TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY
Julkaisu 865 • Publication 865

Jouni Paulus

Signal Processing Methods for Drum Transcription and Music Structure Analysis

Tampere 2009

Tampereen teknillinen yliopisto. Julkaisu 865
Tampere University of Technology. Publication 865

Jouni Paulus

Signal Processing Methods for Drum Transcription and Music Structure Analysis

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB109, at Tampere University of Technology, on the 8th of January 2010, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2009

Thesis advisor
Anssi Klapuri, Docent
Department of Signal Processing
Tampere University of Technology
Tampere, Finland

Former thesis advisor
Jaakko Astola, Professor
Department of Signal Processing
Tampere University of Technology
Tampere, Finland

Pre-examiner
Meinard Müller, Private docent
Max-Planck-Institut für Informatik
Saarbrücken, Germany

Pre-examiner and opponent
Vesa Välimäki, Professor
Department of Signal Processing and Acoustics
Helsinki University of Technology
Espoo, Finland

Opponent
Gaël Richard, Professor
Signal and Image Processing Department
TELECOM ParisTech
Paris, France

ISBN 978-952-15-2305-2 (printed)
ISBN 978-952-15-2316-8 (PDF)
ISSN 1459-2045

Abstract

THIS thesis proposes signal processing methods for the analysis of musical audio on two time scales: drum transcription on a finer time scale and music structure analysis on the time scale of entire pieces. The former refers to the process of locating drum sounds in the input and recognising the instruments that were used to produce the sounds. The latter refers to the temporal segmentation of a musical piece into parts, such as chorus and verse.

For drum transcription, both low-level acoustic recognition and high-level musicological modelling methods are presented. A baseline acoustic recognition method with a large number of features using Gaussian mixture models for the recognition of drum combinations is presented. Since drums occur in structured patterns, modelling of the sequential dependencies with N-grams is proposed. In addition to the conventional N-grams, periodic N-grams are proposed to model the dependencies between events that occur one pattern length apart. The evaluations show that incorporating musicological modelling improves the performance considerably. As some drums are more probable to occur at certain points in a pattern, this dependency is utilised for producing transcriptions of signals produced with arbitrary sounds, such as beatboxing.

A supervised source separation method using non-negative matrix factorisation is proposed for transcribing mixtures of drum sounds. Despite the simple signal model, a high performance is obtained for signals without other instruments. Most of the drum transcription methods operate only on single-channel inputs, but multichannel signals are available in recording studios. A multichannel extension of the source separation method is proposed, and an increase in performance is observed in evaluations.

Many of the drum transcription methods rely on detecting sound onsets for the segmentation of the signal. Detection errors will then decrease the overall performance of the system. To overcome this problem, a method utilising a network of connected hidden Markov models is proposed to perform the event segmentation and recognition jointly. The system is shown to be able to perform the transcription even from polyphonic music.

The second main topic of this thesis is music structure analysis. Two methods are proposed for this purpose. The first relies on defining a cost function for a description of the repeated parts. The second method defines a fitness function for descriptions covering the entire piece. The abstract cost (and fitness) functions are formulated in terms that can be determined from the input signal algorithmically, and optimisation problems are formulated. In both cases, an algorithm is proposed for solving the optimisation problems. The first method is evaluated on a small data set, and the relevance of the cost function terms is shown. The latter method is evaluated on three large data sets with a total of 831 (557+174+100) songs. This is to date the largest evaluation of a structure analysis method. The evaluations show that the proposed method outperforms a reference system on two of the data sets.

Music structure analysis methods rarely provide musically meaningful names for the parts in the result. A method is proposed to label the parts in descriptions based on a statistical model of the sequential dependencies between musical parts. The method is shown to label the main parts relatively reliably without any additional information. The labelling model is further integrated into the fitness function based structure analysis method.

Preface

THIS work has been carried out at the Department of Signal Processing, Tampere University of Technology, during 2002–2009. First, I wish to express my gratitude to my supervisor Prof. Anssi Klapuri for the close guidance and collaboration throughout this work. He has taught me the whole scientific work process from a literature review to publishing the obtained results, and all the steps in between. I also wish to thank my former thesis supervisor Prof. Jaakko Astola for his assistance in the beginning of this work.

I am very grateful to the pre-examiners of my thesis Priv.-Doz. Dr. Meinard Müller and Prof. Vesa Välimäki for the thorough review of the manuscript, and for Prof. Gaël Richard for agreeing to be the opponent in the public defence of this thesis.

No work could be done without the help of colleagues, therefore I wish to thank the whole Audio Research Group (ARG) for providing the pleasant working environment I have had the privilege to enjoy. Especially, I wish to thank Tuomas Virtanen for enduring me for this whole time, and for all the discussions and help. Many thanks to Matti Ryynänen for helping me to make the computers do what I want them to do, and to Pasi Pertilä for the orthogonal discussions. Thanks to Elina Helander, Marko Helén, Hanna Silén, Toni Mäkinen, and (non-ARG) Jenni Pulkkinen for the coffee room discussions and laughs. Many thanks for David Alves for the collaboration with the multichannel method. And thanks for all the other past and present ARG members, including, but not limited to, Antti Eronen, Lasse Heikkilä, Toni Heittola, Teemu Karjalainen, Konsta Koppinen, Teemu Korhonen, Antti Löytynoja, Annamaria Mesaroş, Tomi Mikkonen, Mikko Parviainen, Tuomo Pirinen, Mikko Roininen, and Sakari Tervo.

I wish to thank Dan Ellis and the other members of LabROSA, Columbia University, for the pleasant visit I had there. Many thanks to Derry FitzGerald in accepting me to co-author a book chapter on unpitched percussion transcription with him: it really helped me to organise the earlier work on the field. I am grateful to Prof. Hannu Eskola for organising Teekkareiden tutkijakoulu, which initiated my post-graduate studies well before graduation.

The funding and financial support of the Graduate School in Electronics, Telecommunications and Automation (GETA), Finnish Foundation for Technology Promotion

(TES), and Emil Aaltosen Säätiö is gratefully acknowledged. This work was partly supported by the Academy of Finland, (application number 129657, Finnish Programme for Centres of Excellence in Research 2006–2011).

Many thanks to Sakke and Hanski for encouraging me to exercise a bit to counterbalance the hours in the office; the indirect contribution to the work is probably greater than I can ever imagine. Additionally, I wish to thank all my friends for all the distraction from the work-related thoughts. Finally, I wish to thank my parents Tarja and Erkki, sister Maiju for all the support and belief in me during my studies. Kiitos.

Jouni Paulus
Tampere, December 2009

Contents

List of Abbreviations

DCT	Discrete cosine transform
DFT	Discrete Fourier transform
DTM	Dynamic texture mixture
DTW	Dynamic time warping
GMM	Gaussian mixture model
HMM	Hidden Markov model
ICA	Independent component analysis
ISA	Independent subspace analysis
k-NN	k-nearest neighbours
LDA	Linear discriminant analysis
MFCC	Mel-frequency cepstral coefficient
MIDI	Musical instrument digital interface
NMF	Non-negative matrix factorisation
NMF-PSA	Non-negative matrix factorisation based prior subspace analysis
PCA	Principal component analysis
PSA	Prior subspace analysis
PSF	Perceptual spectral flux
SDM	Self-distance matrix
SNR	Signal-to-noise ratio
STFT	Short-time Fourier transform
SVM	Support vector machine

List of Included Publications

This thesis is a compound thesis consisting of the following eight publications preceded by an introductory overview of the research area and a summary of the publications. Parts of this thesis have been previously published, and the original publications are reprinted with a permission from the respective copyright holders. The publications are referred in the text with [P1], [P2], and so forth.

P1 **J. K. Paulus and A. P. Klapuri**, "Conventional and periodic N-grams in the transcription of drum sequences," in *Proc. of IEEE International Conference on Multimedia and Expo*, vol. 2, Baltimore, Md., USA, Jul. 2003, pp. 737–740.

P2 **J. Paulus and A. Klapuri**, "Model-based event labeling in the transcription of percussive audio signals," in *Proc. of 6th International Conference on Digital Audio Effects*, London, UK, Sep. 2003, pp. 73–77.

P3 **J. Paulus and T. Virtanen**, "Drum transcription with non-negative spectrogram factorisation," in *Proc. of 13th European Signal Processing Conference*, Antalya, Turkey, Sep. 2005.

P4 **D. Alves, J. Paulus, and J. Fonseca**, "Drum transcription from multichannel recordings with non-negative matrix factorization," in *Proc. of 17th European Signal Processing Conference*, Glasgow, Scotland, UK, Aug. 2009, pp. 894–898.

P5 **J. Paulus and A. Klapuri**, "Drum sound detection in polyphonic music with hidden Markov models," *EURASIP Journal on Audio, Speech, and Music Processing*. In press.

P6 **J. Paulus and A. Klapuri**, "Music structure analysis by finding repeated parts," in *Proc. of 1st ACM Audio and Music Computing Multimedia Workshop*, Santa Barbara, Calif., USA, Oct. 2006, pp. 59–68.

P7 **J. Paulus and A. Klapuri**, "Labelling the structural parts of a music piece with Markov models," in *Computer Music Modeling and Retrieval: Genesis* *of Meaning in Sound and Music - 5th International Symposium, CMMR 2008 Copenhagen, Denmark, May 19-23, 2008, Revised Papers*, ser. Lecture Notes in Computer Science, vol. 5493, S. Ystad, R. Kronland-Martinet, and K. Jensen, Eds. Springer Berlin / Heidelberg, 2009, pp. 166–176.

P8 **J. Paulus and A. Klapuri**, "Music structure analysis using a probabilistic fitness measure and a greedy search algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1159–1170, Aug. 2009.

Author's Contributions to the Publications

The thesis author is the main author in all of the included publications except for [P4]. In the publications [P1], [P2], [P5], [P6], [P7], and [P8] the research, implementations, evaluations, and most of the writing was done by the author. A. Klapuri provided guidance during the research and provided valuable feedback in the writing of the publications. In [P3] the main contribution is by the thesis author, while T. Virtanen assisted with the non-negative matrix factorisation algorithm and participated in the writing of the publication with the author. In publication [P4] the thesis author designed the proposed method, provided a large body of the experimental implementations, and wrote the publication with D. Alves who conducted the reported practical experiments.

List of Supplemental Publications

The following publications by the author are also related to the thesis subjects, but are not included as a part of the thesis.

S1 **D. FitzGerald and J. Paulus**, "Unpitched percussion transcription," in *Signal Processing Methods for Music Transcription*, A. Klapuri and M. Davy, Eds. Springer, 2006, pp. 131–162.

S2 **J. Paulus**, "Acoustic modelling of drum sounds with hidden Markov models for music transcription," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, May 2006, pp. 241–244.

S3 **J. Paulus and A. Klapuri**, "Combining temporal and spectral features in HMM-based drum transcription," in *Proc. of 8th International Conference on Music Information Retrieval*, Vienna, Austria, Sep. 2007, pp. 225–228.

S4 **J. Paulus and A. Klapuri**, "Acoustic features for music piece structure analysis," in *Proc. of 11th International Conference on Digital Audio Effects*, Espoo, Finland, Sep. 2008, pp. 309–312.

S5 **J. Paulus and A. Klapuri**, "Music structure analysis with probabilistically motivated cost function with integrated musicological model," in *Proc. of 9th International Conference on Music Information Retrieval*, Philadelphia, Pa., USA, Sep. 2008, pp. 369–374.

The included publication [P5] is based on the two supplemental publications [S2] and [S3] with new implementations of the proposed methods and new evaluations. The included publication [P8] extends the findings from the supplemental publication [S4], describes the method originally proposed in [S5] in more detail, and presents results from more thorough evaluations. The supplemental publication [S1] acted as a basis for the thesis Chapter 2, but the text was completely rewritten to include more recently proposed methods and focusing the presentation on the matters relevant to the methods proposed in this thesis.

Chapter 1

Introduction

For most people, listening to music is a natural behaviour learnt already as a baby from the nursery songs. People learn to listen to music through constant habituation instead of systematic teaching. Developing algorithmic music content analysis systems can be considered as teaching computers to listen to music. At the current state of music content analysis systems, the computer is barely able to recognise some of the instruments used, analyse some of the underlying rhythmic structures, track some of the most prominent harmonic sounds, and recognise some parts of the song-level structure. Many of these tasks are near-trivial to people without any formal musical training, and still computers struggle to accomplish them. Two specific music content analysis tasks will be discussed in this thesis in more detail: drum transcription and music structure analysis. Before discussing them more, a brief introduction to the wider context of this research is presented.

1.1 Music Content Analysis

In a computer system, an acoustic signal is typically presented as a sequence of *sample values* which correspond to the pressure level variations in the medium (e.g., air) the sound is propagating. Though the digital signal contains all the information of the sound, it must be interpreted somehow. This is usually done by extracting *features* describing some higher-level properties of the signal. Further processing of the features to extract even higher-level information from the signal is called *audio content analysis*. Performing this analysis on everyday soundscapes is referred to as *computational auditory scene analysis* [Ell96]. Automatic music content analysis aims to extract information from musical audio, i.e., aim to teach computers to listen to music. Though music listening can be considered to be a mental process including semantic processing of the acoustic information, a basis for it is the low-level signal processing. Music content analysis can be seen as the opposite of *music synthesis* [Roa96, Väl06] where the aim is to generate the sample values based on a high level description of the content.

The input audio to music content analysis systems is usually *monaural*, i.e., having only one channel, or *stereophonic* with two input channels. If the signal is produced so that it contains only one sound at a time, e.g., a single person singing, it is said to be *monophonic*. However, monophonic music is only a small part of all the music. Most music pieces are *polyphonic*, containing several instruments playing simultaneously. *Sound source separation* aims to process polyphonic signals to recover the monophonic streams [Vir06].

1.1.1 Related Research Topics

Music content analysis often starts by analysing the *spectral* (or *frequency*) content of the signal. In *harmonic sounds* the frequency components are approximately integer multiples of the *fundamental frequency*. When hearing a harmonic sound, the fundamental frequency is often perceived as the *pitch* of the sound. The modelling of pitch perception in the *auditory system* is a wide research topic itself [dC06]. The pitch information can be used for *melody transcription* [Pol07], where the pitch is quantised into *notes* with specific *heights* and *durations*. Melody can be produced by any pitched instrument, and *singing transcription* [Ryy06] is a specific sub-task of it. Music often contains several simultaneous pitches forming, e.g., *chords*. Though pitch estimation has been studied for a long time now, and can be considered to be solved in many cases, *multipitch estimation* [Kla08] is a considerably harder problem. Still, some encouraging results have been obtained in simultaneous transcription of melody, chords, and *bass line* from popular music [Ryy08a].

In addition to the pitched content, another important aspect of music is the timing of the sound events. Locating the start times of sounds is referred to as *onset detection* [Bel05]. The onset times and durations of sounds form the basis for the *rhythm* of a piece. Musical rhythm is often considered to be a hierarchical concept and the analysis of the hierarchy is referred to as *musical meter analysis* [Hai06, Kla06b]. The metrical *pulse* at the lowest level (finest temporal resolution) is called *tatum*, and it coincides most of the sound onsets. The next level, *tactus*, is more common for a music listener as *beat*, corresponding to the rate a listener would tap a foot while listening to a song. The number of beat events in a minute is referred to as *tempo*. The analysis of the tactus level, *beat tracking*, has been studied the most from the hierarchical rhythmic structure [Dix07, McK07]. The next natural level on the hierarchy is *measure* or *bar* length defining a temporal grouping of the lower level events. The drum *patterns* in music often repeat once or twice within a measure.

Various aspects of music content analysis, especially transcription, from the point of view of signal processing are discussed in [Kla06a]. The research on music content analysis methods has gained quite much interest, and different methods for some most popular tasks are evaluated in the annual Music Information Retrieval Evaluation eXchange (MIREX) [Dow08] which is organised in conjunction with the International

Society for Music Information Retrieval Conference (ISMIR)[1]. Some of the popular evaluation tasks in include multipitch estimation, beat tracking, and chord detection. Though the target concepts in these tasks are already quite high-level musically meaningful entities, some tasks aim to even higher, almost semantic, level attempting to recognise, e.g., the musical *genre* [Sca06, Lid07]. Though the tasks differ quite much, it has been demonstrated [Pee08] that one basic *machine learning* method can be applied successfully for *mood* and genre classification, *artist recognition*, and *tagging* the pieces with semantic labels.

1.1.2 Drum Transcription

The first main topic of this thesis is *drum transcription*. The word "drum" is here used to denote the unpitched percussions used in Western music, especially in pop, rock and jazz. Examples of these include bass drum, snare drum, cymbals, and tom-toms, that are often arranged into a *drum kit*. In a more strict definition [Fle98], "drum" refers only to *membranophones* having a membrane (a skin) stretched over an opening of a cavity, e.g., a bass drum or tom-toms. For convenience, the definition of a "drum" is here extended to include also some *idiophones*, which are rigid bodies vibrating as a whole, such as cymbals. The main difference between pitched (transcription of melodies, chords etc.) and drum transcription is that there is no specific pitch information associated with the drums (kettle drums used more often in classical music are an exception of this, but they are not included in the target drum set in this thesis). Additionally, the drums are freely decaying sound sources that do not have an explicit duration, instead only the onset time is important. In a way, drum transcription has much in common with *instrument recognition* [HB06, Ero09].

The drum transcription methods are closely motivated by applications, especially music production related ones. If, e.g., a poor drum set or noisy microphones cause some sound quality problems in the recording, it would be useful to be able to recreate the played sequence with some other sounds. An example of this kind of a *drum replacement* application is DrumTracker by Toontrack[2]. In case the sound of the recording is adequate, but the sequence played needs editing, an *automatic sampling* application, such as FxPansion GURU[3], will be useful. Music listening has traditionally been relatively passive action, and adding some interactivity is an interesting future path. The applications Inter:D [Yos05] and Drumix [Yos07a] allow editing the drum content while listening to the song. Different applications targeted for studios allowing manipulation of existing recordings are a large potential customer base. In a less entertainment oriented context searching for a specific *drum loop* from a large database can be done with an example. It is even possible to produce the query with arbitrary sounds, e.g., by beatboxing (imitating drum sounds by speech-like utterances) [Gil05b]. For

[1] http://www.ismir.net/

[2] http://www.toontrack.com/

[3] http://www.fxpansion.com/

more academic applications, it would be possible to create automatic teaching tools for learning drumming, or to musicological research of the drum track content starting from acoustic data instead of symbolic.

1.1.3 Music Piece Structure Analysis

The second main topic of this thesis is *music structure analysis* [Dan08]. Many popular music pieces have a *sectional form*, meaning that the piece can be subdivided into musical *parts* which may be repeated. The temporal scale of musical parts is considerably longer than the one with musical measure, e.g., the duration of a part may be 20–40 s. Examples of the musical parts on this level include “intro”, “verse”, and “chorus”. In the context of this thesis, music structure analysis refers to the process of finding a segmentation of the piece into occurrences of musical parts, and the possible subsequent grouping of occurrences of the same musical part.

The research on music structure analysis has also been motivated by applications. Music *thumbnailing* means extracting a representative sample of a piece, e.g., in online music stores [Zha07], or for a mobile phone ring tone [Ero09]. The representative samples can be used to identify different versions of the same piece [Góm06]. Continuing the interactive listening experience, the knowledge of the piece structure allows a user to navigate within the piece easily, as demonstrated by SmartMusicKIOSK [Got03] and SemanticHIFI [Vin05] players. Especially in modern highly produced music, the different occurrences of a part may be very similar, and this similarity can be utilised in audio compression [Rao04]. The music structure is not necessarily so prominent in classical music, but still it can be used for synchronising different representations of the same piece. In SyncPlayer [Mül07a] the matched versions can be both audio, or audio and a symbolic, e.g., a score. The subdivision of a piece to musical parts allows also combining musically meaningful segments from different pieces to create a mash-up. Learning some structural properties from the signals and attempting to create music algorithmically has also been proposed [Jeh05]. Further applications include controlling light scenes in synchrony with the music in clubs.

1.2 Objectives of the Thesis

This thesis proposes methods for two music content analysis tasks: automatic transcription of unpitched percussions, and analysis of music piece structure. The methods operate on acoustic input signals, and produce a description of the desired aspect of the content.

The drum transcription part of the thesis proposes several alternative methods for signals ranging from signals containing only drum sounds to full polyphonic music. The set of the target drums has to be set in advance, and in most cases is restricted to bass drum, snare drum, and hi-hat. In other words, the task is to locate the temporal locations where the target drums occur in the input signal. The recognition of individual monophonic drum hits can be considered as a special case of transcription, and will not be discussed as a part of this thesis. In the context of the low-level analysis, developing new acoustic features and classification methods are not in the scope of this thesis. In addition to the bottom-up analysis, simple methods for using repetitive rhythmic structures to provide top-down information to assist in the transcription are discussed.

The second main objective of this thesis is to propose methods for musical piece structure analysis, i.e., providing a description of the sectional form of a piece. The description can consists only of parts that occur more than once during the piece, or it can cover the entire duration of the piece. Though the main objective in music structure analysis is to produce a temporal segmentation of the piece and a grouping of segments representing the same musical part, a secondary objective in this thesis was to provide the groups musically meaningful names, such as “chorus” or “verse”.

1.3 Main Results of the Thesis

The main results and contributions of the publications included in the thesis are listed in the following, organised by the topic.

1.3.1 Drum Transcription

The following publications [P1]–[P5] discuss the problem of drum transcription.

Publication 1

Publication [P1] proposes a method to transcribe polyphonic drum signals, i.e., signals containing only drum sounds. The main contribution of the publication is proposing the use of musicological modelling of the temporal dependencies between drum sound events. In addition to the conventional N-grams producing predictions based on the directly preceding context, the publication proposes the use of periodic N-grams that utilise the information from locations at pattern length intervals. The acoustic modelling unit in the publication is a combination of drum sounds, i.e., multiple simultaneous drum hits. Because of the combination modelling the N-gram alphabet size increases rapidly causing zero-occurrence problems in estimating the probabilities. The publication proposes decomposing the prediction to handle individual drums separately, and producing the combination prediction probabilities by combining these. The proposed method is evaluated with signals synthesised from a commercial MIDI drum track database, and the results show the importance of musicological modelling to aid the acoustic recognition.

Publication 2

Publication [P2] describes a method to produce a transcription from an acoustic signal produced with arbitrary monophonic sounds instead of real drum instruments. In other words, the input may be produced, e.g., with beatboxing or tapping any surrounding objects, but only one sound may be active at a time. The output is a transcription of the events so that each unique sound event is mapped to a drum in a "regular" pop and rock drum kit, based on its rhythmic role. The main contribution of the publication is proposing the concept of rhythmic roles: a model of the probability of different drum sounds to occur at different points within a drum pattern. The proposed method is evaluated on the same database as [P1], but the synthesis was done with various multisampled beatboxing and tapping sounds. The results suggest that method is able to produce meaningful transcription from an input in which the sounds used do not have any direct acoustic correspondence with the drum label.

Publication 3

Publication [P3] proposes a method for transcribing polyphonic drum signals. The method uses spectral templates for the target drums and estimates the time-varying gains for each of them given the input signal. The gain estimation relies on non-negative matrix factorisation. Sound event onsets are then searched from the estimated gains with a psychoacoustically motivated onset detection method. The proposed method is evaluated with an in-house database of drum recordings with varying acoustic properties, and compared with two other methods proposed in the literature. The results suggest that the method works very well when the target signal and the used models match.

Publication 4

Publication [P4] presents an extension of the method from [P3] to multichannel recordings. The motivation for this is that in a studio environment the drum kit is recorded with several microphones, and these signals are available when editing the drum tracks. The proposed method assumes that the different microphone channels are able to produce useful information for the transcription process compared to a single-channel mix-down. In practice, multichannel recordings contain a significant amount of acoustic leakage of all drums to all microphone channels which reduces the performance of more simple methods relying on detecting sound onsets from each microphone channel. The method extends the signal model of [P3] to consider all of the input channels simultaneously. The method is evaluated on a publicly available data set for drum transcription research, and compared with the single-channel method and a naíve multichannel method relying only on onset detection. The results suggest that the use of multiple channels, even with such a simple model, improve the transcription accuracy considerably.

Publication 5

Publication [P5] studies the use of a network of connected hidden Markov models (HMMs) in drum transcription from polyphonic music. The publication is based on two earlier publications [S2] and [S3] that are not included in this thesis. Using HMMs solves the signal onset detection and drum recognition jointly. Two approaches are compared: detector-like modelling with "sound" and "silence" models for each target drum, and modelling of drum combinations. The detector-like modelling was found to perform considerably better than the combination models. Acoustic feature dimensionality reduction with principal component analysis and linear discriminant analysis (LDA) are evaluated, and LDA was found to provide a significant performance increase. Finally, unsupervised acoustic model adaptation with maximum likelihood linear regression is evaluated, but the obtained performance increase was found to be relatively small. The results suggest that drum transcription from polyphonic music is possible using established speech recognition methods.

1.3.2 Music Piece Structure Analysis

The following publications [P6]–[P8] included in this thesis discuss the problem of music piece structure analysis from acoustic input signals.

Publication 6

Publication [P6] proposes a method for analysing music piece structure based on the repeated parts. The main contribution of the publication is defining a cost function for descriptions of musical structure with intuitive terms. The publication proposes a formal definition of the cost function with terms that can be calculated from the acoustic input signal. A deterministic search algorithm is presented to solve the resulting optimisation problem. The method is evaluated on an in-house database of 50 popular music songs, and the problems of evaluating music structure analysis systems are discussed.

Publication 7

Publication [P7] proposes a method for assigning musically meaningful labels to structural description found with some analysis method. The proposed method relies on modelling the sequential dependencies between musical parts, and searches for an assignment that maximises the total N-gram probability evaluated over the description. An algorithm for the optimisation task is proposed in the publication. The method is evaluated on two large data sets of musical piece structure annotations, one being an in-house collection of pieces from various genres, and the other containing the songs by The Beatles. Even though data analysis shows that the stereotypical structures are not so often occurring as was expected, the proposed method is able to assign musically meaningful labels relatively accurately, taken into account the simplicity of the model.

