches

Imitation based querying

Sequence matching

Chapter 2

Preparation

With any project of a non-trivial size it is important to spend a good amount of time carefully planning and organising tasks. In the case of this project it proved particularly useful when it became clear that the implementation of the system had to significantly change, from the one that was originally proposed, when the available resources and preferred system features were taken into account.

This chapter outlines the investigations that were undertaken prior to the projects implementation, along with detailed arguments for design decisions. I begin by discussing the task of building an adequate database, followed by a discussion on the possible querying methods. Finally, I end by discussing the tools that will be used to implement the project.

2.1 Database collection

Clearly, the success of the project heavily relies upon whether an adequate database can be built. Therefore, it is a task that I decided to tackle first so as to provide the base for the future work. Here I will discuss the possible data sources that were considered along with a comparison and argument for the final choice.

2.1.1 Beat tracking techniques

Beat tracking, as defined by Goto[2], is the process of inferring the hierarchical beat structure from musical audio signals. The idea here, depicted in figure 2.1, is that real-time audio from popular music songs can be used as a source of data in order to extract features about the song.

The advantage of this is that there is that the amount of data available is effectively limitless and, if perfect beat tracking were achievable, the rhythm extracted will be a perfect representation of the underlying rhythm. Additionally this approach may allow

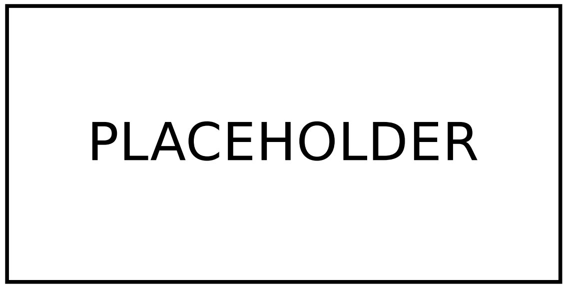

Figure 2.1: Shown here is the aim of beat tracking audio signals.

further features, apart from the beat timings, to contribute to the database - allowing for a richer searching experience.

Perfect beat tracking, however, is a very challenging field. Recent attempts at solving the problem ([3] [4] [5]) have reported efficient algorithms but with only an accuracy of around 60%.

2.1.2 Feature extraction from notation

There are two distinct techniques that can be implemented in order to extract features of drum rhythms from rhythm notation that I will outline here. I will discuss each in turn before returning to the comparison between methods in 2.1.3

Optical Music Recognition

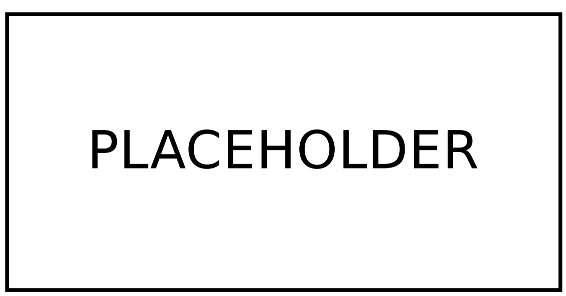

Figure 2.2: Shown here is a standard percussion notation bar

Musical notation has been a practice in many cultures since as long as we have record[10]. As such the notation has been refined over thousands of years, and what we now think of as 'standard notation' has been firmly established since the 18th century[10].

Percussion notation, a specific music notation that pertains to percussion instruments, is much less standardised and only (relatively) recently has there been a push for a standardised practice[11].

There has been a significant research[?][8] into Music OCR[1], which involves using computer vision techniques in order to allow computers to read sheet music. Applying these techniques to percussion notation can enable extraction of feature sets that can be queried.