Publication 8

Publication [P8] proposes a music piece structure analysis system relying on a probabilistically motivated fitness measure for the descriptions. The fitness measure is based on the idea that occurrences of a musical part are similar to each other, and differ from those of the other parts. The concept of similarity and difference is formulated in terms of the probability of two musical segments to be occurrences of the same part. The probability values are determined using different acoustic features and different similarity aspects described earlier in [S4] using a mapping function learnt from data. The overall fitness function for structure descriptions is defined with the pairwise probability values. The optimisation of the fitness function given the acoustic information is formulated as a path search in a directed acyclic graph with varying edge costs. A deterministic greedy algorithm is proposed to solve the search problem, and the greediness of the search algorithm can be controlled with two intuitive parameters. The fitness function is also extended to include a labelling term, based on the earlier publication [P7], to provide the description with meaningful labels. The proposed method is evaluated on three data sets (an in-house data set, songs by The Beatles, and a set intended for evaluating music information retrieval methods) using several evaluation measures, and comparing the performance with an existing state-of-the-art method. The evaluation results suggest that the fitness function based approach is a viable choice for music structure analysis.

1.4 Organisation of the Thesis

The main topics of this thesis are drum transcription and music structure analysis, both from acoustic input signal. The rest of this thesis is organised to reflect this division: Chapter 2 provides an overview of drum transcription, starting from introducing the problem, and then going through the relevant methods that have been proposed for solving the problem. Chapter 3 concentrates on music structure analysis, again starting from providing an introduction to the problem, and then discussing different methods proposed to solve it. Finally, Chapter 4 summarises the thesis findings and provides some ideas for future work on these topics.

Chapter 2

Drum Transcription

IN the field of automatic music transcription drums have obtained far less consideration than the pitched instruments, such as the piano. The reason for this may be that drums are often associated with popular (less serious) music and hence have less importance in academic studies. Even though this might be true, one should note that drums, in some form, have been used as musical instruments for as long as there has been any music. This is because, with the exception of singing, there may not be a simpler way to create an instrument and to play it than to take any two solid objects and hit them together. In the following, the properties of drum instruments will be discussed briefly, a definition of the transcription problem will be given, and an overview of the methods proposed to date to solve the drum transcription task will be provided. An earlier review of drum transcription methods has been given by FitzGerald and Paulus [S1].

2.1 General Description of Drums

In this thesis, the word “drum” is used to denote the target instrument of the methods discussed, such as bass drum or hi-hat. In [S1], the term “unpitched percussion” is used instead; “unpitched” is used to distinguish drums from all the other percussion instruments specifically designed to produce a sensation of pitch, e.g., vibraphone, glockenspiel, or xylophone. Drums appear in various forms around the globe. Being able to automatically transcribe every existing drum instrument would be a goal worth reaching, but quite likely impossible to accomplish. This is because of the omnipresence of drums they have obtained numerous physical instantiations of similar basic ideas. The methods that will be discussed in the following have often been designed to handle the drums used in Western pop and rock music, but many of the principles could be applied also to other target drums.

The target drums discussed in the rest of this document are members of either membranophones or idiophones. In [Fle98], where the physical properties of sound production in different instruments are discussed, the word “drum” is used to denote only membranophones, but here it is extended to include also some idiophones. Idiophones are rigid bodies that produce the sound when they vibrate as a whole. Typical examples of idiophones are vibraphone (each of the metallic bars vibrate, producing a certain pitch), wooden blocks, and within a drum kit, the cymbals. Membranophones on the other hand are some sort of hollow bodies that have a membrane or a skin stretched over an opening. The body can be an otherwise closed body (e.g., kettle drum), contain another opening with a membrane (e.g., bass drum), or contain an unenclosed opening (e.g., bongos).

The excitation to drums is usually produced by hitting the instrument. With idiophones, the sound desired determines the hitting region, while with membranophones usually the membrane is hit (other locations are left mainly to produce accentuations). The hitting can be done by bare hands (e.g., congas), sticks (most Western pop and rock drums), mallets (e.g., kettle drums), or brushes (e.g., Western drums in jazz and blues genres). Naturally, the item used to hit the drum affects the resulting sound. A bare hand produces a softer sound than a stick, and the hand itself produces a distinct sound during the impact. The drumsticks are hard and concentrate the impact energy to a very small point on the drum result into a harsher sound, while the brushes are often used to replace the stick when a softer sound is desired: the impact point consists of several flexible bristles softening the hit. The brushes can also be used to sweep the drum to produce a distinct brushing sound. Mallets, having a soft head in many cases, also result in a softer sound than the stick, and can be used to create controlled builds with cymbals (several hits with increasing intensity, adding energy to the vibrating body and stimulating the ringing without a distinct onset). For the physics related to the way the excitation affects the drum and produced sound, the reader is referred to the book by Fletcher and Rossing [Fle98, Part 5].

An example of a Western pop and rock drum kit containing most of the typical drums is illustrated in Fig. 2.1. The main rhythmic sensation is produced by a bass drum (denoted with number 4 in the figure), a snare drum (number 6), and a hi-hat (number 7). The sizes of the drums are often 22 inch diameter for a bass drum, 14 inch diameter for a snare drum, and 14 inches for a hi-hat. The bass drum is hit by a foot-operated mallet and it produces a low-frequency sound, an example waveform and the corresponding spectrogram are illustrated in the top panels of Fig. 2.2. A snare drum is a smaller membranophone, which is very frequently used in popular music. It has two skins, both on the top and in the bottom of the drum, and there is a metallic (originally catgut was used instead [Mai08, p. 491]) snare belt attached across the bottom skin. When the top skin is hit, the bottom one vibrates against the belt and produces the rich, very distinct sound. The lowest two panels in Fig. 2.2 contain examples of a snare drum hit with and without the snare belt. The main difference in the amount of high-frequency energy is visible in the spectrograms as caused by the snare belt. A hi-hat is an idiophone that has two metal plates against each other attached to a connecting arm. The arm is operated by a foot pedal and can be used to press the plates together (closed hi-hat), let them stay apart (open hi-hat), or hit them briefly together (pedal hi-

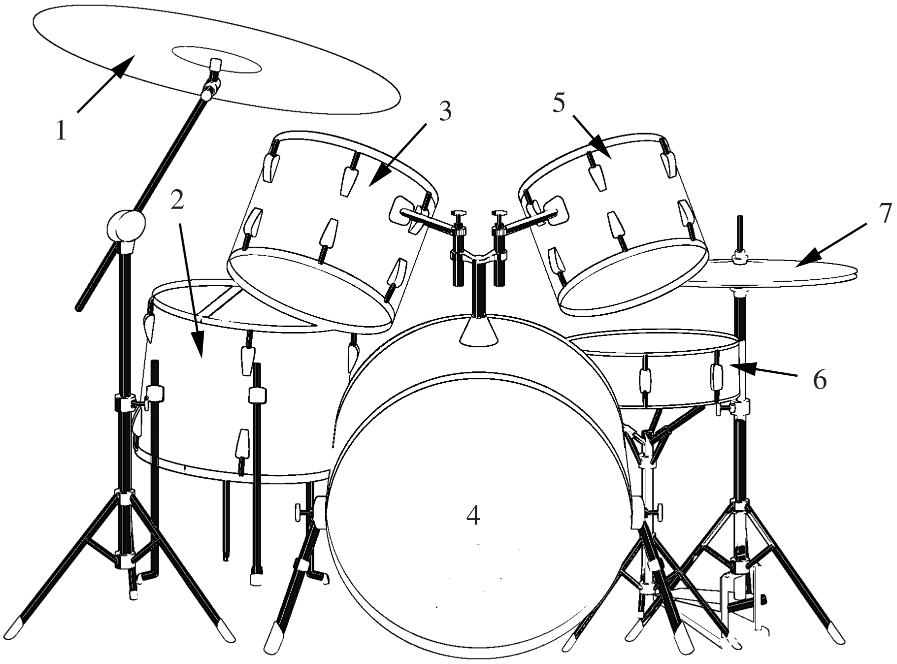


Figure 2.1: A basic drum kit seen from the front. The instruments are from left to right: (1) a cymbal (crash or ride), (2) a floor tom-tom, (3) a tom-tom, (4) a bass drum, (5) second tom-tom, (6) a snare drum, and (7) a hi-hat.

hat). Even though hi-hat is a cymbal, it is often distinguished from the other cymbals in drum transcription method evaluations when defining the target drums because of its special use. In addition to the three above-mentioned drums, a kit often includes one or more tom-toms (a semi-pitched, low-frequency membranophone, numbers 2, 3, and 5 in Fig. 2.1), and various other cymbals (e.g., ride cymbals to bring more texture to the rhythm, and crash cymbals to act as rhythmic accentuations, number 1 in Fig. 2.1.) The panel second from top in Fig. 2.2 illustrates a crash cymbal hit.

Because the three drums: bass drum, snare drum, and hi-hat are most widely used to convey the rhythmic feel in popular music, they are often selected to be the target drums in the transcription systems. Even though it may seem that having only three targets makes the problem trivial, it can be argued that if these three can be reliably transcribed, it enables a large variety of information retrieval applications and still the same principles can be applied on other drums. The importance of the three target drums compared to the other drums in a drum kit is illustrated also by Fig. 2.3 which depicts the relative occurrence frequencies of different drums in ENST drums [Gil06] and RWC Popular music [Got02] data sets. The importance of the three target drums is visible in the graphs.

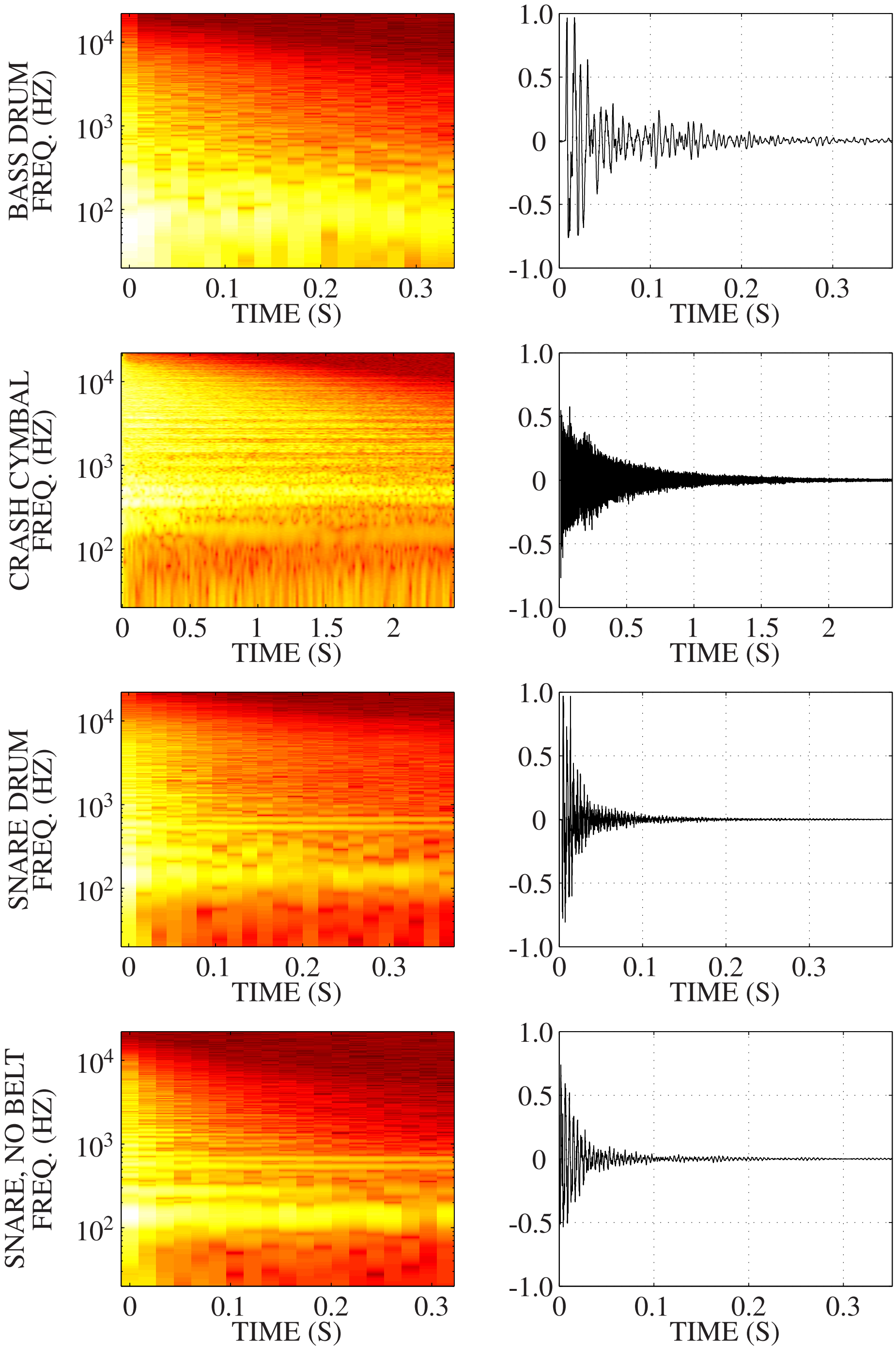


Figure 2.2: Example waveforms and spectrograms of three frequently occurring drums (from top to bottom): bass drum, crash cymbal, snare drum, and snare drum without the snare belt. Note the considerably longer decay time of the crash cymbal compared to the other drums. Brighter colour in the spectrograms indicates higher intensity.

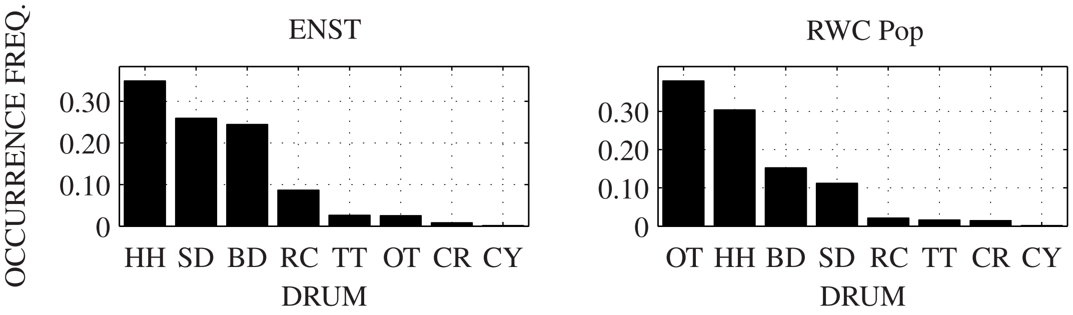


Figure 2.3: Relative occurrence frequencies of different drum hits in ENST drums [Gil06] and RWC Popular music [Got02] data sets. The instruments are denoted by: BD (bass drum), CR (all crash cymbals), CY (other cymbals), HH (open and closed hi-hat), RC (all ride cymbals), SD (snare drum), TT (all tom-toms), and OT (other instruments, e.g., cow bell, triangle, tambourine).

2.2 Problem Definition

In the earlier literature discussing drum transcription the task definition has varied considerably. In the context of this thesis, the task is defined as determining the onset times of drum sounds and recognising which instruments were played. More precisely, the possible drums have been determined in advance, though the set may be different than is actually present in the signal. As the drums are freely decaying sounds after the initial excitation there is no well-defined offset time and thus such will not be estimated. Compared to other audio content analysis tasks, such as auditory scene analysis of our living environment, the sound events to be recognised are very short (in the range of 0.1 s), the number of target classes is limited (in many cases, only the three drums mentioned above), and the events occur in rhythmical patters, often repeating several times in one input signal.

The ultimate goal of drum transcription task is to be able to transcribe any drums from any input signal (with the same system). At the present time this appears still to be far from possible, but many of the proposed methods can be applied on a variety of input signals. The simplest transcription task considers signals where individual hits are concatenated with small intervals to produce a signal where only one drum is playing at a time, but the time instants are unknown. Though this kind of *monophonic drum tracks* are able to deliver the main rhythmic feel of a drum pattern, they are not very realistic inputs to a transcription system. Allowing the sounds to be produced with any arbitrary[1] method, novel practical applications arise. In other words, the input signal could be created by beatboxing (producing drum-like sounds by mouth) [Nak04, Kap04, Gil05b, Haz05, Sin05], or by drumming some basic items on the

[1] The sounds should still be somehow percussive-like, and to be distinguishable from each other.

desktop [P2]. Especially if the sounds are produced by vocals, producing multiple simultaneous sounds is very difficult and the monophonicinity assumption holds. This kind of input might be practical when querying a database of drum loops or songs based on the rhythmic content [Gil05b].

In a problem setup that is more closely related to this thesis the input consists of a *polyphonic drum signal*. In such a signal there are multiple drums played and the sound events and their onsets may overlap temporally. In principle this input could be the recording of a drummer playing without any accompaniment present, e.g., a recording of a drummer's training session from which a transcription is required, or editing a recorded drum track. When the drums are recorded in a studio, each of the membranophones have a close microphone and the cymbals may have some near microphones or they are recorded only with the overhead microphones which capture also some of the ambiance. If these multiple tracks are available and a transcription is wanted, the task becomes slightly different [P4].

From the point of view of many practical applications, it would be desirable that the transcription system could handle normal *polyphonic music* as the input. Indeed, many of the recently published methods are designed to accept polyphonic music as their input, i.e., the input can be any musical piece from radio or a CD. The main challenges with such input are caused by the presence of all the other instruments forming the accompaniment. The accompaniment varies along time, causing segments with similar drum content to differ radically in the actual acoustic signal. In addition, the signal-to-noise ratio (SNR) may become very poor, as seen from Fig. 2.4, or the accompaniment may cause spurious onset detections.

The evaluation of a drum transcription system is relatively straightforward after the set of target drums has been decided upon. For each drum the ground truth annotation and the transcription result are compared, and the events are paired starting from the ones with the smallest deviation between them. Each event may be paired to only one event in the other set and as a result some may remain without a match. Then a temporal threshold is applied accepting only the event pairs that have temporal difference smaller than the given value. As there is no simple rule to decide an appropriate value for the allowed deviation, values in the range from 25 ms [Yos07b] to 50 ms [Gil08] have been used. Pöppel notes in his review article [Pöp97] the following: "If the temporal order of two stimuli has to be indicated, independent of sensory modality, a threshold of 30 ms is observed. Data picked up within 30 ms are treated as co-temporal, that is, a relationship between separate stimuli with respect to the before-after dimension cannot be established." Partly motivated by this comment, the allowed temporal deviation that was used in the included publications [P3] and [P5] was set to 30 ms. The allowed deviation used in publication [P4] is 50 ms to enable comparisons with the results presented in [Gil08].

After the events in the ground truth and transcription have been matched, it is possible to calculate the recall rate

$$R = \frac{\#\text{correct events}}{\#\text{annotated events}}, \tag{2.1}$$

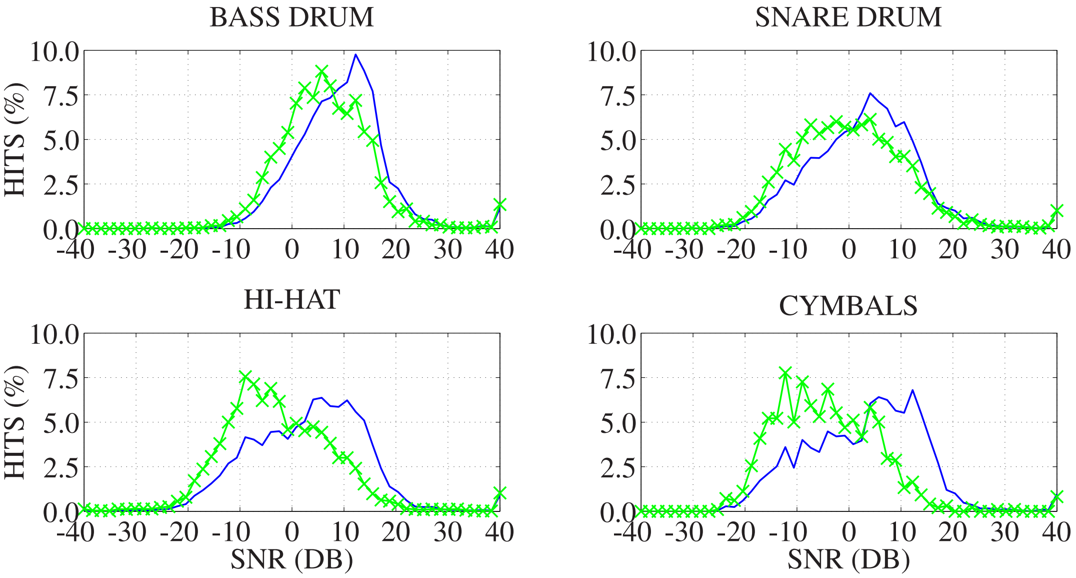


Figure 2.4: Some drum signal to accompaniment ratios (in dB) calculated from the ENST drums data set [Gil06] with "balanced" mixing ratio ($1/3$ accompaniment, $2/3$ drums). The values are calculated using a 92.9 ms frame length with weights from the right half of a Hanning window aligned to the drum onsets. The presented graphs are 50-bin histograms of the ratio of the root-mean-squared energy of the drums-only signal to the accompaniment signal. The histogram counts are normalised to sum to unity. The green line with "×" marker is the normalised histogram counts of segments where the target drum is the only drum having its onset time within a 20 ms window, while the blue line without a marker denotes the normalised histogram of all segments containing the target drum possibly with other drums.

and the precision rate

$$P = \frac{\text{\#correct events}}{\text{\#transcribed events}}, \tag{2.2}$$

as defined by Cleverdon et al. [Cle66]. These two are relatively commonly used quantities in information retrieval method evaluation. For a single composite measure van Rijsbergen [vR79, Chapter 7] proposed the effectiveness measure

$$E = 1 - \frac{1}{\frac{1}{2}\frac{1}{P} + \frac{1}{2}\frac{1}{R}}, \tag{2.3}$$

which later has been transformed into more commonly used F-measure [Jur00, p. 578]

$$F = 1 - E = 2RP/(R+P). \tag{2.4}$$

Though the F-measure provides conveniently a single number for system comparison, also the balance of the precision and recall rates should be observed.

2.3 Approaches

Because the task of drum transcription may initially seem a trivial problem, some methods for solving the problem have not been very systematic in nature. Still, a quite broad range of different approaches have been proposed to solve the problem. FitzGerald and Paulus [S1] divided the approaches into two categories: *segment and classify* and *separate and detect* (these names were proposed by Gillet and Richard [Gil08]). Later, a third category *match and adapt* was proposed in [Gil08]. The methods in the first category segment the input signal temporally and then classify the contents of the segments. The separation methods aim to segregate each target drum as a separate stream from the input signal and then detect the sound event onsets. The third category contains systems which match initial templates (in time, time-frequency, or feature domain) to the input signal, locate potential sounds events, and then aim to adapt the template to fit the input better. A method that cannot be put directly to any of the three categories relies on the use of a network of connected hidden Markov models (HMMs) that perform the segmentation and classification jointly. The main principles of these categorisations are also illustrated in Fig. 2.5. In addition to the methods relying purely on the low-level acoustic recognition, some attempts have been made to utilise musicological modelling in the transcription. The main methods presented in this chapter are gathered in Table 2.1. Before discussing the methods in more detail, some of the frequently used acoustic features are described, and the target classes defined.

2.3.1 Feature Extraction

The purpose of feature extraction is to parametrise the input signal in a way that allows recognising the content, i.e., describe meaningful properties of the signal. The features used in the published methods are often generic acoustic features that have been used also in other tasks, such as speech recognition or instrument classification. The features can be extracted from short, overlapping windows over the segment, or the whole segment can be treated as one feature extraction window. In the former case, the individual feature vectors from the shorter frames can be combined, e.g., by concatenating them over the segment, or by calculating some statistics, such as the mean and variance, over the segment. In case the short frames are used, a simple modelling of the temporal evolution of the features can be obtained by estimating (usually only first and second order) temporal derivatives of the values over the segment and using these delta features as additional features.

Mel-frequency Cepstral Coefficients

A feature set frequently used in audio content analysis tasks consists of mel-frequency cepstral coefficients (MFCCs) [Dav80]. They parametrise the rough shape of the spectral envelope and thus encode some timbral properties of the signal. MFCCs are cal-

Table 2.1: Summary of the discussed methods for drum transcription. The column “Target drums” describes what are the target drum classes (“BD”: bass drum, “CY”: cymbals, “HH”: hi-hat, “SD”: snare drum, “Sh”: shaker, and “TT”: tom-toms), “Input signal” the main acoustic input material the system is intended for, “Approach” the main category of the system, and “Main algorithm” the main algorithm applied in the transcription. “MM” in the algorithm column indicates the use of musicological modelling.

Author / publication	Target drums	Input signal	Approach	Main algorithm
Alves et al. [P4]	BD/SD/HH	close mics	sep & det	multichannel NMF-PSA
Bello et al. [Bel06]	classes: low/mid/high	poly. music	seg & class	clustering
Degroeve et al. [Deg05]	BD/SD/HH/CY/TT	poly. music	classify	feat. selection
Dittmar & Uhle [Dit04]	BD/SD/HH/CY/Sh	poly. music	sep & det	ISA/PSA
FitzGerald et al. [Fit02]	BD/SD/HH	drum loops	sep & det	sub-band ISA
FitzGerald et al. [Fit03c]	BD/SD/HH	drum loops	sep & det	PSA
FitzGerald et al. [Fit03b]	BD/SD/HH	poly. music	sep & det	PSA, harm. suppr.
Gillet & Richard [Gil04]	8 classes	drum loops	seg & class	SVM, HMM, MM
Gillet & Richard [Gil05a]	9 classes	drum loops	seg & class	SVM, modality fusion
Gillet & Richard [Gil07]	BD/SD/HH	poly. music	seg & class	SVM, post-process MM
Gillet & Richard [Gil08]	BD/SD/HH	poly. music	seg & class	SVM, feat. selection, harm. suppr.
Goto [Got01]	BD/SD	poly. music	seg & class	assist tempo estimation
Gouyon & Herrera [Gou01]	BD/SD/HH	drum loops	seg & class	feat. extraction, clustering
Herrera et al. [Her02]	2/5/9 classes	hits	classify	feat. selection, classification
Herrera et al. [Her03]	33 classes	hits	classify	feat. selection, classification
Moreau & Flexer [Mor07]	BD/SD/HH	poly. music	sep & det	NMF, classification
Paulus & Klapuri [P1]	BD/SD/HH	arb. sounds	seg & class	rhythmic role labelling
Paulus & Virtanen [P3]	BD/SD/HH	drum loops	sep & det	NMF-PSA
Paulus & Klapuri [P5]	BD/SD/HH	poly. music	HMM	connected HMMs
Sandvold et al. [San04]	BD/SD/CY	poly. music	seg & class	localised feat. select & models
Schloss [Sch85]	multiple	perc. tracks	seg & class	relative class labels
Sillanpää et al. [Sil00]	7 classes	loops, poly music	seg & class	template matching
Tanghe et al. [Tan05]	BD/SD/HH	poly music	seg & class	SVM
Van Steelant et al. [vS04]	BD/SD	drum loops	seg & class	features, SVM
Virtanen [Vir03]	BD/SD	poly. music	sep & det	sparse coding
Yoshii et al. [Yos06]	BD/SD	poly. music	match & adapt	template matching, MM
Yoshii et al. [Yos07b]	BD/SD/HH	poly. music	match & adapt	template matching & adaptation
Zils et al. [Zil02]	BD/SD	poly. music	match & adapt	analysis by synthesis

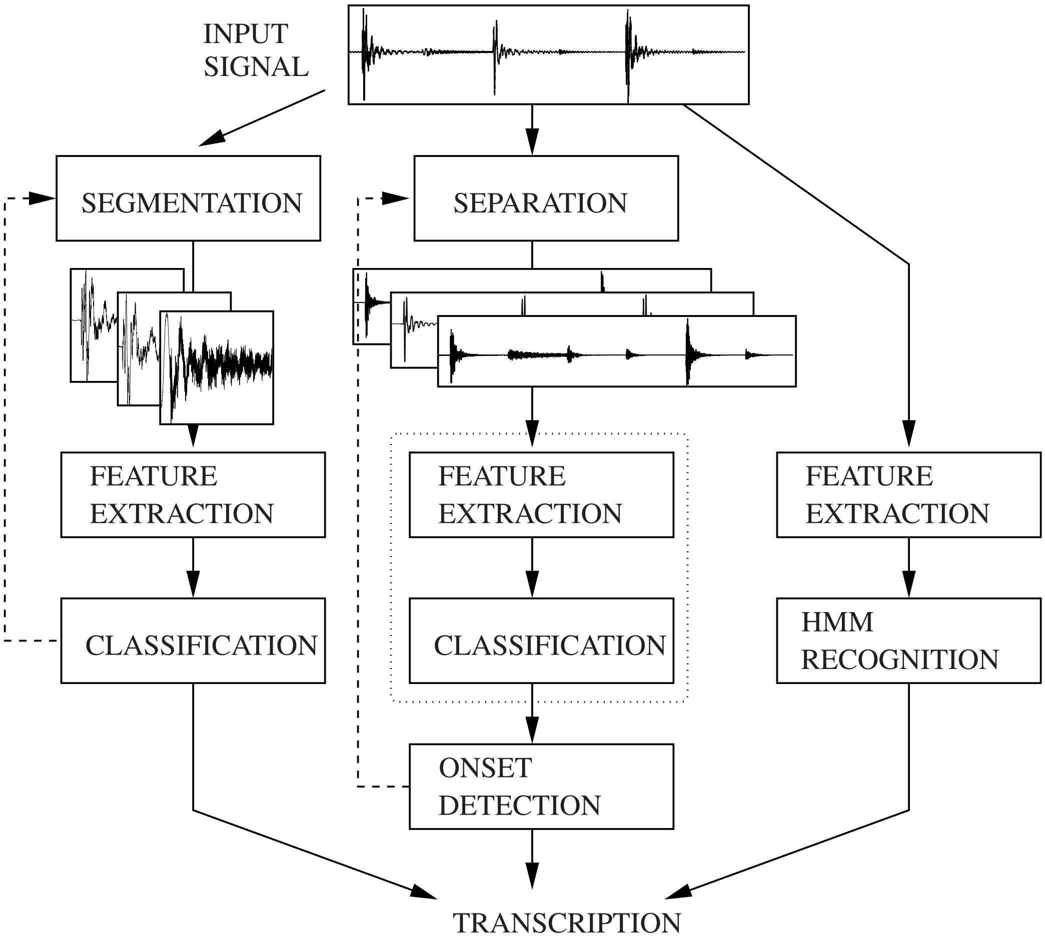


Figure 2.5: Transcription system categories. "Segment and classify" methods (left) segment the input signal to meaningful blocks and recognise the content of each of them. "Separate and detect" methods (middle) segregate each instrument to a separate stream that has to be recognised in the case of unsupervised separation, and finally detect the sound event onsets. The HMM methods (right) determine the segmentation and classification jointly. The dashed arrows denote the adaptation in the "match and adapt" methods, i.e., a feedback loop.

culated through a process illustrated in Fig. 2.6 usually in relatively short (e.g., 20 ms) overlapping frames (in speech recognition applications, a high-pass pre-emphasis is often used). From each frame, the power spectrum is calculated via discrete Fourier transform (DFT). The linear frequency scale is then changed to a logarithmic mel scale better corresponding to human auditory system frequency resolution. Frequency in mel scale F_{MEL} can be calculated from the linear frequency in Hertz F_{HZ} with [O'S87, p. 150]

$$F_{\mathrm{MEL}} = 2595 \log_{10}(1 + F_{\mathrm{HZ}}/700). \tag{2.5}$$

The frequency scale conversion is implemented using a bank of triangular bandpass filters that are uniformly distributed on the mel scale, and calculating the power within

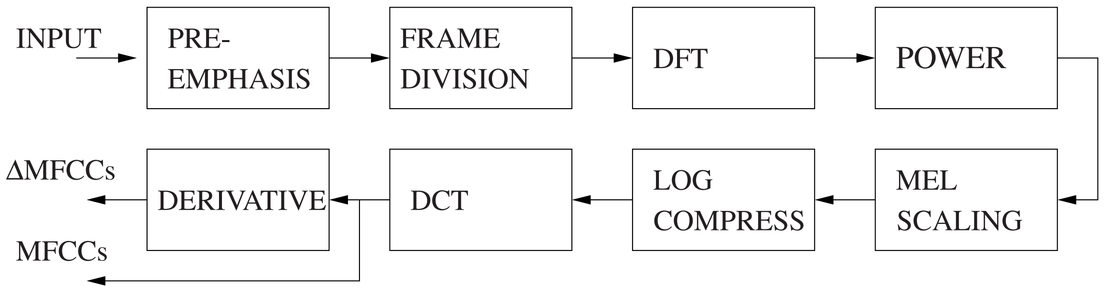


Figure 2.6: Block diagram of MFCC extraction. After an optional high-pass pre-emphasis the input signal is split into short frames, from which their power spectrum is calculated. The energies in triangular response mel-filter subbands are calculated and logarithmically compressed. Finally, the log-powers are transformed into cepstrum by calculating discrete cosine transform from them.

each band. After the mel-scaling, logarithmic compression is applied to the filter bank output to address the non-linear intensity response of human hearing. Finally, discrete cosine transform (DCT) is applied on the compressed filter bank output to produce mel-frequency cepstrum. The DCT is calculated with

$$s_{\mathrm{DCT}}(k) = w_{\mathrm{DCT}}(k) \sum_{n=0}^{N-1} x(n) \cos\left(\frac{\pi(2n+1)k}{2N}\right), \tag{2.6}$$

where $\mathbf{s}_{\mathrm{DCT}}$ is the output from a transform of the length N of input signal $\mathbf{x}$, and $\mathbf{w}_{\mathrm{DCT}}$ is a scaling factor [Gon92, p. 143]

$$w_{\mathrm{DCT}}(k) = \begin{cases} 1/\sqrt{N}, \text{if } k = 0 \\ \sqrt{2/N}, \text{if } 1 \le k \le N-1 \end{cases}. \tag{2.7}$$

Because of the basis vector orthogonality the transform provides some decorrelation of the features. This property is important in the context of using MFCCs with Gaussian mixture models (GMMs) where the decorrelated features allow using diagonal covariance matrices instead of full matrices. Usually only a few (from 5 to 13) lowest coefficients are used as features, and in some applications the zeroth coefficient corresponding to signal energy is discarded. As mentioned above, many applications estimate also the temporal derivatives of MFCCs and these ΔMFCCs are concatenated to the feature vector.

Temporal Pattern Features

The features traditionally used audio content analysis are extracted from short time frames, and calculated from the whole spectrum within the frame. The short duration of the frame is intended to guarantee that the signal remains somewhat stationary within frame and thus could be parametrised with a single spectrum. However, the

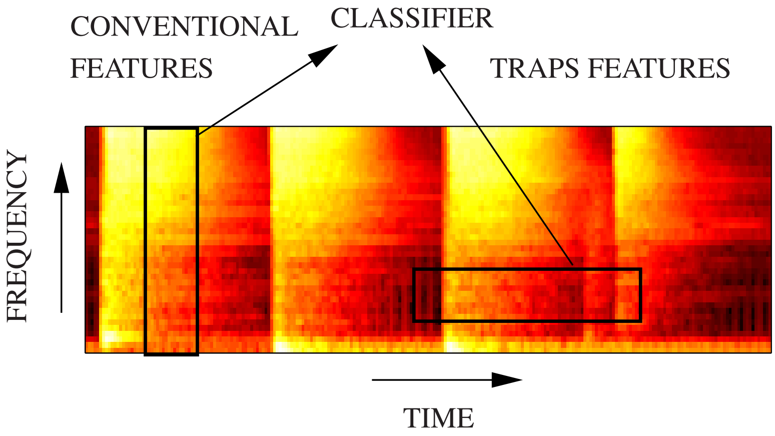


Figure 2.7: The difference between conventional short-time features and temporal TRAPS features illustrated. After [Her98].

importance of the temporal evolution of the feature values has been noted in many classification tasks. In speech recognition it is a standard practise to estimate first and second order temporal derivatives of the features and concatenate these to the feature vector [Rab93]. The HMM architecture itself aims to address the temporal evolution modelling issue by the states in the model: each state has a different feature observation likelihood distribution and the state sequence describes how the feature value may evolve along time.

Hermansky and Sharma [Her98] suggested an orthogonal approach: instead of short-time wideband features narrowband features from long frames should be used, the difference is illustrated in Fig. 2.7. They showed that the TRAPS (TempoRAl PatternS) feature describing the evolution of the energy envelope of narrow subbands offers complementary information compared to the MFCCs and improve the recognition accuracy with speech slightly. The TRAPS idea was used by Paulus and Klapuri [S3] to improve the recognition of a HMM transcription system. They calculate energy envelopes within 1/3-octave bands and 100 ms frames. The shape of the envelope is parametrised in a shift-invariant form with the DFT spectrum and the representation is compacted further with DCT and retaining only few lowest coefficients. The coefficients from all bands are concatenated into a single feature vector and a detector-type GMM classifier for each target drum is trained. In the transcription phase, the probability produced by the GMM that a target drum is present in the long frame is summed to the observation likelihoods of drum combination HMMs (see Sec. 2.3.6) before decoding. The evaluation results suggest that TRAPS is able to provide some complementary information compared to plain HMM system. Furthermore, it was noted that even though the performance did not increase dramatically, the main error type was changed from insertion to more balanced with insertion and deletion.

Other Features

In addition to MFCCs, a large body of other instantaneous features have been used to parametrise the segment content. Majority of these aim to describe the shape of the spectrum in some way, usually with a single scalar. From the vector-form features, band energy ratio may be the most often used. It divides the input spectrum into subbands (usually from 5 up to 40), calculates the energy in each of these, and then normalises by the total signal energy. The number and spacing of the bands depend on the desired frequency resolution, but often the spacing is close to logarithmic to mimic the human auditory system.

Many of the features used in drum transcription are scalar valued describing the shape of the spectrum. For convenience, the normalised magnitude spectrum is denoted with

$$\tilde{s}(k) = \frac{|s(k)|}{\sum_{k=1}^{K} |s(k)|}, \tag{2.8}$$

where $s(k)$ is the DFT spectrum on frequency index $k \in [1, K]$. The spectral features used in drum transcription include the following among others.

- *Spectral centroid*, the centre of mass of the magnitude spectrum, defined as

$$C_{\mathrm{F}} = \sum_{k=1}^{K} k\tilde{s}(k). \tag{2.9}$$

- *Spectral spread*, describing the bandwidth of the signal, defined as

$$S_{\mathrm{F}}^2 = \sum_{k=1}^{K} (k - C_{\mathrm{F}})^2 \tilde{s}(k). \tag{2.10}$$

- *Spectral skewness*, measuring the asymmetry of the spectrum, defined as

$$\gamma_3 = \frac{\sum_{k=1}^{K} (k - C_{\mathrm{F}})^3 \tilde{s}(k)}{S_{\mathrm{F}}^3}. \tag{2.11}$$

- *Spectral kurtosis*, measuring the peakiness of the spectrum, defined as

$$\gamma_4 = \frac{\sum_{k=1}^{K} (k - C_{\mathrm{F}})^4 \tilde{s}(k)}{S_{\mathrm{F}}^4}. \tag{2.12}$$

- *Spectral roll-off*, describing a frequency below which a certain amount r (e.g., 85%) of the energy resides in, defined as

$$k_{\mathrm{ROP}} = \operatorname*{argmax}_{\hat{\mathrm{K}}} \left\{ \sum_{k=1}^{\hat{K}} \tilde{s}(k)^2 \leq r \sum_{k=1}^{K} \tilde{s}(k)^2 \right\}. \tag{2.13}$$

These spectral features can be calculated directly in linear frequency scale resulting from the DFT or after logarithmic scale resampling, as suggested in MPEG-7 standard [Int02]. In his pioneering drum transcription work Schloss [Sch85] calculated the sound decay time from the sound's energy envelope by fitting an exponential decay curve to it, and classified the event to damped or undamped based on the decay time. For a more complete description of different features and their suitability for the drum transcription refer to the publications by Herrera et al. [Her02], in the context of more general audio classification the publication of Peeters [Pee04b], or the feature selection with simulated annealing optimisation the publication of Degroeve et al. [Deg05].

2.3.2 Target Classes

The target class definition is a necessary step in designing a drum transcription method. To date, excluding the publications aiming to recognise individual hits, the set of target drums has varied from containing only bass and snare drums [Got94a, Got01, Zil02, Vir03, vS04, Yos06], the two and hi-hat or other cymbals [Tar04, Yos07b, Gil08], [P3], [P5], up to practically full drum set with tom-toms and other percussion instruments [P1] and [Gil04]. As earlier illustrated by the occurrence frequencies in Fig. 2.3, very few target drums are sufficient. Thus, in the recent publications, the target drum set has settled into consisting of only three drums: bass drum, snare drum, and hi-hat.

In defining the target classes after determining the target drum set two approaches have been adopted: drum combination recognition and detectors for individual drums. If the target drum set consists only of bass and snare drums, the combination recognition approaches aim to detect the classes "silence", "bass", "snare", and "bass + snare", while detectors make a binary decision for both drums independently "bass"/"no bass" and "snare"/"no snare". In this example, the number of resulting recognition classes is equal, but when the number of target drums increases, the number of combinations will increase more rapidly than with the detectors (2^N vs. $2N$, when the number of target drums is N).

Not only does the number of classes increase rapidly with combination modelling, but the combinations occur at very different rates in music. This is illustrated in Fig. 2.8, where the occurrence frequencies of the combinations formed by five drum classes in two data sets have been calculated. Though the order of the combinations differ slightly, the trend is clearly visible: the five most frequently occurring combinations cover approximately 80% of all drum combination events (and contain the same combinations in both data sets), and the ten most frequent combinations cover already approximately 95% of the occurrences in both data sets.

In practical considerations, the use of detectors instead of combinations recognition makes a better use of the (usually very limited amount of) training data. As already illustrated in Fig. 2.8, majority of the combinations occur quite rarely even in large data sets. Because of this, the training of the acoustic models will have only a few examples of the combination sound events, and the resulting models may become inaccurate or they will overfit the examples hindering the generalisation capability. On the other

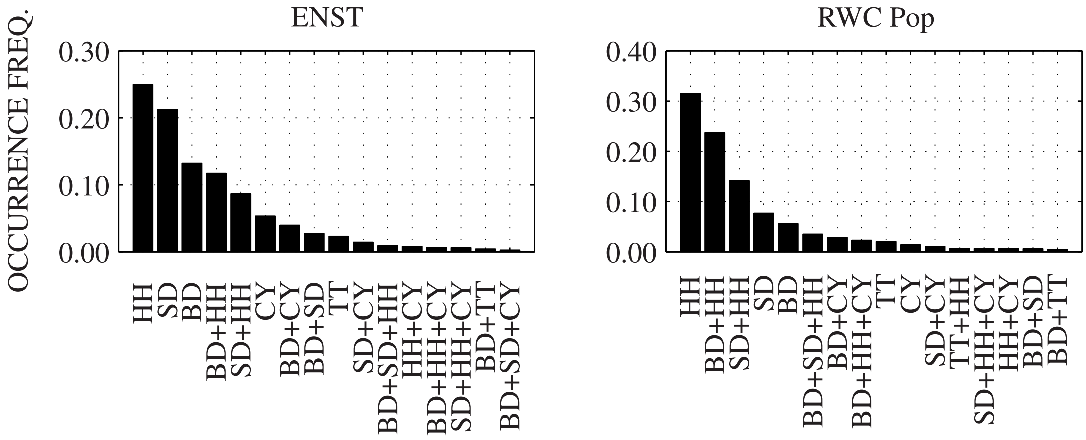


Figure 2.8: Relative occurrence frequencies of various drum combinations in ENST drums [Gil06] and RWC Popular music [Got02] data sets. Different drums are denoted with HH (all hi-hats), BD (bass drum), SD (snare drum), TT (all tom-toms), and CY (all cymbals). Two drum hits were determined to be simultaneous if their annotated onset times differ less than 10 ms. Only the 16 most frequent combinations are shown from each data set.

hand, if detector modelling is done, occurring combinations will serve as a training examples for all the drums present in the combinations, thus making effective use of the data. As a result, most of the more recently published methods rely on using drumwise detectors instead of using drum combinations as the target classes.

2.3.3 Segment and Classify Approach

As the definition of the drum transcription problem stated, the objective is to know when drums were played and which were the drums played. The methods in the "segment and classify" category answer these questions in this same order by operating with the following steps:

1. Divide the input signal into meaningful segments either by

    (a) locating sound event onsets, or by

    (b) creating a temporal grid over the signal, e.g., with musical meter analysis.

2. Extract a set of features from each segment.

3. Classify the contents of the segments based on the feature values.

These steps will now be discussed in more detail.

Temporal Segmentation

Temporal segmentation can be performed either by detecting prominent sound events from the signal or by creating a (constant) temporal grid over the signal. Sound onset detection is a difficult problem in itself, and will not be discussed here in detail. Briefly, most of the onset detection methods operate by transforming the acoustic input signal to a detection function which has peaks in the locations of sound event onsets. This detection function may be based on, e.g., changes in energy envelope in signal subbands or large jumps in the phase information. For a tutorial on the subject, refer to the work of Bello et al. [Bel05].

An alternative to onset detection is to create a temporal grid over the signal so that most of the onsets will coincide with a grid point. In practice this requires applying some sort of a musical meter analysis to determine an appropriate temporal resolution. The temporal level that is most suitable for the task is the level of tatum (or tick [Gou02], or attack-point [Sch85]) as it has been defined by Bilmes [Bil93] as "the regular time division that most highly coincides with all note onsets". In the metrical hierarchy, tatum is a subdivision of tactus (or beat, the foot-tapping rate). For more detailed description about meter analysis, refer, e.g., to the thesis of Gouyon [Gou05] or the book chapter by Hainsworth [Hai06].

Both of the segmentation approaches have some drawbacks and benefits. The temporal grid segmentation presumably works well with music that has a constant tempo and the played drum patterns do not deviate considerably from it. However, if the grid parameters, i.e., tatum length and phase, are estimated incorrectly, the segmentation will be incorrect which may affect the subsequent recognition step. In addition, even though the tatum grid was correct, the expressivity of the drummer's playing may shift the sound events to non-grid point locations and the net effect is the same as with the incorrect grid estimate. On the other hand, while the onset detection is not constrained by assumptions on the temporal structure of the signal and can capture the expressive temporal deviations, there is a risk that some drum onsets will be missed and lost completely. To date, most of the "segment and classify" methods utilise onset detection as the segmentation method, and reduce the risk of missing drum events by lowering the detection threshold (which in turn causes spurious detections that have to be addressed).

The actual segmentation can be done by simply taking the part of the signal starting at a grid point or a located onset and ending at the next grid point or onset [Gou01, Gil04]. To reduce the variation of the segment lengths, in practice it is beneficial to defined the minimum and maximum lengths of the segments.

Segment Content Recognition

Once the feature extraction has been done and the target classes defined, the rest of the transcription problem is a standard classification task. The different classification methods will not be discussed in detail here, but some basic principles will be mentioned instead, an interested reader is referred, e.g., to [Dud01]. A coarse categorisation of supervised classification algorithms employed in drum transcription is

- rule-based methods,
- non-parametric instance-based methods,
- linear discriminant methods, and
- statistical modelling methods.

There has been very little work on systematic comparison of different classification methods in the drum transcription context, with the exception of Herrera et al. [Her02]. Even in the paper above, the comparison did not address recognising drum hits is a musical context, but classifying individual hits instead. However, the obtained results suggest that instance-based methods would be most suitable for the task.

Rule-based Classification Rule-based classification methods, such as decision tree classifiers, aim to construct a set of rules that allow deciding the correct class. In decision trees, the rules can be organised into a branching tree, where each node corresponds to a single decision or a rule. In other words, each branching decision depends on the value of a feature or condition, and the final classification result is obtained by following the decision until a leaf node is reached. For more details on decision trees, refer to [Qui93, Dud01]. To date, only Sandvold et al. [San04] have utilised decision tree classifiers in drum transcription.

Instance-based Classification (Non-parametric Classification) Considering the supervised classification task, instance-based methods are perhaps the most intuitive to understand. In the training phase, the system is given a set of examples and is told the class of each example, and it stores all of them or only a representative subset. Then in the classification phase, the input is compared to the stored instances and the class is decided based on the closest stored example(s). The k-nearest neighbour (k-NN) classifier operates with these steps storing all the training examples, and in the recognition phase selecting the *k* nearest training instances and determining the classification result with a majority vote from them.

The main shortcomings of instance-based classification methods are the amount of space consumed by storing the training examples and the computational load from comparing each stored training sample to the input to be classified. These can be solved, e.g., by clustering the samples and storing only the cluster centroids.

An instance-based classifier was employed in what is presumably the first method to handle polyphonic drum mixtures presented by Goto and Muraoka [Got93, Got94b]. It utilises a power spectrogram based template for each of the nine target drums and matches the templates to the locations of found onsets. Thus, in a way, this is an instance-based classification method with only a single template stored for each target drum. The same matching method has been later extended by Yoshii et al. [Yos07b] to target a smaller drum set, and to include some signal-dependent adaptation. Goto used similar idea in [Got01], but without the pre-defined templates and matching only vectors instead of matrices to determine possible locations of bass and snare drums hits. The final application, though, was beat analysis in music with drums, so actual transcription was not produced. Another example of a method with spectrogram template matching with added top-down processing was proposed by Sillanpää et al. [Sil00].

Linear Discriminant Classification In a two-class case, linear discriminant function aims to define such weights $\boldsymbol{w}$ and a bias term b that the output y_i of

$$y_i = \boldsymbol{x}_i^{\mathsf{T}} \boldsymbol{w} + b \tag{2.14}$$

is positive for one class and negative for the other ($\boldsymbol{x}_i$ is the i^{th} input vector), and that the resulting function contains maximal discriminative power. The discriminative power can be measured, e.g., with in-class scatter versus inter-class scatter as is done in Fisher discriminant analysis (also known as linear discriminant analysis (LDA)). Linear discriminant functions have not been used in drum transcription as such, but instead as a feature pre-processing before other classifiers.

Support vector machines (SVMs) can be considered as an extension of linear discriminant classifiers. They aim to define such parameters $\boldsymbol{w}$ and b that the margin between the formed decision surface and the training examples is maximised [Bur98]. The classes may not be separable by a linear decision surface in the feature space. In such cases, a kernel function $u(\cdot)$ may be used to map the features to a higher dimensionality space where the classes can be separated. In the training phase, the SVM parameters are optimised to fulfil the requirement

$$y_i = \begin{cases} u(\boldsymbol{x}_i)^{\mathsf{T}} \boldsymbol{w} + b \geq +1, \text{if } c_i = +1 \\ u(\boldsymbol{x}_i)^{\mathsf{T}} \boldsymbol{w} + b \leq -1, \text{if } c_i = -1 \end{cases}, \tag{2.15}$$

where c_i is the class associated with the training feature vector $\boldsymbol{x}_i$. The training samples that lay on the margin, i.e., the samples $\boldsymbol{x}_i$ that have the result $y_i = \{-1, +1\}$ are the only ones affecting the final surface and bear the name support vector. In the classification, only the sign of the result is important and the class for feature vector $\boldsymbol{x}_i$ is determined by

$$c_i = \text{sign}(u(\boldsymbol{x}_i)^{\mathsf{T}} \boldsymbol{w} + b). \tag{2.16}$$

In its basic form SVM is a binary classifier and as such suits to operate in a detector-like recognition setup. Some multiclass extensions have been developed (see, e.g., [Tso04]), combining several classifiers in 1-against-1 or 1-against-others setups. Gillet and Richard [Gil04] compared the performance of several binary SVMs and one multiclass SVM targeted to recognise drum combinations, but the performance difference was so small that no conclusions could be made.

SVMs have shown to perform well in the classification tasks in drum transcription by van Steelant et al. [vS04], and Gillet and Richard [Gil04, Gil05b, Gil08]. In [P1] SVMs are used as a part of a recognition system performing the temporal segmentation with a constant grid; an SVM is used to determine if a grid point contains a drum hit to be further recognised, or does it contain only some non-drum sound.

Statistical Modelling Methods Statistical methods aim to construct a model of the class to be recognised. Within the field of drum transcription, practically the only statistical modelling approach for the low-level acoustic features has been Gaussian mixture models (GMMs). GMMs estimate the probability density function of the data $\boldsymbol{x}_i$ from a class as a weighted sum of J normal distributions

$$p(\boldsymbol{x}_i|\{w_j,\mu_j,\Sigma_j\}_{j=1,\ldots,J}) = \sum_{j=1}^{J} w_j \mathcal{N}(\boldsymbol{x}_i;\mu_j,\Sigma_j), \tag{2.17}$$

where the weights are bound by $\sum_{j=1}^{J} w_j = 1$, and $\mathcal{N}(\boldsymbol{x}_i;\mu_j,\Sigma_j)$ is a multivariate normal distribution with mean μ_j and covariance matrix Σ_j evaluated at point $\boldsymbol{x}_i$. Plain GMMs have been used in drum transcription by Paulus and Klapuri [P1] where a GMM was trained for each of the 127 non-empty combinations of seven drums. Due to the large amount of classes, the recognition result was not very good. Gillet and Richard [Gil04] used GMMs to recognise drum combinations and to act as detectors for individual drums, and the recognition result was then combined with a musicological model for the final transcription result in a hidden Markov model -like system.