ASCII Drum Tablature Parsing

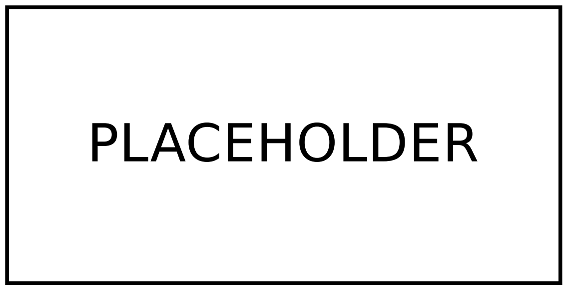

Figure 2.3: Shown here is an ASCII drum tablature bar

In drum tablature, each line corresponds to a single component of the drumkit, as shown in figure 2.3. This contrasts with standard notation, where the height of each note refers to the pitch of the note. Enthusiast transcribers have taken to the ASCII character-encoding scheme in order to share their work over collaborative databases (can I reference http://drumbum.com/drumtabs/ ??).

This ASCII format, while intend to be read by people, may prove to be a great candidate for a general tablature parser. In fact, this problem has been attempted to be solved in a similar case of guitar tablature[13]. However, this solution does not solve it in the general case where the tablature may contain comments to help the human reader. My solution will have to account for this as the collaborative database will not be perfectly void of errors.

2.1.3 Choice of data source

Both beat tracking and music OCR are very challenging fields, each large enough to justify their independent research. ASCII Drum Tablature is very feasible in the time frame and given the large amount of data available in this format it should be feasible to collate an adequate database.

It is important, however, to keep in mind the disadvantages of selecting this data source:

1. As it is a collaborative database, we can not easily verify the accuracy of the transcriptions

[1] Music Optical Character Recognition, sometimes referred to as Optical Music Recognition

2. As there is no standardised format a parser can only be tailored towards a specific practice

2.2 Machine learning vs Information retrieval

This may make more sense in the querying section

2.3 Querying

Now that the datasource has been decided, it is important to discuss how the querying system will work. I will start by outlining methods for inferring the bar from a sequence of beats on the drum. I will then discuss the possible ways of using this sequence to query the database.

2.3.1 Suffix trees

A suffix tree[15] is a string data structure that is useful for many operations on sequences of data. We are mainly interested in it for its ability to find repeated structures in linear time[16].

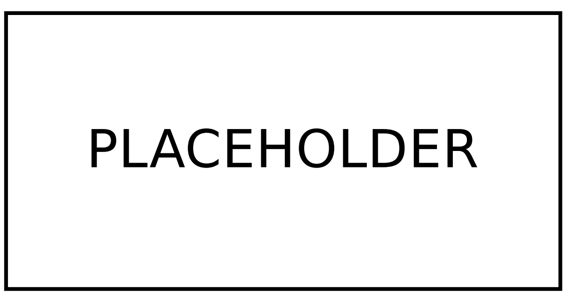

Figure 2.4: Suffix tree of the String "BANANA$"

Ukonnens algorithm (vs others)

reference two papers detailing different ways of making suffix trees. Conclude by saying that Ukonnens allows the system to work on-line

2.3.2 Matching metrics

Describe how matching metrics will be useful

Distance metrics

Talk about how

Matching

RS Bird algo

2.4 Tools Used

2.4.1 Midi

describe midi

2.4.2 Jack

describe jack

2.4.3 Misc

2.5 Summary

This chapter discussed the initial research I did into the relevant areas of Computer Science and Computer Music. This chapter then discusses the advantages and disadvantages of each possible method, ending with a conclusion as to why I decided to make certain design decisions in each section. Finally this chapter discusses the tools used for the implementation of the project

Chapter 3

Implementation

3.1 Overview of system

-parser -midi input -bar extraction -matching

3.2 Parser

As outlined in the (todo: refenence section where I decided to use ASCII drum tablature) the source for the database will in the format of ASCII drum tablature, which is intended to be human-readable, and so parsing it is not a trivial problem.