More often GMMs are used within hidden Markov models (HMMs) to model the (acoustic) observation likelihoods. The HMMs employed in the discussed methods consist of two processes, a discrete finite state machine and a GMM producing observations from the states. The parameter set of an HMM is denoted by

$$\lambda = \{\boldsymbol{A}, B, \pi\}, \tag{2.18}$$

where $\boldsymbol{A}$ defines the state transition probabilities of the state machine, B parametrises the observation likelihoods in the states, and π contains initial (and final) state occupancy probabilities. If the state at time t is denoted with q_t, then the values of the state transition matrix $\boldsymbol{A}$ can be defined as

$$A(i,j) = p(q_{t+1} = j|q_t = i). \tag{2.19}$$

Similarly the observation likelihood parameters in B define the probability of observing $\boldsymbol{o}_t$ given the HMM state and usually is evaluated with the GMM model (2.17). Recognising with HMMs, the problem is to find a state sequence $\boldsymbol{q} = q_1, q_2, \ldots, q_t, \ldots, q_T$ given the parameters λ and the observations (feature vectors) $\boldsymbol{O} = \boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_t, \ldots, \boldsymbol{o}_T$ that is "optimal" in some sense, usually to find the single best path $\hat{\boldsymbol{q}}$ maximising the total observation likelihood

$$\hat{\boldsymbol{q}} = \underset{\boldsymbol{q}}{\operatorname{argmax}} \{p(\boldsymbol{q}|\boldsymbol{O},\lambda)\}. \tag{2.20}$$

This can be solved, e.g., with the Viterbi algorithm. Further details of HMMs and the training of the model parameters are not discussed here, an interested reader is referred to the tutorial by Rabiner [Rab89].

In the drum transcription task, HMMs have been used in two ways: as a musicological model [P1] and [Gil04], and in continuous recognition that will be discussed in more detail in Sec. 2.3.6. The musicological modelling in [Gil04] used the basic "segment and classify" approach with GMMs for the low-level recognition of different target classes in the found onset locations. These recognition results then acted as the observation likelihoods in the HMMs architecture and the state sequence modelled how different drums or drum combinations follow each others. However, more simple SVM classifier without any sequence modelling was found to outperform the HMM approach [Gil04].

The results obtained with the "segment and classify" methods are quite encouraging. However, even though it is trivial for a human to recognise that different makes and models of the same drum still represent the same class, the acoustic features may vary considerably within the classes. This may cause some problems in the generalisation in the model training. As a related research topic, Pampalk et al. [Pam08] compared two models for measuring the similarity of drum sound events and how the output of the models correlate with human perception. Though the obtained results were aimed for sample retrieval from databases, similar research should perhaps be taken into account in the transcription task, too.

The input with beatboxing can still be considered to be similar to the transcription tasks described above as the input sounds share likeness even across people and a supervised classifier can be trained. Such beatboxing recognition systems with "segment and classify" approach have been proposed, e.g., by Nakano et al. [Nak04], Kapur et al. [Kap04], Gillet and Richard [Gil05b], Sinyor et al. [Sin05], and Hazan [Haz05], with the usual application of querying a database of real drum loops with speech-like input.

Proposed Methods

From the publications included in this thesis [P1] and [P2] can be considered to belong to the "segment and classify" category. The former uses a tatum grid for segmentation, an SVM to operate as a "silence" / "drum" classifier in each grid point, and a GMM to recognise the content of the grid points deemed to contain drums. The latter uses unsupervised classification, i.e., clustering, to group the events segmented with an onset detector. After the grouping, the classes were assigned with rhythmic role labels, see Sec. 2.4.2.

2.3.4 Match and Adapt Approach

The "match and adapt" methods may also be considered to belong to either of the "segment and classify" or "separate and detect" categories, depending on their basic approach. However, they are here treated as a separate category to emphasise the importance of the adaptation idea. The main motivation to the use of adaptation in recognition is that in music signal, the actual target instrument sounds vary considerably between signals, and especially in polyphonic music the background "noise" is different in every signal. These result in the situation where pre-trained generic models do not necessarily match the properties of the actual hits and the recognition performance suffers.

The "match and adapt" philosophy is exhibited in the purest form in the method by Zils et al. [Zil02], in which bass and snare drums are transcribed in a time-domain "analysis by synthesis" approach. As the initial templates the method uses low-pass and band-pass filter impulse responses for bass and snare drum, respectively. Then the possible onsets of the target drum are searched by calculating correlation between the input signal and the template. The reliability of the found onsets is assessed, and the template is adjusted to be the mean of the matched template and the weighted average of the found events, and this process is repeated until the set of found onsets has converged. Even though the approach may seem crude, it was reported to locate the two drums "perfectly" in approximately half of the test cases.

Another method with multiple adaptation steps was proposed by Yoshii et al. [Yos07b]. The method locates possible drum sound onset locations from the input signal (each target drum is handled separately), matches a fixed-length spectrogram extracted after the onset location to a pre-calculated template, and estimates the reliability that the target drum occurs in the segment. Then the segments with the highest reliability to contain the target drum are used to adapt the template, and the process is repeated until convergence. In the evaluations the template adaptation showed to improve the recognition result especially with bass drum, whereas the improvement was less significant with snare drum and hi-hat.

In addition to these iterative template matching methods, two publications have proposed to use regular classifiers, but after initial recognition with generic models retraining the models based on the recognition result. Sandvold et al. [San04] aim to recognise the classes "kick", "snare", "cymbal", "kick+cymbal", "snare+cymbal", and "not-percussion" with localised models. The general models are C4.5 [Qui93] decision trees utilising an automatically determined subset of a large (115 descriptors) feature set. After the initial recognition few most reliably recognised sound events are selected and the feature selection and classifier training was repeated with them. The final transcription is then obtained using these *localised models*. The evaluations showed a considerable reduction in the required feature dimensionality, and a notable increase in the recognition accuracy. However, the reliability estimation phase was done by picking the re-training instances by hand instead of a fully automatic system. Gillet and Richard [Gil05c] had less success with their attempt with localised models. They used an SVM classifier, mapped the SVM output to a classification result probability (see Sec. 2.4.1), selected the most reliable hits, and retrained the classifiers with them. The authors reported that the use of localised models did not improve the overall recognition result. When they analysed the reason for this, the re-training instance subset was revealed to consist of loud or solo hits that were not in the normal rhythmic pattern context. Thus the localised models were more prone to recognise such off-pattern hits instead of improving the result on the basic repeating pattern.

2.3.5 Separate and Detect Approach

The "segment and classify" methods answer the transcription problem in the following order: 1. when did a sound event occur, and 2. what drums were present in the event. In general, the "separate and detect" methods reverse the order of the questions and aim to to first segregate each target drum as a separate stream and then detect the sound event onsets from the stream. The "separate and detect" methods work roughly with the following main steps:

1. Transform the input signal to an appropriate mid-level representation, e.g., a magnitude spectrogram.

2. Separate each target drum to an individual stream.

3. Only with unsupervised separation methods: extract features and recognise each stream.

4. Detect sound event onsets from the streams.

The main differences between the separation based methods stem from the employed source separation technique. A broad categorisation divides the approaches to *unsupervised* and *supervised* source separation methods. In the supervised separation there exists a specific model for each target drum that will be utilised in the separation process, while the unsupervised methods attempt to obtain the result purely based on the input signal and some assumptions of the sources.

Considering source separation, the method best known is probably independent component analysis (ICA) [Com94]. ICA is intended to estimate N sources from at least as many input signals by maximising the statistical independence between the resulting source signals. However, drum transcription is usually done from a single-channel or stereo input and the time-domain ICA is not applicable as such.

The restriction imposed by the single-channel input is alleviated when a different mid-level representation is used instead of the "raw" time-domain signals. The idea of independent subspace analysis (ISA) originally proposed by Hyvärinen and Hoyer [Hyv00] was taken into practice by Casey and Westner [Cas00]. Multiple sensor signals are constructed from a single input signal by transforming it to a time-frequency representation, i.e., to a spectrogram, and by treating the different frequency channels as sensor signals.

Signal Models

With the spectrogram representation, the input signal spectrogram $\boldsymbol{X}$ can be modelled to be a sum of N individual source (or component) magnitude spectrograms

$$\boldsymbol{X} \approx \sum_{i=1}^{N} \boldsymbol{X}_i. \tag{2.21}$$

In the context of source separation methods for drum transcription, with the exception of the convolutive model used by Virtanen [Vir06], the individual spectrograms $\boldsymbol{X}_i$ are considered to be the outer product of two vectors, a time basis $\boldsymbol{a}_i$ and a frequency basis $\boldsymbol{s}_i$,

$$\boldsymbol{X}_i = \boldsymbol{s}_i \boldsymbol{a}_i^{\mathsf{T}}. \tag{2.22}$$

In other words, it is assumed that the sound source produces fixed frequency content with a time-varying gain. When this is combined with the model (2.21) of source summation, the overall signal model can be expressed as a matrix product

$$\boldsymbol{X} \approx \boldsymbol{S}\boldsymbol{A}, \tag{2.23}$$

where the basis functions are combined into source spectral content matrix

$$\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_N], \tag{2.24}$$

and a matrix of time-varying gains

$$\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_N]^{\mathsf{T}}. \tag{2.25}$$

An example of the applicability of this model is illustrated in Fig. 2.9 in which three sources have produced the observed spectrogram.

Virtanen [Vir06, Chapter 4] extended the model so that each of the sound sources has a two-dimensional time-frequency representation, and the source spectrogram is produced by a convolution of an impulsive time-varying gain basis and the two-dimensional source spectrogram $\boldsymbol{S}_i$

$$\boldsymbol{X}_i = \boldsymbol{S}_i \otimes \boldsymbol{a}_i^{\mathsf{T}}. \tag{2.26}$$

The convolution operation $\otimes$ between a matrix $\boldsymbol{S}$ with J columns and a vector $\boldsymbol{a}$ is defined as

$$X(k,t) = \sum_{j=0}^{J-1} S(k, j+1)a(t-j). \tag{2.27}$$

Even though the separation results obtained with the convolutive model suggest that it is able to separate drum sources from a polyphonic mixture more reliably than the more simple model of (2.22), it has not been applied in drum transcription due to its considerably higher computational cost.

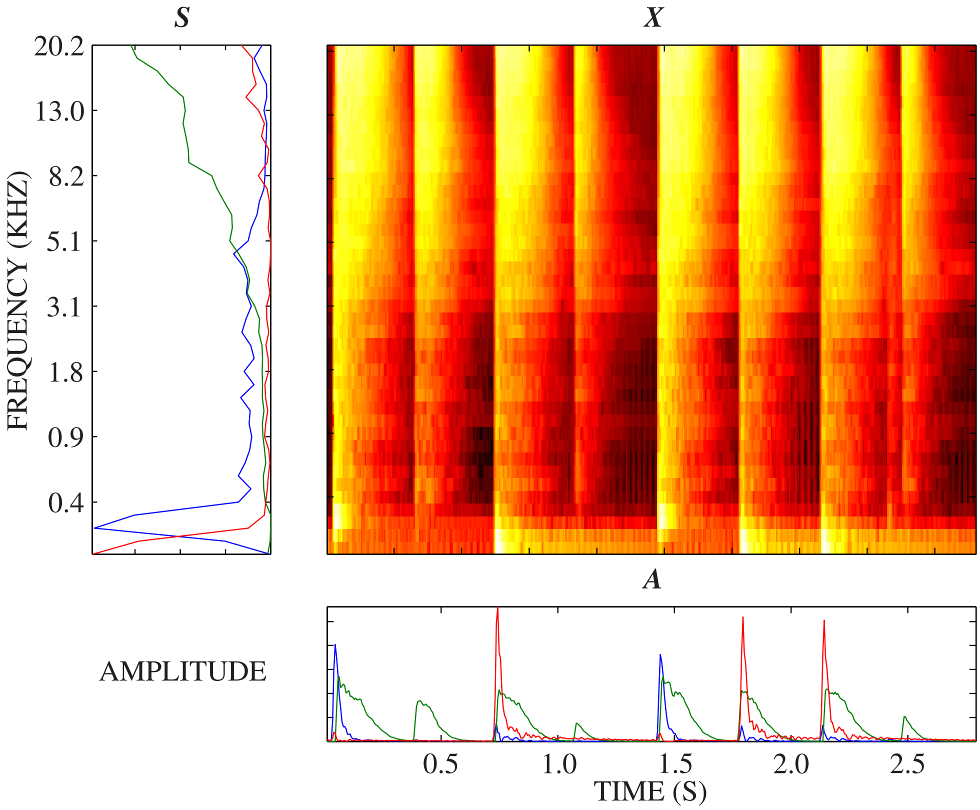


Figure 2.9: An example of factorisation of a spectrogram to three sources with fixed spectra $\boldsymbol{S}$ and time-varying gains $\boldsymbol{A}$. The input signal consists of a simple pattern of three drums: “SD+HH, HH, BD+HH, HH, SD+HH, HH, BD+HH, BD+HH, HH”. The mid-level representation is a mel-frequency spectrogram, and the basis functions are calculated unsupervised with non-negative matrix factorisation. The found basis have the following correspondence to the drums: green is hi-hat (HH), blue is snare drum (SD), and red is bass drum (BD). It can be noted how onsets in the gains $\boldsymbol{A}$ correspond to the played pattern.

Unsupervised Source Separation Methods

Unsupervised source separation methods aim to solve the separation problem without any specific prior templates for the sources, even the number of sources is unknown. In other words, the problem is to solve from

$$\boldsymbol{X} \approx \sum_{i=1}^{N} \boldsymbol{s}_i \boldsymbol{a}_i^{\mathsf{T}} \tag{2.28}$$

the number of sources N, the source spectra $\boldsymbol{s}_i$, and the gains $\boldsymbol{a}_i$ given only the input spectrogram $\boldsymbol{X}$. A common practice is to manually set the number of sources into a few different values, depending on the information available of the problem in advance, and select the appropriate one based on the obtained results.

Independent subspace analysis [Cas00] takes a single-channel input signal and calculates a spectrogram from it. Then principal component analysis (PCA) is used to determine the number of components (sources of variance) in the signal and to reduce the data dimensionality to equal the number of sources. PCA also decorrelates the sources, but they are not independent. The independence is obtained by using ICA on the reduced-dimensionality data. The resulting sources are considered to be the main sources in the input, and they can also be resynthesised. The use of ISA for drum transcription has been evaluated by FitzGerald [Fit04] and Orife [Ori01]. The main problem with the obtained result is that there are no restrictions implied on the sources and quite often they contain negative values in both basis function sets. Considering the data model of addition of magnitude spectrograms, the negative values do not have valid physical interpretation. In addition to the negativity problem, the recovered sources will have to be recognised either by some heuristic rules or by a supervised classifier.

Non-negative matrix factorisation (NMF) [Lee01] aims to solve the matrices $\boldsymbol{A}$ and $\boldsymbol{S}$ given the input $\boldsymbol{X}$ from the model (2.23) only by restricting all the three matrices to be non-negative. Lee and Seung [Lee01] proposed iterative update rules for estimating the component matrices to minimise Euclidean distance or Kullback-Leibler divergence between the input and the reconstruction. The divergence cost function between the original input matrix $\boldsymbol{X}$ and the reconstruction $\tilde{\boldsymbol{X}} = \boldsymbol{SA}$ is defined as

$$d_{\mathrm{DIV}}(\boldsymbol{X}, \tilde{\boldsymbol{X}}) = \sum_{i,j} X(i,j) \log \frac{X(i,j)}{\tilde{X}(i,j)} - X(i,j) + \tilde{X}(i,j), \tag{2.29}$$

where the summation is done over all indices in the matrices. The multiplicative update rules to minimise the cost function (2.29) are given as

$$\boldsymbol{A} \leftarrow \boldsymbol{A}.* \frac{\boldsymbol{S}^{\mathsf{T}}(\boldsymbol{X}./(\boldsymbol{SA}))}{\boldsymbol{S}^{\mathsf{T}}\boldsymbol{1}} \tag{2.30}$$

and

$$\boldsymbol{S} \leftarrow \boldsymbol{S}.* \frac{(\boldsymbol{X}./(\boldsymbol{SA}))\boldsymbol{A}^{\mathsf{T}}}{\boldsymbol{1}\boldsymbol{A}^{\mathsf{T}}}, \tag{2.31}$$

where $\boldsymbol{1}$ is a all-one matrix of the same size as $\boldsymbol{X}$, and $.*$ and $./$ denote elementwise multiplication and division, respectively [Lee01]. In the context of drum transcription unsupervised NMF has been used to separate the drum signal from a polyphonic mixture, as proposed by Helén and Virtanen [Hel05].

Dittmar and Uhle proposed a hybrid between "segment and classify" and "separate and detect" approaches in [Dit04]. The system is designed to transcribe drums from polyphonic music. After detecting sound onsets, one frame of spectrogram is extracted from each onset location. The frames are gathered into a new matrix that is analysed with ISA-like approach with non-negative independent component analysis [Plu03] to produce spectral basis vectors that are used to extract time-varying gains from the spectrogram of the input signal. The different drums are recognised from the spectral basis vectors with a k-NN classifier. This work was extended by Gruhne et al. [Gru04] to produce MPEG-7 descriptions of drum patterns.

Similar approach to automatically generate spectral basis functions for a supervised separation method (prior subspace analysis, described in the next section) with basis vector clustering was proposed by FitzGerald et al. [Fit03a]. However, instead of a classifier to recognise the found drums, a rule-based approach was employed. Moreau and Flexer [Mor07] presented preliminary results of another hybrid approach. The input signal spectrogram is decomposed using NMF, a set of spectral and temporal features are extracted from the basis functions, and the components are recognised to be one of the target drums or not a drum with a k-NN classifier.

Virtanen proposed to use constraints of non-negativity, temporal continuity and temporal sparseness to separate the sources with non-negative sparse coding [Vir03]. The non-negativity constraint suits the physical interpretation of the data model, as magnitude spectrograms are always non-negative and they can be summed in reality only with non-negative gains. The temporal continuity and sparseness constraints prefer the solved gains $\boldsymbol{a}_i$ to change in a smooth way and to contain only few large values. The separation algorithm was evaluated by transcribing bass and snare drums from polyphonic music. The algorithm separates a fixed number of most prominent sources, recognises the bass and snare drum sources by comparing the obtained spectral basis functions to pre-calculated templates, and finally detects sound onsets from the recovered gains.

For a more complete review of unsupervised source separation methods applied on monaural signals, but not restricted to drum transcription, refer to, e.g., the thesis of Virtanen [Vir06].

Supervised Source Separation Methods

As the name suggest, supervised source separation methods utilise more detailed information of the properties of the target signals to accomplish the segregation. Most often the information is in the form of the spectral content of the target drums.

Prior subspace analysis, as proposed by FitzGerald et al. [Fit03c], includes prior knowledge of the drum sound properties in the separation step. It is done by calculating initial guesses, or templates, of the frequency basis functions from example hits. The initial templates are then organised into a matrix $\boldsymbol{S}_{\mathrm{PR}}$ and approximations of the time-varying gains $\hat{\boldsymbol{A}}$ are obtained through

$$\hat{\boldsymbol{A}} = \boldsymbol{S}_{\mathrm{PR}}^{+}\boldsymbol{X}, \tag{2.32}$$

where $\boldsymbol{S}_{\mathrm{PR}}^{+}$ is the pseudoinverse

$$\boldsymbol{S}_{\mathrm{PR}}^{+} = (\boldsymbol{S}_{\mathrm{PR}}^{\mathsf{T}}\boldsymbol{S}_{\mathrm{PR}})^{-1}. \tag{2.33}$$

Since the obtained gains are not statistically independent, they are further processed with ICA to obtain independent time-varying gains $\boldsymbol{A}$. Finally, new estimates of the frequency basis functions are obtained by another pseudoinverse

$$\boldsymbol{S} = \boldsymbol{X}\boldsymbol{A}^{+}. \tag{2.34}$$

The knowledge of the spectral properties of the sources improves the recognition result, but still does not solve the negativity problem inherent to the used computations, and the ICA step may permute the order of the sources requiring them to be recognised despite the use of template initialisation.

In [Fit03b] FitzGerald et al. experiment to improve the performance of PSA in the presence of pitched instruments, i.e., other instruments. Hi-hat recognition performance was improved by weighting the input spectrogram $\boldsymbol{X}$ by the average power spectral density over the signal. Effectively this emphasises the high frequencies where most of the energy of cymbals resides. The interference of the pitched instruments with bass and snare drums was reduced by thresholding the initial gain estimates $\hat{\boldsymbol{A}}$. The thresholding retained the most important peaks in the gain, but removed the lower level interference.

The basis vector negativity and permutation problems can be solved when the source independence assumption is relaxed and supervised NMF is used to solve the temporal basis vectors $\boldsymbol{a}_i$, as proposed by Paulus and Virtanen [P3]. The non-negative matrix factorisation based prior subspace analysis (NMF-PSA) method proposed therein calculates spectral templates $\boldsymbol{s}_i$ for each of the target drums by applying the full unsupervised decomposition to a large number of training hits, and constructs the matrix $\boldsymbol{S}$ from the individual template spectra with (2.24). Then this matrix is kept fixed while the time-varying gains are recovered by applying (2.30) until the solution converges. Finally, the sound event onsets are detected from the gains with a method approximating the relative intensity detection properties of human hearing (after [Kla99]). The NMF-PSA method was proven to perform very well when the input signal and the model match, i.e., when there are very few non-target drums in the signal. Similar approach was independently developed by FitzGerald et al. [Fit05], but they extended it also to pitched sounds by introducing an additional term modelling the shift of the spectrum between different pitches.

In a studio, a drum kit is often recorded having close microphones near the membranes of each membranophone, overhead microphones over the drummer listening the whole kit and some of the room ambiance, and possibly close microphones also near the cymbals, especially hi-hat. The motivation to use the close microphones is that they capture mainly the acoustic signal of the drum they are closest to, and as a result the well-separated signals are easier to manipulate and mix in the later stages of record production. As these multichannel signals are available in the studio, Alves et al. [P4] proposed to extend the NMF-PSA of [P3] to handle multichannel recordings as the input. Instead of calculating the spectrogram $\boldsymbol{X}$ from a single-channel signal, it is calculated from each of the N_C input channels and they are combined by

$$\hat{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \\ \vdots \\ \boldsymbol{X}_{N_\mathrm{C}} \end{bmatrix}. \tag{2.35}$$

Similar stacking is done when calculating the spectral templates for the drums to produce multichannel version of the template matrix by

$$\hat{\boldsymbol{S}} = \begin{bmatrix} \boldsymbol{S}_1 \\ \boldsymbol{S}_2 \\ \vdots \\ \boldsymbol{S}_{N_\mathrm{C}} \end{bmatrix}. \tag{2.36}$$

With these two stacked matrices, the time-varying gains $\boldsymbol{A}$ are calculated from

$$\hat{\boldsymbol{X}} \approx \hat{\boldsymbol{S}}\boldsymbol{A} \tag{2.37}$$

by applying the update rule (2.30). The onsets are detected from the estimated gains with the same procedure as in [P3].

Proposed Methods

Among the publications included in this thesis, [P3] and [P4] belong to the “separate and detect” family. Both of them implement an NMF-PSA method, the former for single-channel input, and the latter for multichannel input available, e.g., in recording studios.

2.3.6 HMM Recognition Approach

With “segment and classify” methods the segmentation step has to make a hard decision of the locations the onsets. Considering the later steps of the transcription methods, it would be crucial that the detection would return at least all the true onsets, and desirable that the number of extraneous onsets would be as small as possible. Finding this balance may be difficult in many cases. When the segment classification can be considered to correspond to isolated word recognition in speech recognition, the direct analogue of continuous speech recognition can also be applied in drum transcription. The main difference between the isolated and continuous recognition is that in the latter there is no explicit segmentation before recognition, but the process itself decides the segmentation simultaneous with recognising the events. In other words, isolated recognition aims to decide the model that is most likely to have produced the observed sequence, whereas continuous recognition not only decides the most likely state sequence, but also the most likely model sequence to have produced the observation

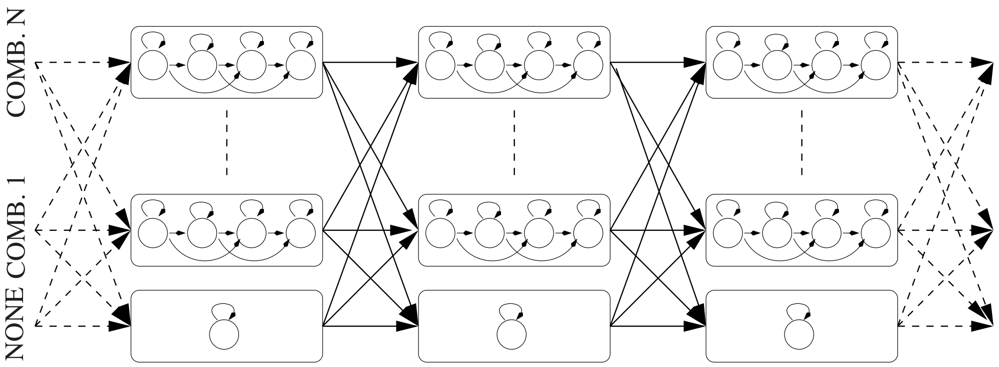


Figure 2.10: Illustration of the basic idea of drum transcription with HMMs modelling drum combinations. The decoding aims to find the optimal path through the models in view of the observed acoustic information.

sequence. The decoding can be done, e.g., with explicit separate models in token passing [You89], or by concatenating the individual models to a larger compound model and applying Viterbi decoding. In either case an additional parameter set has to be decided: the transition probabilities between the individual models.

Ryynänen showed that HMM recognition is able to decide note onsets in pitched instrument transcription [Ryy08a]. Similar approach was adopted by Paulus and Klapuri in [P5] for drum transcription from polyphonic music. As with earlier classification task, the HMM modelling can focus either on drum combinations or detecting the presence of individual drums. The difference of these is illustrated in Figs. 2.10 and 2.11: with combinations, the whole signal to be transcribed will be covered by the different combinations (including a background model when no target drum is playing), whereas with the detectors, each target drum is handled separately from the others and only a sequence of "sound" and "silence" for each is determined. The sound models are trained from individual sound events that are segmented from the training signals, and the material that is not used to train the sound models is used to train the background (or silence) model. In the evaluations when compared with a "segment and classify" methods using an SVM classifier [Tan05] and a "separate and detect" method using NMF-PSA [P3], the HMM method outperforms the reference methods both with signals containing only drums, and with polyphonic music.

2.4 Musicological Modelling

Majority of the drum transcription systems discussed above operate mainly based on the acoustic information only. Considering once again the analogue to speech recognition, this corresponds to attempting to recognise speech one phoneme at a time without any knowledge of the contextual dependencies between the phoneme to be recognised and other phonemes in words and sentences. The context, however, is able to provide a considerable amount of useful information to the recognition process, and modelling it

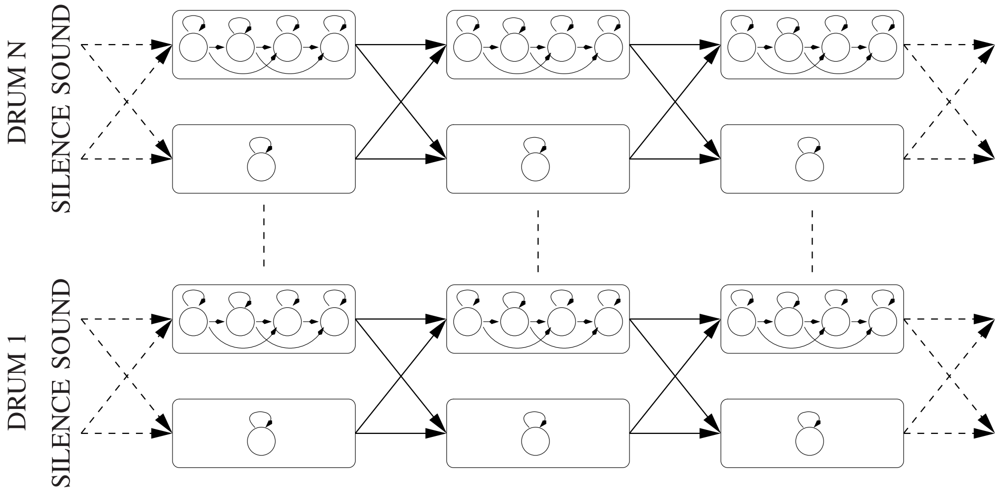


Figure 2.11: Illustration of the basic idea of drum transcription with HMMs modelling of drum detectors. Each target drum is associated with two models, “sound” and “silence”, and the decoding is done for each drum independently from the others.

has become an important field of study [Jur00]. The usefulness of modelling temporal dependencies of musical events (or *musicological modelling*) has been demonstrated also in melody, bass line, and chord transcription with HMMs by Ryynänen [Ryy08a]. In his methods, the HMMs model the feature evolution during a note event, and musicological modelling describes the temporal dependencies between consecutive notes. Considering that the whole basis of drum patterns is that they repeat and that the repeats are responsible for creating a sensation of rhythm, use of musicological modelling would be intuitively justified. Some drum transcription systems have attempted to incorporate musicological modelling and will be discussed in the following.

2.4.1 Enabling from Crisp Classifiers

The high-level musicological models usually provide a probability value for the events given the context. To be able to utilise these models meaningfully, the lower-level acoustic recognition result has to also be expressed as a probability value. With the “segment and classify” methods this means that the output of the classification algorithm has to be a value that can be interpreted as a probability. In case the classifier is some Bayes classifier, the output is a posterior probability already. However, if the recognition is done with classifiers producing crisp (as opposed to fuzzy or soft) decision, the output has to be moderated somehow. With the exception of SVM classifiers, there has not been much research on how this moderation should be done. Several methods for moderating the SVM output were compared by Rüping [Rüp04], and the results suggest that the logistic regression approach proposed by Platt [Pla99] would be most suitable.

The SVM output y_i given the input $\boldsymbol{x}_i$ is a real number given by (2.15). In the actual classification only the sign of y_i is used to determine to which of the two classes the input vector belongs to. Platt proposes to interpret the output value to the posterior probability of the input to belong into the positive class (i.e., $c_i = 1$) by a sigmoidal mapping

$$p(c_i = 1|y_i) = \frac{p(y_i|c_i = 1)p(c_i = 1)}{p(y_i|c_i = 1)p(c_i = 1) + p(y_i|c_i = -1)p(c_i = -1)} \tag{2.38}$$

$$= \frac{1}{1 + e^{\alpha_1 y_i + \alpha_2}}, \tag{2.39}$$

where α_1 and α_2 are parameters that are determined from training data with a variant of Levenberg-Marquardt algorithm given in [Pla99]. The sigmoidal mapping (2.38) assumes that the logarithm of the posterior odds for the negative class can be modelled as a line, i.e.,

$$\alpha_1 y_i + \alpha_2 = \log \frac{p(c_i = -1|y_i)}{p(c_i = 1|y_i)}. \tag{2.40}$$

This mapping has been utilised to enable the musicological modelling by Paulus and Klapuri [P1], and Gillet and Richard [Gil05c, Gil07].

2.4.2 Models

Several models of the context in the transcription process have been proposed ranging from simple feature vector concatenation to generalised N-grams. These will be discussed in the following.

Sliding Windows

Possibly the simplest way to include contextual information in the recognition of a drum sound event is to utilise the neighbouring feature vectors in addition to the feature vector directly from the sound event. Concatenating the feature vectors allows the use of any supervised learning algorithm for the actual classification task [Die02]. This kind of feature vector concatenation has been used in drum transcription by Gillet and Richard [Gil05b] who use an SVM classifier with combined feature vector of two consecutive sound events. Even this quite simple context modelling provides improvement over using only a single feature vector.

Conventional N-grams

More explicit sequence modelling method often met also in speech recognition is the use of N-grams. An N-gram of length N uses the $(N-1)^{th}$ order Markov assumption stating that the probability of event s_i depends only on the $N-1$ previous events

$$p(s_i|s_{1:(i-1)}) = p(s_i|s_{(i-N+1):(i-1)}), \tag{2.41}$$

where a sequence of events is denoted in a more compact form by

$$s_{1:(i-1)} \equiv s_1, s_2, \ldots, s_{(i-1)}. \tag{2.42}$$

The N-gram probability $p(s_i|s_{(i-N+1):(i-1)})$ can be estimated from training data as the ratio of the number of occurrences of the sequence $s_{(i-N+1):(i-1)}, s_i$ to the number of the prefixes $s_{(i-N+1):(i-1)}$ of length $N-1$

$$p(s_i|s_{(i-N+1):(i-1)}) = \frac{\#(s_{(i-N+1):(i-1)}, s_i)}{\#(s_{(i-N+1):(i-1)})}. \tag{2.43}$$

The number of probabilities to be estimated with a vocabulary of size V is V^N causing it to increase rapidly as a function of the vocabulary size and the N-gram length. As a consequence, observing all the sequences in the training data will become increasingly improbable, and using the pure count ratio of (2.43) will cause zero-probability entries. Several solutions for this problem have been proposed, including assigning small probability value to the events never seen with Witten-Bell [Wit91] or Good-Turing [Goo53, Chu91] discounting, or estimating the probabilities from lower-order N-grams with back-off [Kat87] or with deleted interpolation [Jel80].

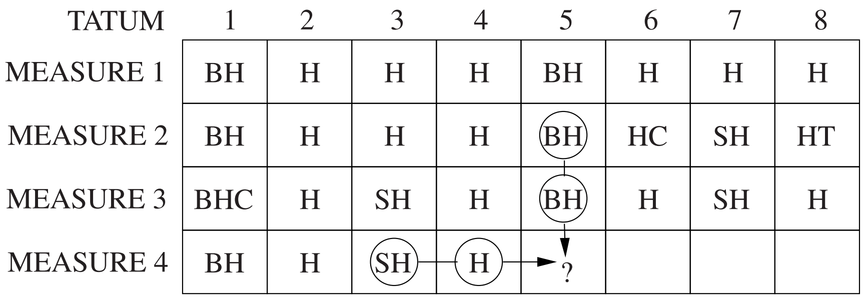
In addition to the zero-frequency problem, determining the length of the context is not straightforward. Variable order Markov models have been proposed to solve this problem by estimating the overall information gain of increasing the context length for each sequence [Ron96]. For more details on N-grams and language modelling, refer to [Jur00].

In drum transcription the data sparsity problem arises quite early if the target classes are defined as drum combinations due to the low occurrence frequencies of a majority of the combinations, as illustrated earlier in Fig. 2.8. To solve this problem, Paulus and Klapuri [P1] and Gillet and Richard [Gil07] have proposed to estimate the N-grams for individual drums and to produce the combination probabilities by

$$p(s_i|s_{(i-N+1):(i-1)}) = \prod_{r_j \in s_i} p(r_{j,i}|r_{j,(i-N+1):(i-1)}) \prod_{r_j \notin s_i} \left(1 - p(r_{j,i}|r_{j,(i-N+1):(i-1)})\right). \tag{2.44}$$

In the equation above, $r_j \in s_i$ denotes that the drum r_j is present in the combination s_i, and $r_j \notin s_i$ that it is not. Combining the individual probabilities in this way assumes that the drums occur independent from each other, which strictly speaking does not hold and the the decomposition may produce false estimates for some combinations. As suggested in [P1] this may be alleviated by incorporating combination prior probabilities to the probability estimate given by (2.44). Regardless of the modelling inaccuracies with combination modelling, the clear benefit from using this decomposition is that training data is utilised more efficiently.

N-grams have been used in drum transcription by Paulus and Klapuri [P1], and Gillet and Richard [Gil04]. In the first method, the acoustic recognition is done with a combination of an SVM for silence detection and GMMs for combinations of 7 target drums from a tatum grid segmented signal. The N-grams are added as a post-processing step and a Viterbi-like decoding is done to solve the most likely drum combination sequence. The latter method relies on onset detection, acoustic recognition with GMMs, and then applying the N-gram models to model the events in the



TATUM	1	2	3	4	5	6	7	8
MEASURE 1	BH	H	H	H	BH	H	H	H
MEASURE 2	BH	H	H	H	BH	HC	SH	HT
MEASURE 3	BHC	H	SH	H	BH	H	SH	H
MEASURE 4	BH	H	SH	H	?			

Figure 2.12: Illustration of prediction with conventional (horizontal arrow) and periodic (vertical arrow) N-grams of the length 3. Time scale is quantised to tatum grid and folded to rows from musical measures. The target instruments are denoted with bass drum (B), crash cymbal (C), hi-hat (H), snare drum (S), and tom-tom (T).

sequence. This all was formulated viewing the resulting model as an HMM. The publication [Gil04] does not present a comparison of plain GMM recognition with the HMM-like method, so no conclusions can be made concerning the effect of using sequence modelling.

Periodic N-grams

In speech and language modelling in general, the sequential dependencies are very strong: the few directly preceding units predict the following unit effectively. However, with drum sequences this is not the case, as there is usually a short pattern that is repeated (possibly with small variations), and the repeats generate the rhythmic percept. This suggests that with drums it is not the directly preceding context that is important, but the context from a pattern length earlier instead. This is illustrated in Fig. 2.12, where the task is to predict the content of the location marked with the question mark, either based on the two directly preceding events “SH, H”, or based on the two periodically preceding events “BH, BH”.

Paulus and Klapuri [P1] proposed the concept of periodic N-grams which would utilise this periodic dependency by modifying the N-gram definition into

$$p(s_i|s_{1:(i-1)}) = p(s_i|s_{i-(N-1)L}, s_{i-(N-2)L}, \ldots, s_{i-L}). \tag{2.45}$$

The interval length L defines the locations from which the context for the prediction is taken. Setting the value to $L = 1$ reduces the model to a conventional N-gram model. The authors propose that the interval length should be set to a value corresponding to the length of the musical measure in the piece, which usually corresponds to the pattern length (or half of it). Naturally, this assumes that the temporal quantisation is done on a constant, e.g., tatum grid instead of onset detection based segmentation to allow skipping fixed number of events in the history.

Generalised N-grams

The N-gram concept was taken even further by Gillet and Richard in [Gil07] in the generalised N-gram models. The prediction for an event may depend on events that occur $\tau = \{\tau_1, \tau_2, \ldots, \tau_{(N-1)}\}$ time units earlier, i.e.,

$$p(s_i|s_{1:(i-1)}) = p(s_i|s_{i-\tau_{(N-1)}}, s_{i-\tau_{(N-2)}}, \ldots, s_{i-\tau_1}). \tag{2.46}$$

In case $\tau = \{N-1, N-2, \ldots, 2, 1\}$ the model corresponds to conventional N-grams, and with $\tau = \{(N-1)L, (N-2)L, \ldots, 2L, L\}$ the model corresponds to the periodic N-gram model. Using arbitrary intervals instead of directly or periodically consecutive ones allows to focus on different time scales in the same model. Measured with the information content of the models, the generalised N-grams showed to contain more predictive power than conventional or periodic N-grams by looking at directly preceding and a pattern length earlier events.

Pattern Complexity

The use of N-gram models requires that the musicological model is trained, which always requires balancing between the prediction power and overfitting the model. Additionally, using periodic or generalised N-grams requires the generation of a reliable grid and pattern length estimation. To avoid the problems that these steps may cause, Gillet and Richard [Gil07] proposed to define a fitness measure for the transcription result and use a genetic algorithm to optimise it. In other words, no explicit musicological model is applied, but instead some properties of a "good" transcription result are defined. The fitness function consists of two terms: the acoustic recognition likelihood and a complexity measure as a negative term. The complexity measure estimates the Kolmogorov complexity of the hypothesised transcription result by a compression algorithm. The motivation for the complexity measure is that drum sequences are constructed from regular and repeating patterns. Considering compression, these both properties allow more efficient encoding of the data, thus the smaller the complexity of a sequence is, the more correct it is. This assumption was confirmed by Gillet and Richard when they noted that the average complexity of the ground truth for their evaluation data has a smaller complexity than the transcription result obtained with only acoustic recognition. In the evaluations the fitness function based post-processing of the transcription improved the recognition result to par with the one obtained by generalised N-grams. However, the authors themselves criticise the computational cost of the genetic algorithm employed in the method.

Pattern Periodicity

In addition to the (periodic) N-grams for modelling the periodicity of drum patterns, some less formal approaches have been taken. A common theme has been to utilise the property that drum patterns repeat in somewhat similar form over and over again.

Sillanpää et al. [Sil00] utilised the local periodicity to predict the next occurrence of drum hit to follow after an interval of length equal to the interval from the preceding hit. Gillet and Richard [Gil05c] combined the probabilities of acoustic recogniser from three positions: the transcription location, a pattern length earlier, and a pattern length after the location. Even though the method was shown to improve the recall rate of the method, the precision rate lowered more and the overall performance was degraded. Yoshii et al. [Yos06] propose a multipass approach where periodic patterns are searched from the transcription result and the high-level information is then used to adapt the templates used in the acoustic recognition. The evaluation results suggest that the periodicity utilisation with acoustic model adaptation improve the transcription accuracy slightly.

Compositional Rules

All of the methods described above rely on supervised classification in the low-level acoustic recognition before any musicological model can be applied. The supervised classification requires that the input patterns are produced with methods resulting to similar (in view of the features used) observations. However, not all drum-like sounds are produced with drums, but also using drum onomatopoeia[2] and tapping any surrounding objects.

If the sound set used is not known in advance, training a supervised classifier is difficult if not impossible. In such a case the classification has to be done with an unsupervised method, i.e., with clustering. In the method by Bello et al. [Bel06] the sound events are clustered into three groups, and the groups are labelled as “low”, “mid”, or “high” based on their average spectral content. These categories are claimed to be enough to describe the rhythmic content of drum loops with the stereotypical class assignments: bass drum belongs to “low”, snare drum and tom-toms are in “mid”, and cymbals belong to “high”. After mapping a normal drum set to these three categories for evaluation, the method showed quite high performance. Further development of the method was presented by Ravelli et al. [Rav07] who incorporate also some information of the metrical structure to the resulting description by denoting whether the sound event occurs on or off-beat. The resulting string representation of drum loops is used in pattern matching and manipulation. Even though the presented evaluations used input signals generated with regular drums, the method should work also with any arbitrary input as long as the spectral relationships of the three clusters are present.

The most difficult case from the point of view of drum transcription is when the sounds used do not resemble real drums in any way and they do not have any clear acoustic relationships that could be utilised in determining the identities as was done above, e.g., the tapping of office supplies on a desk. Such input was considered by Paulus and Klapuri [P2], utilising compositional rules to determine the identities of

[2] Imitating the sound produces by a drum with speech-like utterances, e.g., bass drum with “bum”, snare drum with “tschak”, and hi-hat with “ss”. Beatboxing is an advanced form of this imitating the sounds very closely with less focus on speech-likeness.

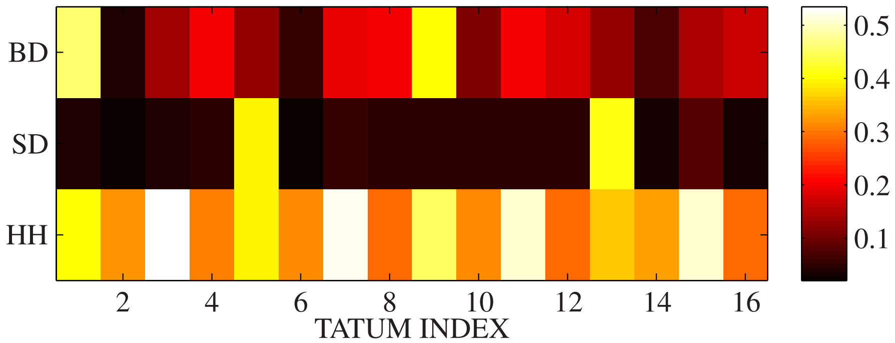


Figure 2.13: Illustration of the probabilities for different rhythmic roles to occur at different tatum indices within a musical measure. The probabilities are calculated from the 16 tatums per pattern pieces in a commercial MIDI drum pattern database [Vam] allowing only one role to be active at each tatum index. The different rhythmic roles are denoted by BD (bass drum), SD (snare drum), and HH (hi-hat).

the events. The method creates a tatum grid over the signal with the added knowledge of musical measure. The located sound events are clustered and information regarding the cluster membership of each onset is stored to the nearest tatum grid points. Then a probabilistic model is used to assign each cluster with a musically meaningful drum role label. The model allows only one role to be present at a time, and considers the probability of role c to be present at the j^{th} tatum grid point of a measure which has the length of J tatums as $p\left(c|(j,J)\right)$. These probabilities in the case of pattern length of 16 tatums are illustrated in Fig. 2.13. Having a mapping M from clusters to rhythmic role labels, the cluster sequence can be transformed into a rhythmic role sequence, and the total probability over it calculated with

$$p(M) = \prod_i p\left(c_i|(j_i,J)\right). \tag{2.47}$$

In (2.47) i indexes the tatum points of the signal, c_i is the role at tatum index i after applying the mapping M, and j_i is the in-measure index of tatum i. The optimisation task is to find the mapping M_{OPT} that maximises the overall probability

$$M_{\text{OPT}} = \underset{\text{M}}{\text{argmax}}\{p(M)\}. \tag{2.48}$$

The evaluations showed that using such a compositional rules it is possible to assign musically more meaningful drum names to events produced with arbitrary sounds that do not give any acoustic clue of the drum identity.

Ellis and Arroyo [Ell04] proposed to solve similar rhythmic basis functions in a continuous form by analysing a large set of rhythmic patterns with PCA. The experiments showed that the eigenrhythm method was able to recover some stereotypical rhythmic components, but the final application of these patterns in genre classification was less successful.

Overall, the results obtained by different authors in utilising musicological modelling in drum transcription suggest that the use of high-level information is useful. Contrary to language modelling in speech recognition, it seems that drum transcription does not necessarily benefit from conventional N-gram models, but instead basing the prediction of observations rhythmically meaningful periods apart would be more useful. However, the improvements obtained by using the musicological models have so far been quite modest. The reason for this may be that the low-level acoustic recognition result is still not accurate enough to serve as a reliable basis for top-down processing with these models. It should also be noted that to date, no explicit musicological modelling has been applied with "separate and detect" methods.

Proposed Methods

From the publications included in this thesis [P1] proposes the use of periodic N-grams to assist in the transcription, and compares their performance with conventional N-grams. Additionally, the decomposition of drum combinations for the N-gram probability estimation and recomposition in the transcription phase are presented. [P2] proposes to use compositional rules to assign drum labels for sequences produced with arbitrary sound sets. The connected HMM method [P5] uses simple bigrams (N-grams of length 2) for modelling the transition probabilities between consecutive models.

2.4.3 Non-audio Modalities

All of the methods described above are designed to handle only an acoustic signal as the input. In some cases there is also visual information available to be utilised in the transcription. In [Gil05a] Gillet and Richard combine an existing acoustic transcription method to utilise also video data recorded from the drummer playing. The results suggested that even though the video information alone did not have very good performance, they allowed to improve the performance of the acoustic recognition system slightly. The visual analysis system was developed further by McGuinness et al. [McG07] and the results agreed with those obtained by Gillet and Richard.

2.5 Summary

This chapter has provided an overview of the drum transcription task and the methods proposed to solve it. The main methods have been gathered to Table 2.1 on page 17 to provide an overview of the authors and approaches. The problem difficulty ranges from recognising individual drum hits to transcribing multiple simultaneous drums from polyphonic music. Because of the multifacetedness of the problem, several different approaches have been adopted and the main principles have been described. In some of the earlier publications the target drum set has contained ambitiously a large set of drums, but as the complexity of the problem has been understood, many of the more recent publications have targeted only bass drum, snare drum, and hi-hat. These three drums have been successfully transcribed from a signal containing only drums, but the performance with polyphonic music input still leaves some room for improvement. Recently, more and more methods have utilised either some sort of model adaptation, musicological modelling, or even both to improve the transcription performance. Though the problem of drum transcription may seem initially simple, the performance of the current state-of-the-art methods suggests that this is not the case in reality and still more work is required to obtain reliable result.

Chapter 3

Music Piece Structure Analysis

> Form and content: the two poles around which, traditionally discussions of works of art revolve. Form is supposed to cover the shape or structure of the work; content its substance, meaning, ideas, or expressive effects.
>
> Middleton [Mid99]

THE difference between arbitrary sound sequences and music is not well-defined: what is random noise for someone may be ingenious musical composition for somebody else. What can be generally agreed upon is that it is the structure, or the relationships between the sound events that create the perception of musicality. This structure starts from the level of individual notes, their acoustic properties, durations, and intervals in time and frequency. Notes form larger structures, phrases, chords, and chord progressions, and these again form larger and larger constructs in a hierarchical manner. At the level of musical pieces the subdivision can be made to musical sections, such as *intro*, *chorus*, and *verse*, and recovering a description of this structure is what is meant by *structure analysis* in this thesis. The kinds of form and the assumptions made here fit mainly Western popular music, but some of the employed principles can be utilised to analyse other kind of music, too, as demonstrated by the work of Müller and Kurth on classical music [Mül07b]. Some of the musical sections that will be referred to in the rest of this chapter include the following:[1]

- *Intro* - Usually a unique section in the beginning of the piece. In many pop songs, may contain a version of the chorus melody or chord progression.

- *Outro, ending* - A unique section in the end of the piece acting as the opposite of intro.

[1] These should not be taken as authoritative definitions, but as more informal descriptions.

- *Verse* - First of the two main musical sections in a majority of pop songs. The verse often carries the story of the piece in a narrative form, and different occurrences of it often have slightly differring instrumentations and lyrics.

- *Chorus, refrain* - Second of the two main musical sections in pop songs, providing contrast to the verse. In many cases, the most often repeating and the most energetic section in the song. Chorus is likely to remain almost identical across the repetitions. In some pieces, the last repetition of the chorus may be modulated one or two semitones up.

- *Solo* - An occurrence of verse, chorus, or some other part in which the main vocals are replaced with instrumentals.

- *Bridge* - A contrasting, interconnecting section between other parts, e.g., chorus and verse.

- *Break, breakdown* - A section with greatly reduced instrumentation providing a breather before gradually building back to the main parts.

- *Interlude* - A more generic name for contrasting sections with differring instrumentation or rhythm. Break and bridge can be considered to be more specialised instances of an interlude.

- *Theme* - A section with reoccurring common melody with small variations. Used mainly with instrumental pieces to replace chorus and verse.

Some idea of the occurrence frequencies of the musical parts mentioned above can be obtained from Fig. 3.1 which illustrates the most often occurring musical part labels[2] in three different data sets. The first data set, *UPF Beatles*, consists of 174 songs by The Beatles. The sectional form of the pieces has been annotated by musicologist Alan W. Pollack [Pol01], and the segmentation time stamps were added at Universitat Pompeu Fabra. The second data set, *TUTstructure07* consists of 557 pieces from various popular music genres [P8]. The data set was collected and annotated at Tampere University of Technology. The third data set, *RWC Pop*, consists of 100 pieces from the Real World Computing Popular Music Database [Got02] representing typical Japanese and American chart hits from 1980's and 1990's. The structure annotations were provided as a part of the AIST annotations [Got06].

Structure in music is instantiated by two main principles: *sequences* and *repetitions*. The sequences are formed by consecutive notes and chords following each other. The repetition is important an factor, as Middleton states in [Mid99]: "It has

[2] One or more of following qualifiers can be applied to the labels: <label> and a, b, c, or d: variations of the part in the same piece; pre-<label>, post-<label>: a clearly separate part, but always connected to <label> either by preceding or following it; <label>h: only a half of <label>, <label>s: part can be interpreted to be either <label> or solo, <label>i: instrumental version of the part. Furthermore, labels a, b, and c have been used to denote parts without distinctive semantic label.