As shown in figure 3.1 the parser takes an ASCII file containing specifications of the rhythms found in the song and delivers each bar-long rhythm, saved seperately,

-important to remove all erronous sequences

-define the "Ideal" format of Tablature

-7.8mb of parsed data at 135bytes each approx 60000 rhythms

-7.8MB vs 11.9MB

3.2.1 Overview of Parser

3.2.2 Lexical Analysis

In the parser, the lexical analysis involves tokenising the text to

Token all the text with regular expressions

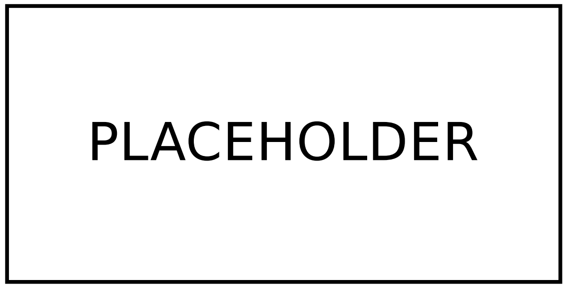

Figure 3.1: Here is an example of how the system will translate and format the drum tablature for pattern matching in the database

Comment removal

Step through some steps of the algorithm in a place wherer comments will be removed

3.2.3 Information lost

Outline cases where we have lost data

3.3 Bar Extraction

We input into the system as a midi file with repeated play and use suffix trees to extract the beat

3.3.1 Suffix Tree's

Define suffix tree's and explain why useful

Ukonnens Algorithm

Explain why we need an online solution from a usability standpoint Overview of algorithm

Supermaximal repeats

Definition How to get them from suffix tree (Gusfield)

3.3.2 Bar inferrence

Use Supermaximal repeat as a reference. Longest Supermaximal repeat is the bar

Quantisation

split bar into 16ths

3.4 Distance metrics

Definition of Distance metrics

3.4.1 2D Edit distance

3.4.2 Hamming Distance

Cyclic extensions

3.5 Summary

Chapter 4

Evaluation

4.1 Comparison with original aims

4.2 Testing

4.2.1 Dataset collection

4.3 Performance Analysis

4.4 Summary

Chapter 5

Conclusion

5.1 Lessons learnt

5.2 Future work

Bibliography

[1] Wang, Avery. *The Shazam music recognition service*, Communications of the ACM, 2006.

[2] Goto, Masataka. *An audio-based real-time beat tracking system for music with or without drum-sounds*, Journal of New Music Research, 2001.

[3] Ellis, Daniel PW. *Beat tracking by dynamic programming*, Journal of New Music Reasearch, 2007.

[4] Ellis, Daniel PW; Poliner, Graham E. *Identifying cover songs' with chroma features and dynamic programming beat tracking* Acoustics, Speech and Signal Processing, 2007.

[5] Davies, Matthew EP; Plumbley, Mark D. *Context-dependent beat tracking of musical audio* Audio, Speech, and Language Processing, 2007.

[6] Mathews, Max. *The Digital Computer as a Musical Instrument*, Science, 1963.

[7] Rothstein, Joseph. *Midi: A comprehensive introduction* AR Editions, 1995.

[8] Bainbridge, David; Bell, Tim. *The challenge of optical music recognition* Computers and Humanities, 2001

[9] Dodge, Charles; Thomas A. Jerse. *Computer music: synthesis, composition and performance*, Macmillan Library Reference, 1997.

[10] Scelta, Gabriella F. *The History and Evolution of the Musical Symbol.*

[11] Weinberg, Norman. *Guidelines for Drumset Notation*, Percussive Notes, 1994.

[12] Johansen, Linn S. *Optical music recognition*, 2009.

[13] Knowles, Evan. *Parsing Guitar Tab*, http://knowles.co.za/parsing-guitar-tab/, 2013.

[14] Doornbusch, Paul. *Computer sound synthesis in 1951: the music of CSIRAC*, Computer Music Journal, 2004.

[15] Weiner, Peter. *Linear Pattern Matching Algorithms*, The Rand Corporation, 1973.

[16] Gusfield, Dan. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology* Cambridge University Press, 1999.