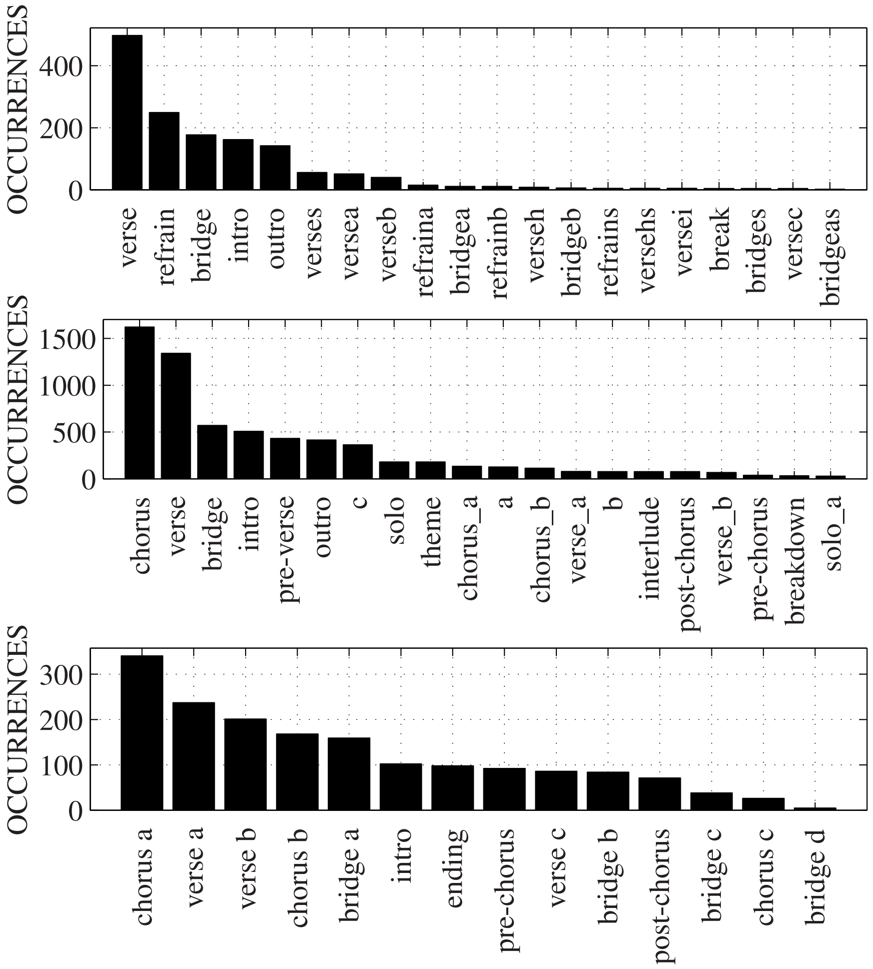


Figure 3.1: Number of occurrences of most frequently occurring musical part labels in three data sets: *UPF Beatles* (top), *TUTstructure07* (middle), and *RWC Pop* (bottom). Only labels that have more than one occurrence and belong to the 20 most frequently occurring labels in each data set are shown.

often been observed that repetition plays a particularly important role in music - in virtually any sort of music one can think of, actually. . . . In most popular music, repetition processes are especially strong." The repetition takes place especially in the rhythmic parts of the piece (e.g., repeating drum patterns), but also the musical sections are often repeated. Some idea of the amount of musical part repeats is provided by Fig. 3.2: the left column of the figure illustrates histograms of the number of musical part occurrences in the pieces in the three data sets, and comparing them to the right column illustrating the number of unique musical parts. The statistic show that, on average, a musical part in a piece occurs approximately twice. However, as shown in Fig. 3.3, a considerable number of parts have only single occurrences in pieces.

In the rest of the chapter, the following terminology will used:

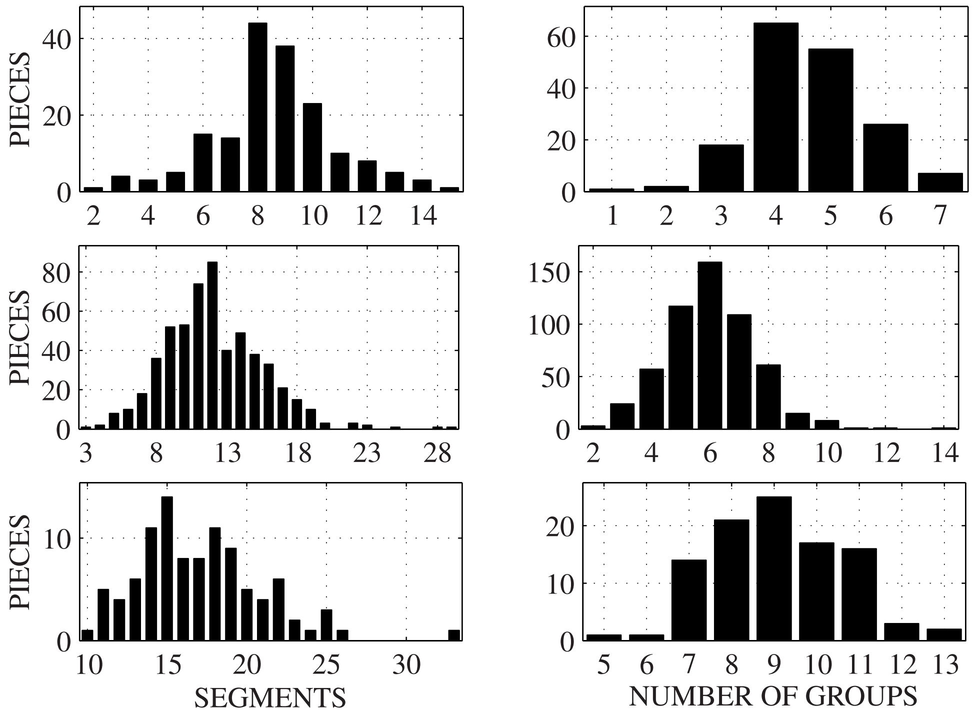


Figure 3.2: Histograms showing the number of segments (left) and the number of unique groups (right) in a piece, calculated from three data sets: *UPF Beatles* (top), *TUTstructure07* (middle), and *RWC Pop* (bottom).

- Part - a musical concept that can refer to either a single instance or all the instances of a musical section, such as chorus or verse.
- Segment - a technical concept referring to the temporal range of a single occurrence of a musical part.
- Group - one or more segments that represent all the occurrences of the same musical part.
- Label - a musically meaningful name for a group.

For a tutorial and a review of earlier methods for music structure analysis from acoustic signals, refer to the book chapter by Dannenberg and Goto [Dan08].

3.1 Problem Definition

Similar to the drum transcription problem, different researchers have pursued slightly different goals under the title "structure analysis". A common theme, however, is that the temporal scale of the analysis has been approximately same in all the cases, i.e., the level of musical sections. The methods discussed in the following operate by taking

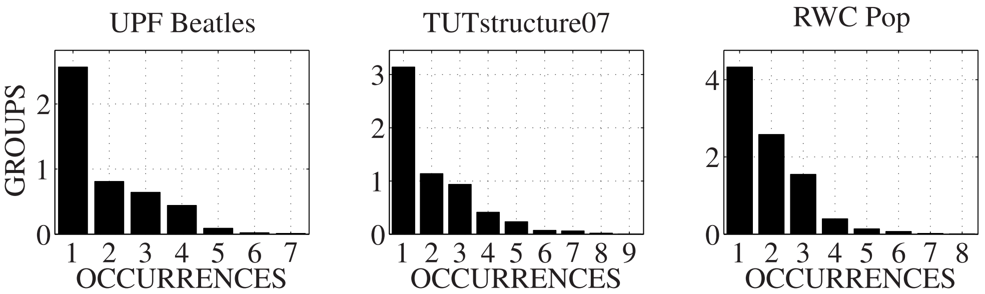


Figure 3.3: Histograms showing the average number a musical part occurs in a piece, calculated from three data sets: *UPF Beatles*, *TUTstructure07*, and *RWC Pop*.

an acoustic signal as the input and produce some information about the structure. The output of the discussed methods varies from images created for visualisation, to a representation where each found section is defined by its time range and given some meaningful label.

In the simplest form, no explicit structural analysis is performed, but some transformation of the acoustic features of the piece are used to provide a visualisation of the structure of the piece. Even though the visualisation method would not utilise any formal model, the ability of humans to locate patterns can extract useful information from the result.

A second category of methods aim only to segment the song, i.e., locate points where the instrumentation or some other characteristics changes in the extent that a human listener would say that there is a musical boundary at that point. The segmentation info can be applied, for example, in a playback skipping to the point of next musical change, and having an accurate segmentation enables also more complex structure analysis to be done with less computations.

Many of the proposed methods have been motivated by creating music thumbnails, i.e., selecting a representative part or parts of the piece that provide a compact preview of the piece, for example, at online music stores. In some cases the chorus is defined to be the most representative part of the piece and the methods aim at locating it. This choice is often motivated by the fact that in popular music the chorus section is usually repeated in a more or less identical form several times during the song and it is the “catchiest” part, thus making it a good choice for the thumbnail. Thus, some authors define the most often repeating section to be the chorus and aim at locating it [Got03, Bar05, Ero07, Pee07].

Another task level is defined when the idea of locating the occurrences of the most often repeating part is extended to retrieve all parts that are repeated [P6] and [Pee07]. Finding all repeated parts provides already a fairly good description since the remaining non-repeated segments between the repeated parts can be be also returned in the result. This leads quite naturally to the task definition where the structure of the entire piece has to be described. The description now consists of a segmentation of the piece and a grouping of the segments that are occurrences of the same musical part. The groups can then be lettered “A”, “B”, “C” and so forth in the order of their first occurrence. Since some of the musical parts have distinct “roles” in the Western pop music, some methods aim to assign the groups with labels, such as “verse” and “chorus”.

Proposed Methods

From the methods proposed in this thesis [P6] aims to recover a description of the parts that have more than one occurrence, while [P8] aims to produce a description of all the parts in the piece. Publication [P7] proposes a method for labelling structure descriptions with musically meaningful labels, such as “verse” and “chorus”.

3.2 Features and Data Representations

Since the acoustic signal sample values are relatively uninformative by themselves, some feature extraction has to be employed. The question is, once again, what are the important aspects of the signal that humans observe when determining the sectional form of a piece, and how these aspects could be formulated in a mathematical way. Bruderer et al. [Bru06] conducted experiments to find the perceptual cues that humans use to determine a segmentation point in music. The results suggest that “global structure”, “change in timbre”, “change in level”, “repetition”, and “change in rhythm” indicated the presence of a structural boundary to the test subjects. The cue category “global structure” is not necessarily well-defined compared to the other cues as it is an umbrella category for the general perception of song structure itself.

3.2.1 Frame Blocking for Feature Extraction

The feature extraction in audio content analysis is normally done in relatively short, 10–100 ms frames. In music structure analysis where each point in the piece is often compared with every other point, a very short frame causes computational resource problems (if the feature is extracted from 20 ms frames with 50% overlap from a 3-minute piece, there will be 18000 frames and $1.62 \cdot 10^8$ frame pairs). For this reason many of the proposed methods employ a considerably larger frame length in the order of 100–600 ms. Not only does this reduce the amount of data, but it also allows focusing on a musically more meaningful time scale.

The idea of musically meaningful time scale has been taken even further in some methods that propose the use of event-synchronised feature extraction. In other words, instead of a fixed frame length and hop, the division is defined by the temporal locations of sound events [Jeh05] or the occurrences of a metrical pulse, e.g., tatum or beat [Mad04, Shi05, Lev06, Mar06, Che09], [P6]. Using an event-synchronised frame division has two benefits compared to the use of fixed frame length: tempo-invariance and sharper feature differences. The tempo-invariance property allows to adjust the frame rate according to the input signal tempo variations, e.g., with decreasing tempo, the frame length and hop will increase accordingly. The event-synchronised frame blocking also allocates consecutive sound events to different frames. In the case of acoustically differring events, this produces clearer difference between the features extracted from the two frames. The event-synchronisation can be obtained also by calculating the feature from the short frames and then averaging the values over the length of the event-synchronised frames [Ero07], [S4], and [P8].

Proposed Methods

The methods proposed [P6] utilises feature extraction frames defined by the beat pulse. Publication [P8] uses a fixed 92.9 ms frame length with 50% overlap, but the feature values are averaged over beat-synchronised frames.

3.2.2 Features

Because the general timbre of the music seems to be important for structure perception, features parametrising it are often employed in structure analysis methods. Perceptually timbre is closely related with the recognition of sound sources, and depends on the relative levels of the sound at critical bands and on the temporal evolution of these. Timbral features reflect the instrumentation of the piece, and should therefore enable discriminating between parts that differ in that respect, e.g., quite often verse and chorus. Majority of the structure analysis methods that utilise timbral information use MFCCs to describe it (see Sec. 2.3.1). Maddage [Mad06] used MFCC-like idea, but replaced the mel-scale filter bank with 4-12 triangular filters in each octave. Other parametrisations omit the DCT step and use some non-mel spacing in band definitions, e.g., the MPEG-7 *AudioSpectrumEnvelope* descriptor [Int02] has been used [Wel03, Lev08], or very similar constant-Q spectrograms, e.g., [Abd06, Cas06]. Aucouturier and Sandler [Auc01] compared different parametrisations of timbral information in music structure analysis, and found MFCCs to outperform the other features, such as linear prediction coefficients. MFCCs calculated from an example piece are illustrated in the top panels of Fig. 3.4.

Another important aspect of music is the pitched content. It forms harmonic and melodic sequences that the listener will notice and use to deduce the structure. A chromagram[3] is frequently used to describe the harmonic content. Chroma relies on the cyclic perception of pitch in human auditory system causing pitches an octave apart, i.e., half or double the frequency, to have a similar harmonic role. In the discretised version of chroma, the spectral energy is first assigned to bins with one semitone width (also half-semitone bins have been employed, e.g., [Cha05, Mar06]), and then the octave-equivalence classes are folded together. The process is illustrated in Fig. 3.5 where the magnitude spectrum is assigned to semitone bins, illustrated by the piano

[3] Chromagram refers to a spectrogram-like representation with the feature calculated from several frames and stored as a matrix in which each column corresponds to a time frame. The feature itself is referred as chroma, or pitch-class profile.

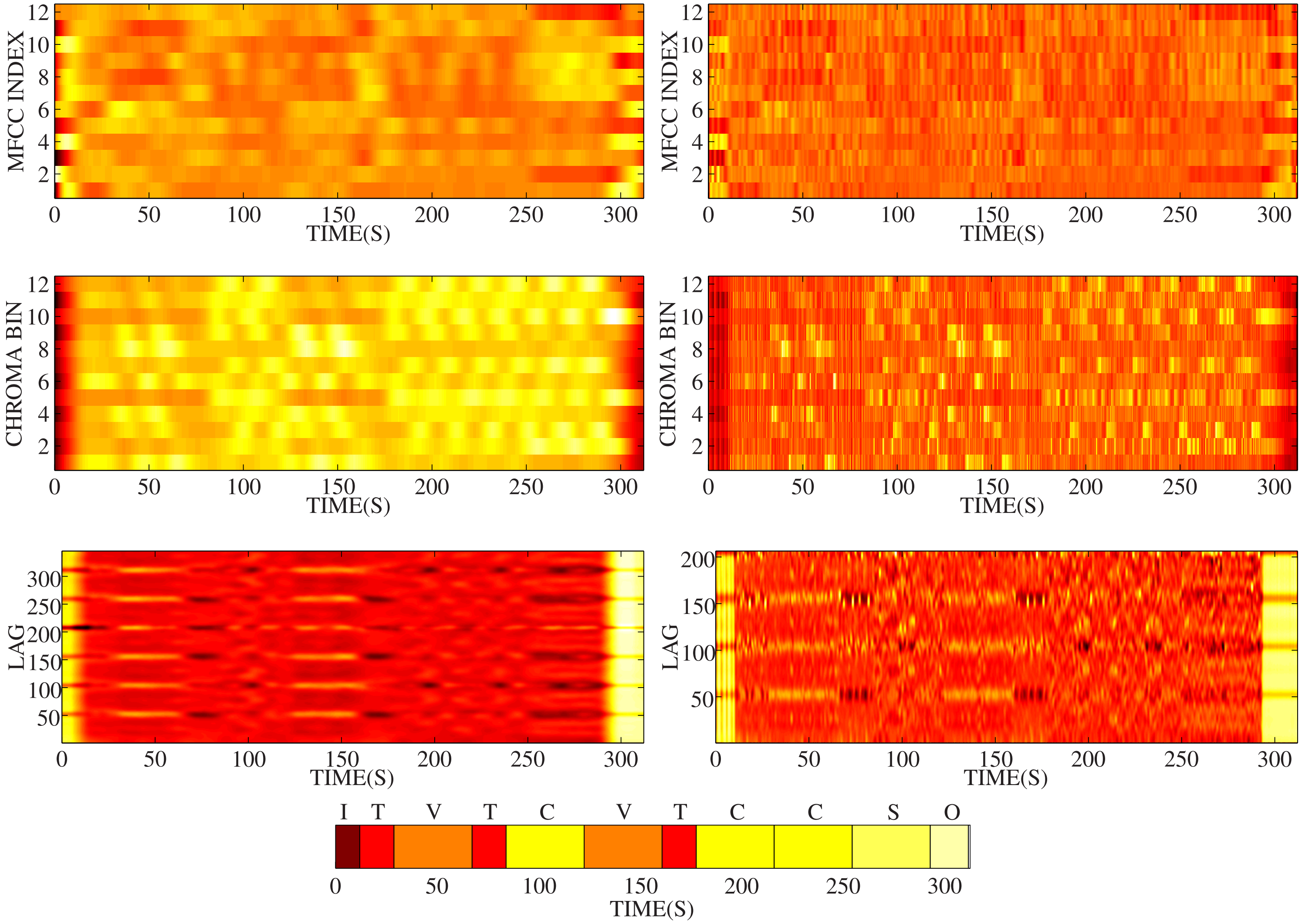


Figure 3.4: Feature matrices with MFCCs (top), chroma (middle), and rhythmogram (bottom), calculated with feature manipulation focusing the time-scale to short (right) and long (left) time scales. The annotated structure of the piece is given in the bottom panel, and the parts are indicated with: intro (I), theme (T), verse (V), chorus (C), solo (S), and outro (O). The example piece is “Tuonelan koivut” by Kotiteollisuus.

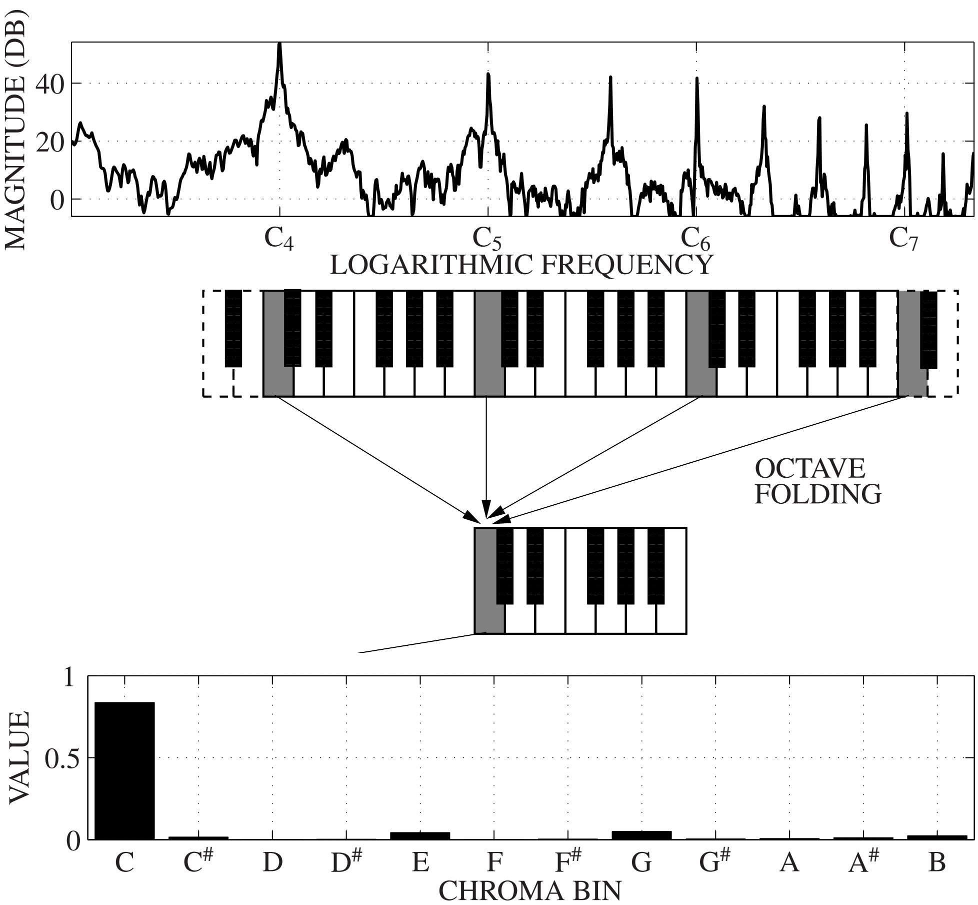


Figure 3.5: Illustration of the chroma calculation with octave folding from an input signal containing s a single piano note C_4, Top panel contains the magnitude spectrum, and the remaining panels illustrate how the energy corresponding to all occurrences of chroma “C” are gathered through octave folding to the single chroma vector bin.

keys. The resulting feature vector is illustrated in the bottom of the figure after normalising the values to sum to unity.

Several methods for calculating chroma have been proposed. The most straightforward one is to calculate the magnitude or power spectrum with discrete Fourier transform (DFT), resample the frequency axis to correspond to the semitone scale, and finally sum over the octave-equivalence classes. The spectral energy resampling can be done by assigning each DFT bin to only one semitone bin, as proposed by Bartsch and Wakefield [Bar05], or with smooth weighting functions, as proposed by Goto [Got03].

Some alternatives for the DFT analysis have been proposed. The method by Ryynänen and Klapuri [Ryy08b] replaces the DFT analysis with a multipitch estimation front-end, and was adopted to music structure analysis by Paulus and Klapuri [P8]. The method estimates the saliences of different fundamental frequencies in the range 80–640 Hz. The linear frequency scale is then transformed into a musical one by selecting the maximum salience value in each semitone bin. Finally, the octave folding

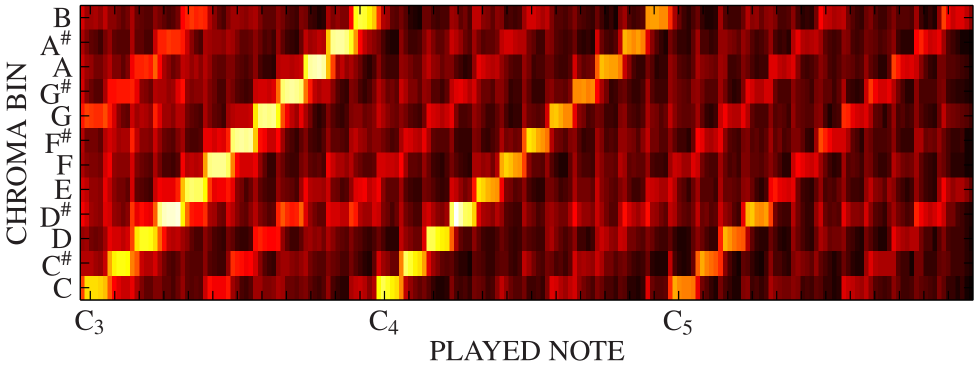


Figure 3.6: An example of a chroma feature matrix calculated from an input signal of an ascending scale of three octaves from C_3 to B_5 played with a MIDI piano. The brighter the image element, the larger the value in the chroma matrix.

is done as with the other chroma calculation methods. An example of a chromagram calculated in this way from an ascending three-octave scale is illustrated in Fig. 3.6. Müller and Kurth [Mül07b] proposed using a bank of time-domain band-pass filters corresponding to the semitone bands to replace the DFT analysis and frequency scale resampling.

Other chroma-like features are compared in a music structure analysis application by Ong et al. in [Ong06b]. Recently, Müller et al. [Mül09] proposed a method to increase the timbre-robustness of chroma. The method applies a linear operation on the semitone band energies in each frame to discard some information correlating with timbral properties of the signal. Some timbre-robustness is achieved also by the fundamental frequency salience estimation front-end in the method from [Ryy08b] through spectral whitening. Some more advanced chroma and pitch based features including also local energy information are described in the book by Müller [Mül07a, Chapter 3]. Chromagrams for an example piece are illustrated in the middle panels of Fig. 3.4.

In addition to the timbral and harmonic content parametrisations, the third aspect relating to the findings of Bruderer et al. is the rhythmic content. There has been very little effort in creating a rhythmic feature for music structure analysis. In addition to the simple loudness curve parametrisation proposed by Jehan [Jeh05], the *rhythmogram*, originally proposed by Jensen [Jen04], has possibly been the only effort in this direction. Rhythmogram is closely related to musical meter or beat analysis as it aims to find periodicities in an onset accentuation signal. The original publication proposes to use perceptual spectral flux (PSF) as the accentuation signal. PSF is defined as

$$h(t) = \sum_{k=1}^{K} w(k) \left(X(k,t)^{1/3} - X(k,t-1)^{1/3} \right), \tag{3.1}$$

where $\boldsymbol{w}$ is a perceptually motivated frequency weighting accounting for the frequency-dependent sensitivity of hearing, k frequency index, $\boldsymbol{X}$ magnitude spectrogram, and t

is a time index. Next, autocorrelation of the PSF is calculated by

$$A_{\mathrm{CORR}}(i,t) = \sum_{i=0}^{a-1} h(t)h(t+i) \tag{3.2}$$

in sliding windows of several second in length and the correlation values for lags up to 2 s are retained. Peaks in this function indicate periodicities with period equal to the lag. Paulus and Klapuri noted in [S4] that the use of the rhythmogram information in addition to timbral and harmonic features provides useful information to structure analysis. They replace the PSF front-end with a musical accent signal which is a by-product of musical meter analysis method [Kla06b] used for creating an event-synchronised frame division. Rhythmograms calculated from an example piece are illustrated in the lowest panels of Fig. 3.4.

The "dynamic features" proposed by Peeters [Pee04a] are also aiming to parametrise the rhythmic content by describing the temporal evolution of features in frames up to 10 s in length. The specific feature proposed in [Pee04a] consists of calculating short-time Fourier transform (STFT) of the energy envelopes of mel-scale filter bank and using the result as the feature representation for one frame. This feature resembles the TRAPS feature discussed earlier in Sec. 2.3.1.

In addition to the timbral, harmonic, and rhythmic features discussed above, also other more "traditional" features, such as the ones discussed in Sec. 2.3.1, have been utilised in some music structure analysis methods, e.g., by Ong [Ong06a] and Xu et al. [Xu04]. As a related multimodal method, Zhu et al. [Zhu05] propose to use the image representation of the piece lyrics on a karaoke videos to distinguish between chorus and verse after locating them based on the acoustic information. The differentiation is relatively simple: choruses have similar melodic content and similar lyrics, while the verses have similar melodic content, but differring lyrics.

Even though different features describe different musical properties, to date very few methods have utilised more than one feature at a time (except the methods with a large number of more simple features combined with feature vector concatenation, e.g., [Xu04, Ong06a]). Eronen [Ero07] proposed to combine MFCCs and chroma features in the self-distance matrix (SDM) domain (see Sec. 3.2.3), similar to Peeters [Pee07] who combined SDMs from MFCCs, chroma, and spectral contrast features. Levy et al. [Lev07] compare combining the information from timbral and harmony related features either by feature vector concatenation, or after initial processing of timbral features and chord recognition from harmonic features against using the individual features. The obtained results suggest that feature combination improves performance. Similar approach was later adopted by Cheng at al. [Che09]. Paulus and Klapuri [P8] combine the information obtained from MFCCs, chroma, and rhythmogram in probability domain, as will be described in Sec. 3.3.4.

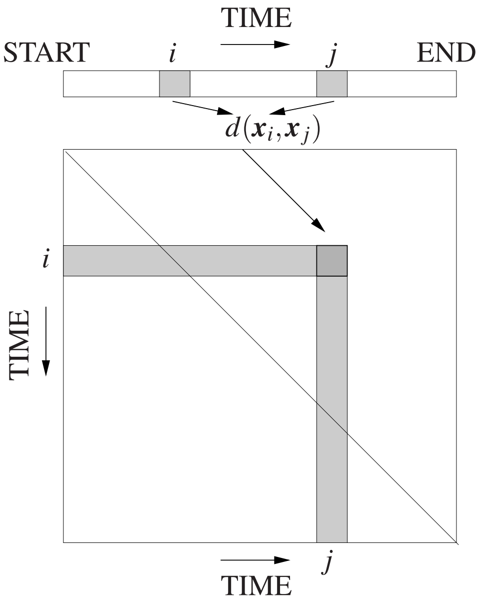


Figure 3.7: Self-distance matrix formation. The distance between feature vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is calculated and set to the SDM element $D(i,j)$. After [Coo03].

Proposed Methods

The methods proposed in this thesis utilise several different acoustic features. In [P6] the initial piece segmentation is done using 13 MFCCs and their time derivatives, and the segment matching is done with chroma from three two-octave ranges to produce a 36-dimensional feature vector. The supplemental publication [S4] evaluates the suitability of MFCCs, chroma, and rhythmogram features and their combinations in structure analysis. Additionally, different temporal resolutions are evaluated to focus the feature at different time scales. The method proposed in [P8] uses MFCCs and chroma with two different temporal scales, and a rhythmogram calculated in relatively long frames.

3.2.3 Self-distance Matrix

As the musical structure is implied strongly by repetition, different parts of the input piece have to be compared with other parts to locate repetitions. A frequently used mid-level representation, self-distance matrix is obtained by calculating the distance $d(\boldsymbol{x}_i,\boldsymbol{x}_j)$ between all frame pairs $i,j = \{1,2,\ldots,N_\mathrm{F}\}$ and storing the result in a matrix form.[4] As illustrated in Fig. 3.7, the element $D(i,j)$ of the SDM denotes the distance between the two frames. The origins of the matrix representation can be considered to stem from the recurrence plots proposed by Eckmann et al. [Eck87] for the analysis

[4] The dual of SDMs are self-similarity matrices in which each element describes the *similarity* between the frames instead of distance. Most of the following operations can be done with either representation, although here we discuss only SDMs.

of chaotic systems. In the context of music structure analysis, the SDMs were initially proposed by Foote [Foo99] for music visualisation.

Different distance and similarity measures have been proposed for the calculation of SDMs, the two most often used being the Euclidean distance

$$d_{\mathrm{E}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \| \boldsymbol{x}_i - \boldsymbol{x}_j \|, \tag{3.3}$$

and the cosine distance

$$d_{\mathrm{C}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0.5 \left(1 - \frac{\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle}{\| \boldsymbol{x}_i \| \| \boldsymbol{x}_j \|} \right), \tag{3.4}$$

in which $\| \cdot \|$ denotes the vector norm and $\langle \cdot, \cdot \rangle$ dot product. The main difference between the two distances is that the Euclidean distance is affected by the scaling of the feature values whereas the cosine distance depends only on the directional difference of the vectors. Additionally, the range of Euclidean distance is not limited while the cosine distance is always in the range $[0, 1]$.

The distance measures (3.3) and (3.4) are defined to compare single frames. In case the feature is parametrising a property occurring as sequences, such as chroma feature describing melodic or chord progressions, it is sometimes justified to include the local temporal evolution of the feature in the distance measure. Foote [Foo99] proposed to utilise a method resembling the sliding windows discussed earlier in Sec. 2.4.2 by averaging the distance values from $N + 1$ consecutive frames and using that as the distance value instead, i.e.,

$$\hat{d}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{N+1} \sum_{n=-N/2}^{N/2} d(\boldsymbol{x}_{i+n}, \boldsymbol{x}_{j+n}). \tag{3.5}$$

Similar approach was later adopted by Müller and Kurth [Mül07b].

The idea of combining information from several consecutive frames to one SDM element has been extended from the sliding window discussed above. Shiu et al. [Shi05] calculate the average distance from the feature vectors within segments of one musical measure in length, but instead of sliding the window, the consecutive windows do not overlap. This way the information of low-level features is combined to an SDM on the level of musical measures. This idea was extended by Paulus and Klapuri [P6] by calculating the match between the feature vector sequences of two musical measure with dynamic time warping (DTW), allowing more flexibility in the matching (for a description of DTW, see Appendix A). The idea of hierarchical SDMs was taken to the extreme by Jehan [Jeh05], where SDMs of multiple levels of temporal hierarchy are calculated starting from individual frames to musical patterns. Each higher level in the hierarchy was calculated based on the SDM of the finer temporal structure.

Recurring patterns in the feature vector sequence $\boldsymbol{x}_i, i = \{1, 2, \ldots, N_{\mathrm{F}}\}$ will form patterns in the SDM. If the distance measure used is symmetric, i.e., $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = d(\boldsymbol{x}_j, \boldsymbol{x}_i)$, the resulting SDM will also be symmetric around the main diagonal. The two

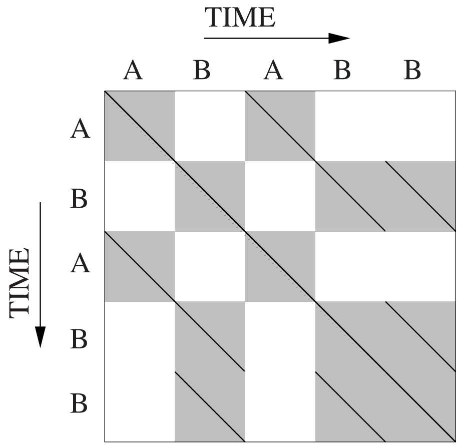


Figure 3.8: An example of the patterns formed in SDMs. The sequence consists of two parts, A and B, repeating as indicated, and darker element denotes lower distance.

most important patterns that will be formed in an SDM from the patterns of the features are illustrated in an idealised SDM in Fig. 3.8. If the features measure something like the instrumentation that stays somewhat constant over the duration of a musical part, *blocks* of low distance will be formed. Such a feature could be, e.g., MFCCs. In case the feature describes a property that does not remain constant over the part occurrence, but instead evolves forming a sequence, diagonal *stripes* of low distance will be formed. An example of a such feature is chroma. Locating and interpreting these patterns with various methods is the main approach employed in many of the structure analysis methods in the literature.

As Peeters [Pee04a] noted, the features alone do not determine whether blocks or stripes are formed, but the temporal parameters of the feature extraction process are also important. In other words, the longer the temporal window is that the feature vector describes, the more likely it is that blocks will be formed in the SDM. This was demonstrated in [Pee04a] by varying the length of the dynamic feature calculation window: with relatively short windows stripes were formed in SDM, but as the window length was increased, the more the stripes were surrounded by blocks. Similar observation was confirmed with MFCCs, chroma, and rhythmogram by Paulus and Klapuri [S4]. They low-pass filtered the MFCCs and chroma features over time with various cut-off frequencies before the SDM calculation, and varied the autocorrelation window length in the rhythmogram calculation. The aim was to select such time scale parameters that the distance measures for block- and stripe-likeness in the SDM was optimal, i.e., prominent stripes when two parts are true occurrences of the same part as well as prominent blocks. Similar smoothing filter was proposed by Müller and Kurth for a chroma-based feature in [Mül07b]. The effect of the time scale parameter on the feature values extracted from an example piece is illustrated in Fig. 3.4, and the SDMs resulting from the presented feature matrices are illustrated in Fig. 3.10.

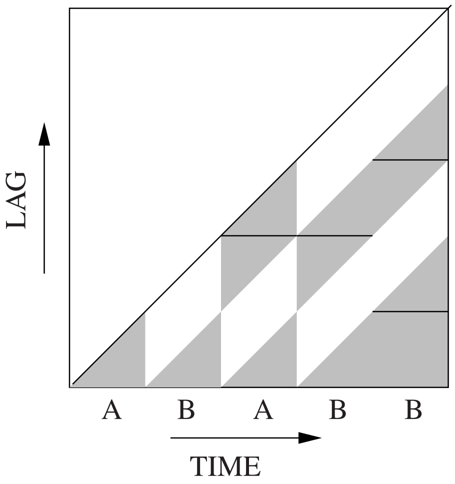


Figure 3.9: Time-lag matrix representation of the SDM from Fig. 3.8. The non-main diagonal stripes will be transformed into horizontal lines with the vertical position describing the interval (lag) between the occurrences.

Another way to present the information in an SDM is to transform it to a time-lag format [Got03]. In an SDM both the axes represent absolute time, but in time-lag matrix ***R*** the other axis is changed to represent time difference (lag) instead

$$R(i,j) = D(i,i-j), \text{if } i-j > 0. \tag{3.6}$$

The ordinate transformation discards the duplicate information of symmetric SDM. Fig. 3.9 represents the time-lag version of the information contained in the SDM of Fig. 3.8. The diagonal stripes formed by the repeated sequences appear as horizontal lines in the time-lag representation, and may be easier to extract. Even though time-lag representation transforms the stripe information into a more easily interpretable form, the block information is transformed into parallelograms and may now be more difficult to extract.

Proposed Methods

Among the methods proposed in this thesis [P6] uses an SDM calculated with cosine distance measure from beat-synchronised MFCCs for the initial segmentation. The segment matching uses an SDM on the level of musical measures. Each element in the SDM describes a DTW-based dissimilarity between the measure-length sequences of beat-synchronised chroma frames matched with cosine distance measure. The method proposed in [P8] uses beat-synchronised SDMs calculated with cosine distance measure.

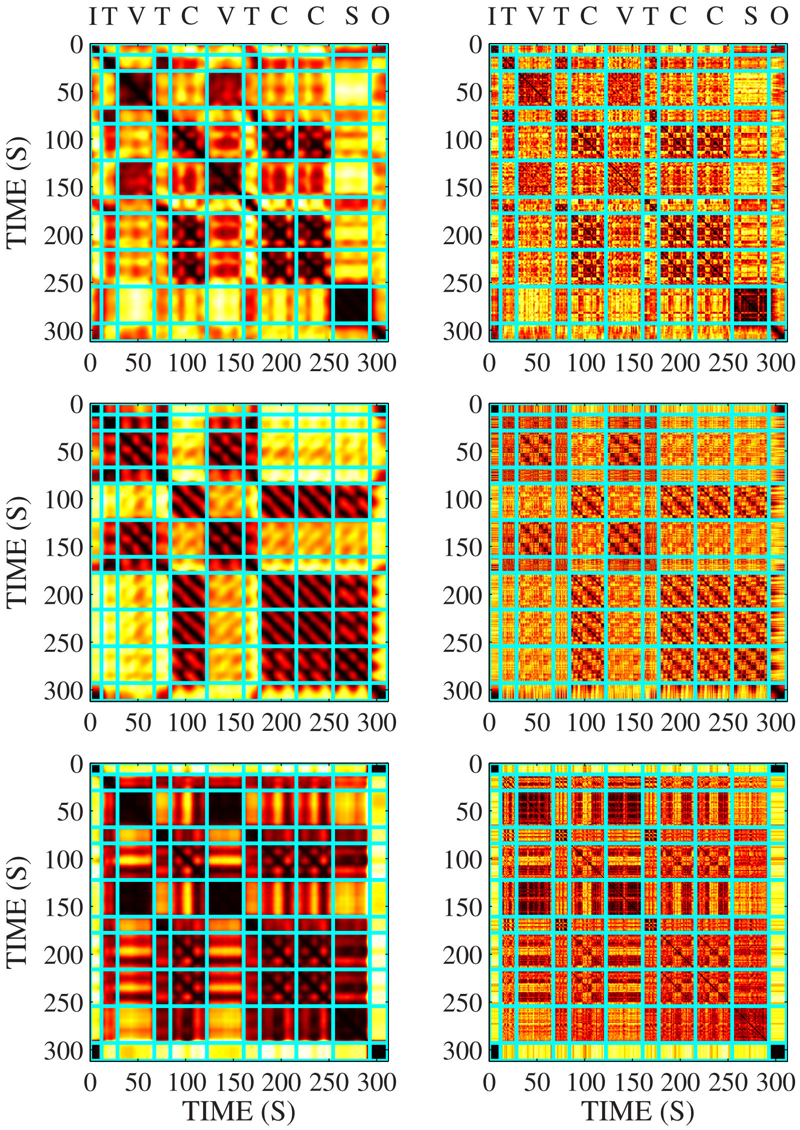


Figure 3.10: Example SDMs calculated using cosine distance (3.4) on the feature matrices shown in Fig. 3.4 (top row MFCCs, middle row chroma, bottom row rhythmogram, left column coarser time scale, right column finer time scale, darker pixel denotes lower distance). The annotated structure of the piece is indicated by the overlay grid, and the part labels are indicated in the top of the figure with: intro (I), theme (T), verse (V), chorus (C), solo (S), and outro (O). The figure shows how different parts share some of the perceptual aspects, but not all, e.g., chorus and solo have similar harmonic but differring timbral content.

3.3 Approaches

Various methods have been proposed for music structure analysis. The main methods are listed in Table 3.1, and Fig. 3.11 illustrates an overview block diagram containing some of the operational entities the proposed methods employ. This section will

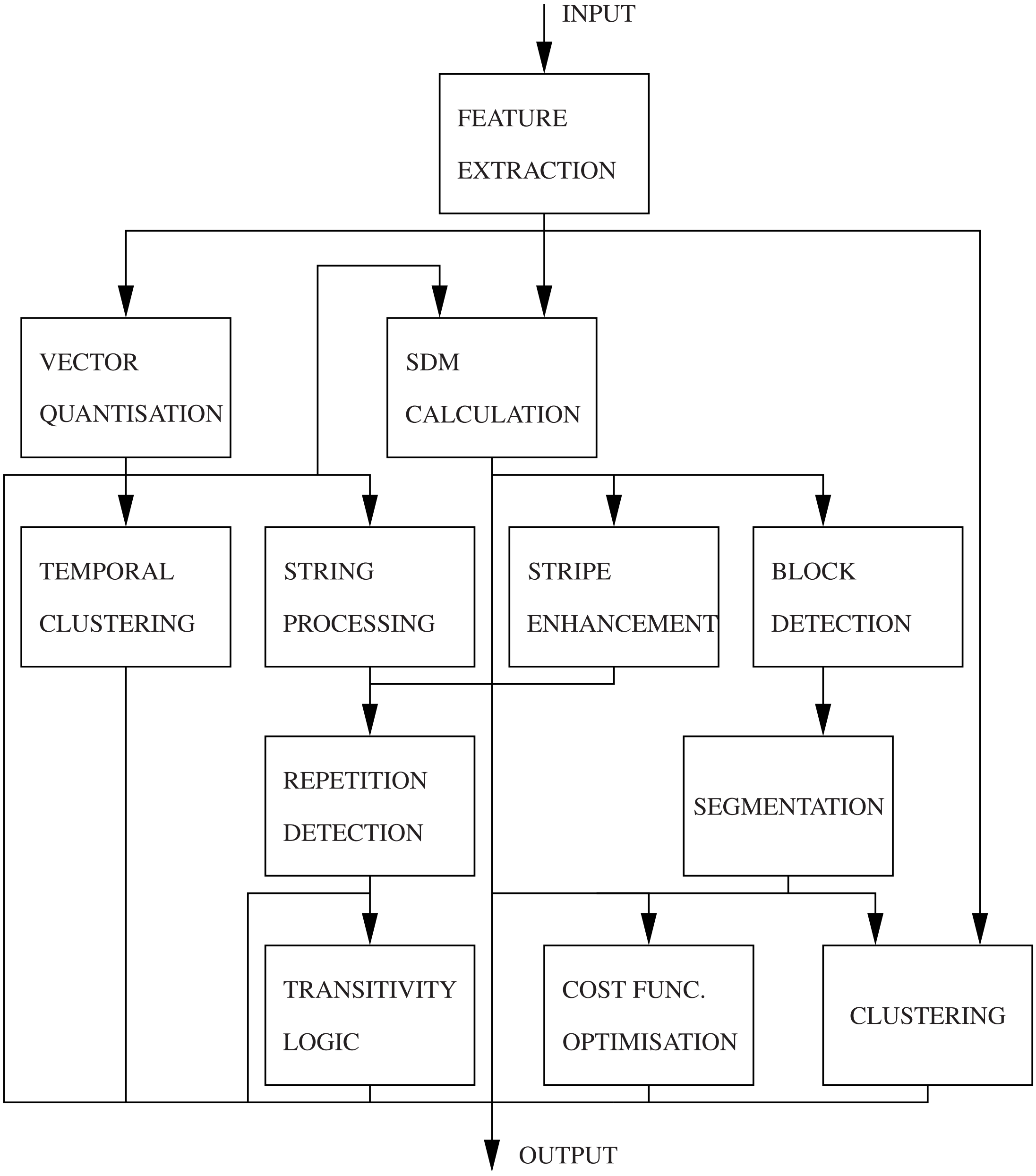


Figure 3.11: An overview block diagram of various operational entities employed in music structure analysis methods.

describe the main approaches in more detail, and the interconnections between the operations should become apparent.

A categorisation of music structure analysis methods proposed by Peeters [Pee04a] divides them to *sequence* and *state* approaches. The sequence approaches assume that the musical signal contains sequences of events that are repeated in the same order later in the piece, thus forming the diagonal stripes in the SDM. Because the sequence

Table 3.1: A summary of discussed methods for music structure analysis.

Author / publication	Task	Acoustic features	Approach	Method
Aucouturier et al. [Auc05]	full structure	spectral envelope	state	HMM
Barrington et al. [Bar09]	full structure	MFCC / chroma	state	dynamic texture model
Bartsch & Wakefield [Bar05]	thumbnailing	chroma	sequence	stripe detection
Chai [Cha05]	full structure	chroma	sequence	stripe detection
Cooper & Foote [Coo03]	summarisation	magnitude spectrum	state	segment clustering
Dannenberg & Hu [Dan02]	repetitions	chroma	sequence	dynamic programming
Eronen [Ero07]	chorus detection	MFCC+chroma	sequence	stripe detection
Foote [Foo99]	visualisation	MFCC		self-similarity matrix
Foote [Foo00]	segmentation	MFCC		novelty vector
Goto [Got03]	repetitions	chroma	sequence	stripe detection (RefraiD)
Jehan [Jeh05]	pattern learning	MFCC+chroma+loudness	state	hierarchical SDMs
Jensen [Jen07]	segmentation	MFCC+chroma+rhythmogram	state	diagonal blocks
Levy & Sandler [Lev08]	full structure	MPEG-7 timbre descriptor	state	temporal clustering
Logan & Chu [Log00]	key phrase	MFCC	state	HMM / clustering
Lu et al. [Lu04]	thumbnailing	constant-Q spectrum	sequence	stripe detection
Maddage [Mad06]	full structure	chroma	state	rule-based reasoning
Marolt [Mar06]	thumbnailing	chroma	sequence	RefraiD
Müller & Kurth [Mül07b]	multiple repetitions	chroma statistics	sequence	stripe search & clustering
Ong [Ong06a]	full structure	multiple	sequence	RefraiD
Paulus & Klapuri [P6]	repeated parts	MFCC+chroma	cost func.	
Paulus & Klapuri [P8]	full description	MFCC+chroma+rhythmogram	cost func.	
Peeters [Pee04a]	full structure	dynamic features	state	HMM, image filtering
Peeters [Pee07]	repeated parts	MFCC+chroma+spec. contrast	sequence	stripe detection
Rhodes & Casey [Rho07]	hierarchical structure	timbral features	sequence	string matching
Shiu et al. [Shi05]	full structure	chroma	state	state model stripe detection
Turnbull et al. [Tur07]	segmentation	various		various
Wellhausen & Höynck [Wel03]	thumbnailing	MPEG-7 timbre descriptor	sequence	stripe detection

methods locate only the repeats, it is more than likely that the analysis result will not cover the entire piece. The state approaches in turn consider the piece to be produced by a (finite) state machine, where each state produces some part of the signal. Considering the SDM representation, the state approaches can be thought to describe the formed blocks. Some instantiations of the two categories will described in more detail in Secs. 3.3.2 and 3.3.3 after discussing methods for segmentation only.

3.3.1 Segmentation

Even though only segmenting the input audio is not directly in the scope of this thesis, it is a necessary pre-processing step in the methods proposed in the included publications [P6] and [P8]. Tzanetakis and Cook [Tza99] propose to segment a signal by extracting a set of features from the signal and calculating Mahalanobis distance

$$d_{\mathrm{M}}(\boldsymbol{x}_i,\boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathsf{T}}\Sigma^{-1}(\boldsymbol{x}_i - \boldsymbol{x}_j) \tag{3.7}$$

between successive frames $j = i+1$. Σ is the covariance matrix of the features, calculated over the whole signal. Large differences in the distance values indicate possible segmentation points. Foote [Foo00] utilises the self-similarity matrix representation calculated from MFCC features with cosine distance, and detected segment boundaries with a 2D corner point detection matrix. Similar method is used to create candidate segmentation points in [P6] and [P8], and therefore it will now be described in more detail.

After calculating the SDM, an $m \times m$ corner point detection kernel matrix $\boldsymbol{K}$ is correlated along the main diagonal to locate corners of blocks of low distance. The kernel matrix has a 2×2 checkerboard -like structure

$$\boldsymbol{K} = \begin{pmatrix} \boldsymbol{Q}_{\mathrm{TL}} & \boldsymbol{Q}_{\mathrm{TR}} \\ \boldsymbol{Q}_{\mathrm{BL}} & \boldsymbol{Q}_{\mathrm{BR}} \end{pmatrix}, \tag{3.8}$$

where the following symmetries hold

$$\boldsymbol{Q}_{\mathrm{TL}} = -\boldsymbol{J}\boldsymbol{Q}_{\mathrm{BL}} = -\boldsymbol{Q}_{\mathrm{TR}}\boldsymbol{J} = \boldsymbol{J}\boldsymbol{Q}_{\mathrm{BR}}\boldsymbol{J} \tag{3.9}$$

and $\boldsymbol{J}$ is an $(m/2) \times (m/2)$ matrix with ones on the main anti-diagonal and zeros elsewhere. It reverses the order of matrix rows when applied from left and the order of columns when applied from right. The values in $\boldsymbol{Q}_{\mathrm{BR}}$ can be simply all -1. However, for the detection of a corner point the values close to the centre of the kernel should be more important than the values further. To accomplish this, Gaussian radial weighting has been applied to define the kernel element values by

$$Q_{\mathrm{BR}}(i,j) = -\exp\left(-\frac{r^2}{2\sigma^2}\right), \tag{3.10}$$

where the radius r is defined by

$$r^2 = \frac{4}{m^2}\left((i-1)^2 + (j-1)^2\right), \tag{3.11}$$

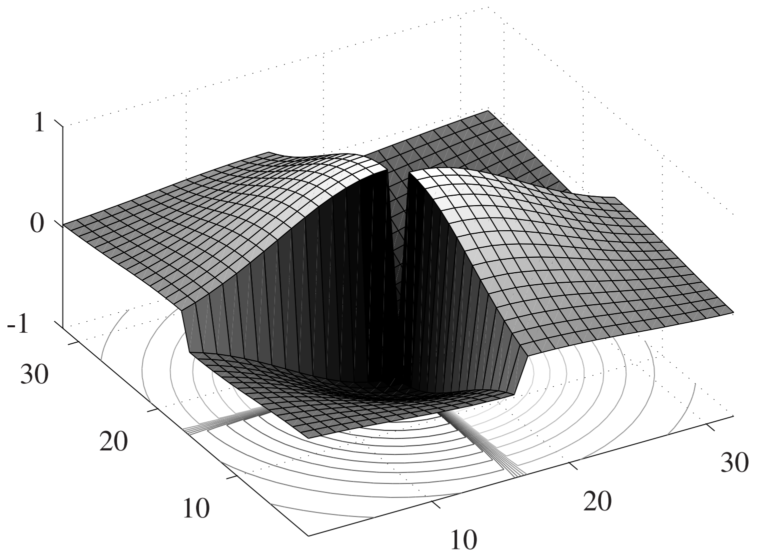


Figure 3.12: An example corner point detection kernel with Gaussian radial smoothing ($m = 32$, $\sigma = 0.5$.)

and σ is a shape parameter. An example detection kernel matrix used to create segmentation points in [P8] is illustrated in Fig. 3.12.

The result of the correlation of SDM $\boldsymbol{D}$ with the detection kernel matrix $\boldsymbol{K}$ is called the novelty vector and is calculated with

$$n(k) = \sum_{i=-m/2}^{m/2-1} \sum_{j=-m/2}^{m/2-1} K(\frac{m}{2}+i, \frac{m}{2}+j)D(k+i, k+j). \tag{3.12}$$

As the name suggests, the novelty value has peaks wherever there are corner points in the input SDM, thus indicating the start of new segment. An example SDM with the resulting novelty vector is illustrated in Fig. 3.13. The segmentation can then be done by locating a desired number of the peaks, possibly with some minimum distance and height constraints.

Jensen adopted a completely different approach for locating the main diagonal blocks from the SDM in [Jen07]. The method uses a cost function to determine the "optimal" segmentation of the signal. The cost function consists of a fixed cost of adding a new segmentation point, and a signal-dependent term for each candidate segment. A segment in the time axis of the signal defines a submatrix, or a block, on the main diagonal of SDM. The internal segment gain is defined as approximately the average similarity within the defined block. Now, adding a new segment will have the fixed addition cost and the similarity term operates as a counterweight. The resulting optimisation problem is formulated as searching the shortest path in a edge-weighted directed acyclic graph which can be solved, e.g., with the Dijkstra's algorithm [Cor90, Chapter 25].

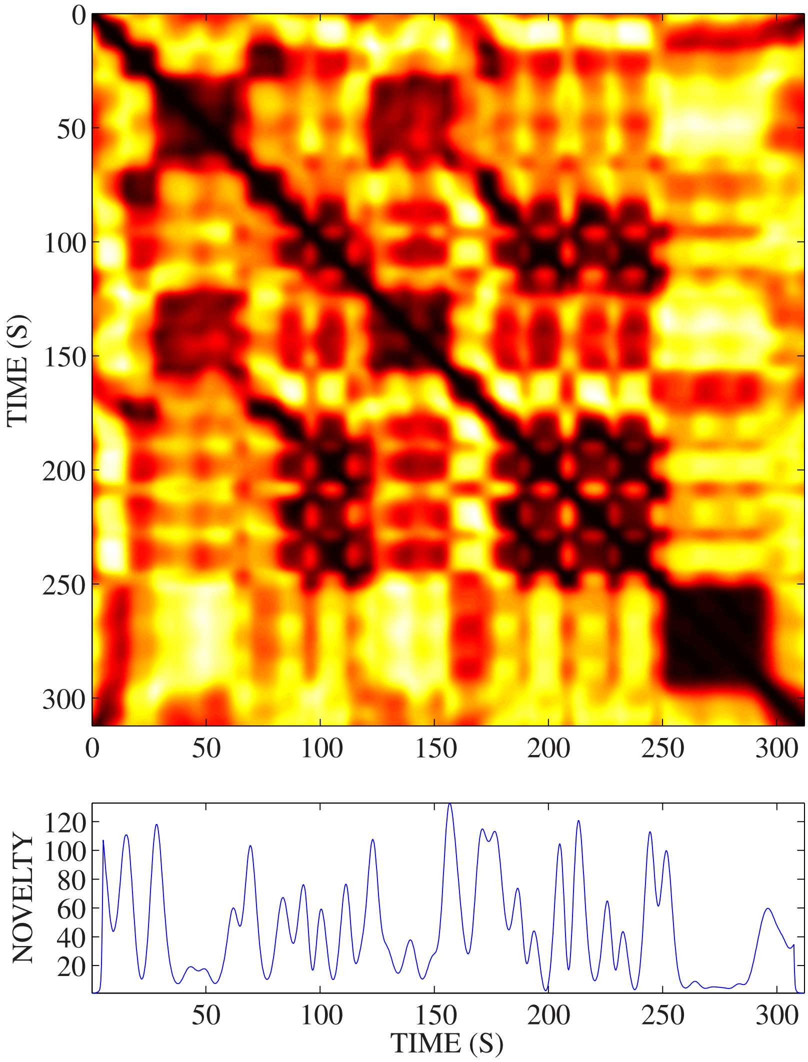


Figure 3.13: The block MFCC SDM from Fig. 3.10 and the novelty function resulting from applying the kernel from Fig. 3.12. A peak in the novelty vector indicates that there is a corner point in the SDM, and it may be a segment border location.

For other methods to music segmentation, an interested reader is referred to the publication by Turnbull et al. [Tur07] in which several acoustic features and both supervised and unsupervised segmentation methods are evaluated.

Proposed Methods

The methods proposed in this thesis utilise the novelty function based segmentation generation to produce a set of candidate segmentation points from which the final segmentation points are then selected. In [P6] the novelty function is calculated from

MFCCs, while in [P8] it is calculated from all the three features used, and summed to produce the final detection function.

3.3.2 State Approach

A direct continuation of the segmentation described earlier is to analyse the content of the created segments and to cluster them. This is an example of a *state* approach which assumes that the underlying generative process is a state machine, and each state produces distinct observations, i.e., feature vectors. A method with the segmentation and a following clustering stage was proposed by Cooper and Foote [Coo03]. A second similarity matrix is calculated now using the similarities between the found segments. The content of a segment s_i is parametrised by a multivariate normal distribution $\mathcal{N}_i$ with a mean vector μ_i of length N and a covariance matrix Σ_i. The similarity between two segments s_i and s_j is defined with

$$d_{\text{SEGS}}(s_i, s_j) = \exp\left(-d_{\text{KL}}(\mathcal{N}_i || \mathcal{N}_j) - d_{\text{KL}}(\mathcal{N}_j || \mathcal{N}_i)\right), \tag{3.13}$$

where $d_{\text{KL}}(\cdot||\cdot)$ is the Kullback-Leibler divergence between two multivariate normal distributions [Gol03]

$$d_{\text{KL}}(\mathcal{N}_i || \mathcal{N}_j) = \frac{1}{2}\left(\log\frac{|\Sigma_j|}{|\Sigma_i|} + \text{trace}(\Sigma_j^{-1}\Sigma_i) + (\mu_i - \mu_j)^{\mathsf{T}}\Sigma_j^{-1}(\mu_i - \mu_j) - N\right). \tag{3.14}$$

Above $|\cdot|$ denotes the determinant of a matrix. Having the similarities for all segment pairs, the segments are grouped with spectral clustering [Wei99] through singular value decomposition. Logan and Chu [Log00] used similar Gaussian parametrisation on fixed-length segments (1 s) that were then clustered with agglomerative hierarchical clustering[5] until the segment dissimilarity exceeds a set threshold.

The method proposed by Goodwin and Laroche [Goo04] performs the segmentation and clustering at the same time. The method itself resembles the cost function based segmentation by Jensen [Jen07] discussed earlier, with the difference that the searched path can now return to a state defined earlier if it is globally more efficient for the structure description.

The concept of "state" is taken more explicitly in methods employing HMMs for the analysis, e.g., the methods proposed by Logan and Chu [Log00], Aucouturier and Sandler [Auc01], or Gao et al. [Gao03]. The idealised assumption is that each musical part can be represented by a state in an HMM, and the states produce observations from the underlying probability distribution. The methods employ a fully connected HMM topology with Gaussian mixture models for the observation densities. A fully connected HMM with four states is illustrated in Fig. 3.14. The analysis operates by training the HMM with the piece to be analysed, and then decoding the same signal

[5] In agglomerative hierarchical clustering, each data point is initially a separate cluster. At each iteration, the two closest clusters are merged, and this is repeated until some stopping condition is met.

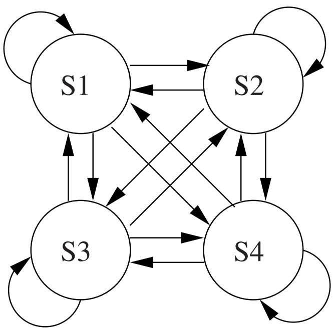


Figure 3.14: Topology of a 4-state fully connected HMM.

with the model. Effectively this implements vector quantisation of the feature vectors with some temporal dependency modelling included in the state transition probabilities. Though this model has certain appeal, it does not work very well in practice because the result is often temporally fragmented, as noted by Peeters et al. [Pee02]. The fragmentation is due to the fact that the individual states start to model individual sound events instead of longer musical parts.

Because of the temporal fragmentation of the analysis result obtained with HMM approach, several post-processing methods have been proposed to alleviate the problem. Many of these utilise a large number of states, e.g., 40–80, in the HMM analysis, and use the resulting state sequence as a mid-level representation for further analysis. As the number of states is large, each of them now represents a certain short sound event and the musical patterns will form patterns in the state sequence. Fig. 3.15 shows the resulting state sequences of an example piece after analysing it with fully connected HMMs with 8 and 40 states.

The state sequence representation is included also for general audio parametrisation in the MPEG-7 standard as the *SoundModelStatePathType* descriptor [Int02]. Abdallah et al. [Abd05] proposed to calculate histograms of the states with a sliding window over the entire sequence, and use the resulting histogram vectors as new feature vectors for each frame. The histogram calculation allows considering the local context of a frame, i.e., similar musical segments will be parametrised using approximately the same set of states and the histogram reveals the similarity of the set contents. For the further utilisation of the state histograms Abdallah et al. [Abd05] proposed a probabilistic clustering, and they later extended the method to include statistical modelling of the cluster durations in [Abd06]. Levy et al. [Lev06] increased the amount of knowledge of the context with a variant of fuzzy c-means clustering applied on the histograms. This approach was formalised further by Levy and Sandler in a probabilistic framework [Lev08].

Slightly different approach to reduce the resulting fragmentation was proposed by Peeters [Pee02]. After calculating SDM and creating a segmentation, the mean feature vector values are used to provide initial cluster centroids. These centroids are then updated in k-means clustering of individual frames. The outcome of the clustering may

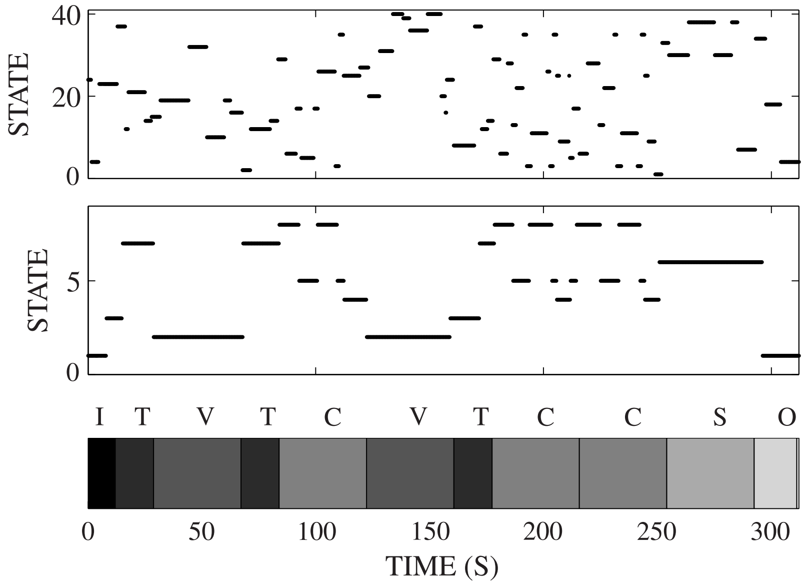


Figure 3.15: Example state sequences resulting from HMM based analysis. The feature data used is the long time-scale MFCCs presented in Fig. 3.4. The top panel presents the state sequence after using an 40-state fully connected HMM, while the middle panel presents the result from a 8-state HMM, and the lowest panel illustrates the annotated structure. For the part names, see Fig. 3.4.

be fragmented, and this fragmentation is addressed by using the cluster centroids as an initialisation of HMM training. The results suggest that employing approximately correct number of clusters and the way the HMM training is initialised from the cluster centroids of temporally co-located data improves the result compared to direct HMM training.

In a recent publication Barrington et al. [Bar09] propose to use dynamic texture mixture models (DTM) for the structure analysis. DTM is basically a state model, where each (hidden) state produces observations that have a temporal structure. The mixture term controls how the states can be clustered to form distinctive state sequences. The main novelty of the method compared to the HMM-based state methods is that the observation model itself takes into account the temporal behaviour of the produced observations and no heuristic post-processing is necessary.

3.3.3 Sequence Approach

The methods proposed to utilise the HMM state sequence as their input have attempted to locate regions of similar state distribution regardless of the order of the states. From the point of view of the audio this corresponds to the situation where the occurrences of a musical part are played with similar sound events, but the order of the sounds does not matter. Considering the perception of repeated structures, the order of the events is also important. The *sequence* approaches aim to utilise the repeated sequences to provide a description of the musical piece structure.

The search for diagonal stripes from the SDM is a common theme in many of the sequence approaches. Because the stripes are often easily detected by a human, several more or less heuristic image processing methods have been proposed to locate them. A smoothing effect resembling the one obtained by determining the SDM values as a mean of consecutive framewise distances (see (3.5)) can be obtained by low-pass filtering the SDM along the diagonals [Wel03, Bar05]. Marolt [Mar06] proposed to enhance the stripes by calculating multiple SDMs with different sliding window lengths and then combining them with elementwise multiplication. Lu et al. [Lu04] employed multiple iterations of erosion and dilation[6] filtering along the diagonals to enhance the stripes by filling small breaks and removing too short line segments. Ong [Ong06a] extended the erosion and dilation filtering into two-dimensional filter to enhance the entire SDM. Goto [Got03] employed a two-dimensional local filter to enhance the stripes; similar enhancement was later utilised by Eronen [Ero07]. Peeters [Pee07] proposed to low-pass filter along the diagonal direction, and high-pass filter along the anti-diagonal direction to enhance the stripes.

After enhancing the stripe prominence, the stripe segments can be found, e.g., by thresholding. The *RefraiD* approach originally proposed by Goto [Got03] has later been employed in several methods. It uses the time-lag version of SDM to select the lags that are more likely to contain repeats, and then detect the line segments along the horizontal direction of the lags. Each of the found stripes specifies two occurrences of a sequence: the original one and a repeat. For chorus detection, or simple one-clip thumbnailing, selecting a sequence that has been repeated most often has proven to be an effective approach. In the case that more comprehensive structural description is wanted, multiple stripes have to be detected and some logical reasoning to deduce the underlying structure, as proposed by Dannenberg [Dan02].

Similar to the dynamic programming approaches used for segmentation [Goo04, Jen07], some of the stripes can be found by a path search. Shiu et al. [Shi06] interpret the self-similarity values as probabilities, and define a local transition cost to prefer diagonal movement. Then Viterbi search is employed to locate the optimal path through the lower (or upper) triangle of SDM. The stripes have large similarity values, thus the probability values are also large, and the path is likely to go through the stripe locations. Another method to locate stripe segments by growing them in a greedy manner was proposed by Müller and Kurth [Mül07b].

As with the state approaches, HMMs have also been employed in sequence approaches. Rhodes and Casey [Rho07] employed a string matching method to the HMM state sequence representation to create a hierarchical description of the structure. Though the algorithm was presented to operate on a finite alphabet formed by the HMM states, the authors suggest that similar operations could be accomplished with

[6] Erosion filter replaces the element value with the minimum in the filtering range, and dilation with the maximum.

feature vectors after modifying the matching algorithm to accept vector inputs. Aucouturier and Sandler [Auc02] proposed another method for inspecting the HMM state sequences with image processing methods. The main idea is to calculate a binary co-occurrence matrix (resembling an SDM) based on the state sequence, i.e., the element has value 1, if the two frames have the same state assignment, and 0 otherwise. Then a diagonal smoothing kernel is applied on the matrix to smooth out small mismatches between sequences. Finally, stripes are searched from the resulting matrix with Hough transform, which is claimed to be relatively robust against bad or missing data points.

3.3.4 Cost Function Based Approach

Most of the music structure analysis methods described above rely on rather straightforward approaches in the analysis, such as "locate blocks of low distance on SDM main diagonal" or "locate stripes from SDM and deduce the repeated structure". These ground the actual analysis on the observations without too much consideration on the motivation behind them. An alternative approach to these is to first define what are the properties of a good structural description, formulate the properties in terms of the observations, and finally optimise the resulting fitness function with respect to the given acoustic input.

Paulus and Klapuri [P6] proposed to define a cost function for a piece structure description in more abstract terms (experiments on a similar approach were later reported by Peiszer [Pei07]). The method aims to locate repeated parts from a piece, and defines a cost for a structural description with the following three main terms.

- *Within-group dissimilarity*: The segments that are claimed to be occurrences of the same musical part, i.e., repeats of each others, should be acoustically similar.
- *Amount unexplained*: It is desirable that the provided structural description would cover as much of the piece as possible. (The method [P6] is intended to handle only repeated parts).
- *Complexity*: If the found description covers the entire piece and each found group has a small within-group dissimilarity, but the description consists of tens or hundreds of segments and groups (similar to the HMM vector quantisation methods with a large number of states), the result is practically useless for the end-user. Thus a term penalising complexity is added.

When the terms are combined with some weights, an abstract version of the overall cost is obtained

$$C = \text{dissimilarity} + w_{\mathrm{U}}\ \text{unexplained} + w_{\mathrm{C}}\ \text{complexity}, \tag{3.15}$$

where w_{U} and w_{C} are the relative weights for the amount left unexplained and description complexity, respectively.[7]

[7] It is also possible to motivate this kind of cost function through the minimum description length principle [Dud01, Chapter 9].

To be able to evaluate the cost function (3.15) for some structural description E, more formal definitions have to be made. The notation used in the original publication [P6] is here changed to match the one used in [P8]. All the possible segmentations of the input piece form a set $\mathbb{S}$. If the segment border locations are not restricted in any way, the size of the set may be very large, see (B.1) in Appendix B. A subset $\mathbb{S}' \subset \mathbb{S}$ of the set with S segments $s_i \in \mathbb{S}'$ that do not overlap temporally form a part of the description E of the piece. In the method for finding only repeated parts [P6], the selected subset does not need to cover the entire duration of the piece, but this is required in the method [P8] discussed later. Each segment s_i has a length $L(s_i)$ associated to it. The grouping of segments is defined with a function $g(s_i)$, so that two segments belong to the same group, i.e., are occurrences of the same musical parts, if the function returns the same value for both of them $g(s_i) = g(s_j)$. Now the structural description can be seen as a combination of a set of segments and a grouping function $E = (\mathbb{S}', g)$. The segments with the same group assignment can also be defined as a set

$$\mathbb{G}_n = \{s_i | g(s_i) = n, s_i \in \mathbb{S}'\}, \tag{3.16}$$

and thus a description is a union of such groups $E = \bigcup_{n=1}^{N} \mathbb{G}_n$, where N is the number of unique groups in the description.

Using the definitions above, the cost function (3.15) can now be written as

$$C(E) = \sum_{n=1}^{N} d_{\mathrm{G}}(\mathbb{G}_n) \sum_{s_i \in \mathbb{G}_n} L(s_i) + w_{\mathrm{U}} \left(1 - \frac{\sum_{n=1}^{N} \sum_{s_i \in \mathbb{G}_n} L(s_i)}{\Lambda} \right) + w_{\mathrm{C}} \log(1+N), \tag{3.17}$$

where Λ is the length of the piece, and $d_{\mathrm{G}}(\mathbb{G}_n)$ is the within-group dissimilarity of the segments assigned to group $\mathbb{G}_n$. The optimal description E_{OPT} is the one minimising the cost

$$E_{\mathrm{OPT}} = \underset{\mathrm{E}}{\operatorname{argmin}}\{C(E)\}. \tag{3.18}$$

The effect of the three different terms and their weights is illustrated in Fig. 3.16. A detailed definition of the within-group dissimilarity $d_{\mathrm{G}}(\mathbb{G}_n)$ and the features used in its calculation, as well as a description of an algorithm for optimising (3.18) are provided in [P6].

The main weakness in the cost function based method described above, as well as with most of the other methods relying on locating individual stripes or blocks in the SDM, is that they operate only on parts of the SDM. In other words, when locating stripes, each of the stripes is basically handled separately without any contextual information focusing only on the distance values on the stripe itself and on its close vicinity. Considering data clustering problem, each of the formed clusters should be compact, i.e., have small intra-cluster distances, and the clusters should be well-separated, i.e., have large inter-cluster distances. These same principles apply relatively well also in music structure analysis: all occurrences of a part should be similar to each others and differ from occurrences of other parts.

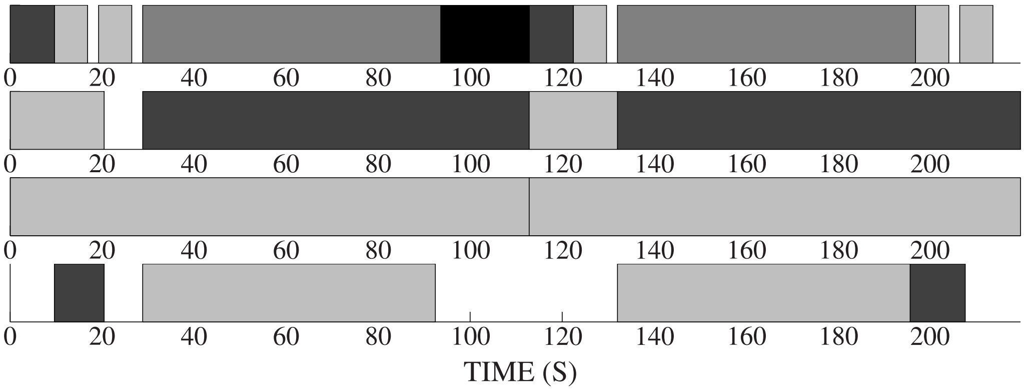


Figure 3.16: Example of the effect of the relative weights in the cost function (3.15). The annotated ground truth structure is illustrated in the top panel. The second panel is the analysis result with some reasonable values for the weights. Increasing the complexity term weight produces the result in the third panel, and decreasing the weight of unexplained parts produces the result in the lowest panel.

Paulus and Klapuri [P8] propose a fitness measure for structural descriptions formalising the "small intra- & large inter-cluster distance" idea for musical parts. The temporal fragmentation occurring in frame-by-frame approaches is alleviated by calculating the distances from segments of several frames, similar to the earlier approach [P6]. Additionally, instead of using plain distance between segments, the distances are interpreted as probabilities $\hat{p}\left(g(s_i) = g(s_j)\right)$ that the segments s_i and s_j belong to the same group. The overall fitness of a description E is defined as

$$P(E) = \sum_{s_i \in \mathbb{S}'} \sum_{s_j \in \mathbb{S}'} W\left(s_i, s_j\right) l(s_i, s_j, g), \tag{3.19}$$

where

$$l(s_i, s_j, g) = \begin{cases} \log\left(\hat{p}\left(g(s_i) = g(s_j)\right)\right), & \text{if } g(s_i) = g(s_j) \\ \log\left(1 - \hat{p}\left(g(s_i) = g(s_j)\right)\right), & \text{if } g(s_i) \neq g(s_j) \end{cases}. \tag{3.20}$$

The weighting factor $W\left(s_i, s_j\right)$ is defined as the number of elements in (or the area of) the submatrix $\boldsymbol{D}_{[i,j]}$ defined by the two segments

$$W\left(s_i, s_j\right) = L(s_i) L(s_j). \tag{3.21}$$

In other words, the overall fitness measure is a weighted sum of log-probabilities assigned to each of the submatrices defined by the piece segmentation. This is illustrated in Fig. 3.17, where the tinted submatrices denote segment pairs from which the in-group probability is considered, and the white submatrices the segment pairs with the probability that the segments belong to different groups. The problem now is to locate the description E_{OPT} maximising the overall fitness

$$E_{\text{OPT}} = \underset{\text{E}}{\text{argmax}} \{P(E)\}. \tag{3.22}$$

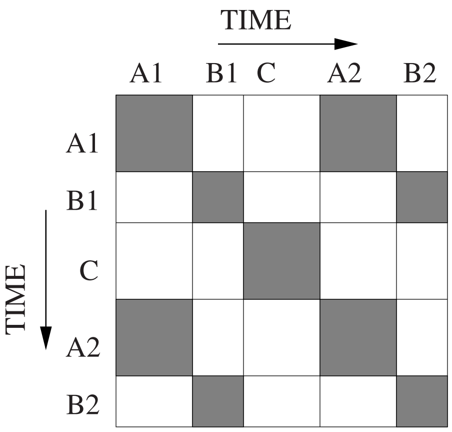


Figure 3.17: An illustration of the idea of the clustering fitness measure motivated structure description fitness measure. The description claims that the structure is “A,B,C,A,B”, i.e., “A1” and “A2” belong to same group as well as “B1” and “B2”. It would be desirable to have low distance values in the tinted submatrices, and large distances in the others.

The resulting optimisation task requires the probabilities for two segments to belong to the same group $\hat{p}\left(g(s_i) = g(s_j)\right)$ be defined. The method proposed in [P8] aims to address several of the perceptual cues [Bru06] simultaneously by calculating the pairwise distances $d\left(s_i, s_j\right)$ between two segments s_i and s_j with three different acoustic features: MFCCs, chroma, and rhythmogram, and defining two complementary distance measures: *block* and *stripe* distances. The use of multiple features is intended to address the different aspects of music: MFCCs for timbre, chroma for harmonic content, and rhythmograms for rhythmic content. The distance measures aim to parametrise the patterns that are formed in SDM (as illustrated in Fig. 3.8) with parametrisations whose main ideas are illustrated in Fig. 3.18. The details of calculating the distance values $d\left(s_i, s_j\right)$ between segments and a method for calculating the probability $\hat{p}\left(g(s_i) = g(s_j)\right)$ of the two segments to belong to the same group are given in the publication [P8].

The problem of optimising (3.22) is combinatorial problem with a very large number of possible solutions (see Appendix B for a discussion on this) making an exhaustive search practically impossible to be applied. Publication [P8] formulates the problem as a search in a directed acyclic graph with varying path costs, and proposes an algorithm for the search.

Proposed Methods

The methods proposed in this thesis rely on defining a cost or a fitness function to structural descriptions. In [P6] the task is to locate the description minimising the cost function (3.17), while in [P8] the task is to locate the description maximising the fitness function (3.19).

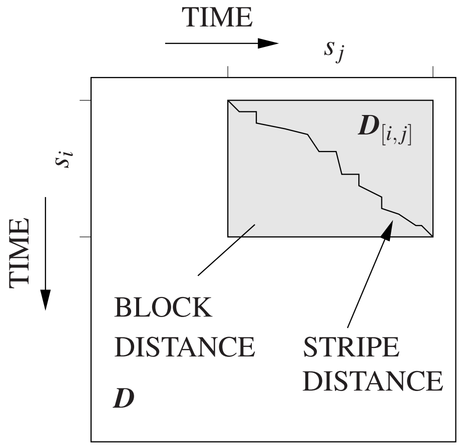


Figure 3.18: Illustration of the basic ideas behind the *stripe* and *block* distances between two segments s_i and s_j of a piece. The stripe distance is based on the path of least cost through the submatrix $\boldsymbol{D}_{[i,j]}$ while the block distance is based on the average distance value within the submatrix.

3.4 Musicological Modelling

Middleton states in [Mid99]: "Genres are defined, in part, by conventions governing the musical process: what can be, what cannot be, what is usually, done; of course; this includes conventions relating to form." This suggest that there should be some stereotypical structures in musical pieces from the same genre. Even though it is often claimed in a more or less serious way that majority of Western popular music follows the same sectional form over and over again, the "knowledge" of this form has not been applied widely in music structure analysis systems. Maddage [Mad06] assumes that the song structure is one of a pre-stored ones, and fits the acoustic observations to the models. Shiu et al. [Shi05] assume that the musical parts in the analysed piece follow the state model illustrated in Fig. 3.19. The transition probabilities between the states were set manually and the stripe-like information in an SDM was used to provide observations for the states. The final structural description was then found by applying Viterbi search on the resulting optimisation problem. Another system utilising statistical knowledge of the sequential dependencies between musical parts was proposed by Paulus and Klapuri [P8].

Most of the methods that attempt to provide a full description of the structure of a piece provide only the time stamps defining the segmentation and a grouping information. Though this is valuable for any human end-user or further information retrieval systems, the knowledge of the "identity" of each part would be appreciated also, as indicated by the user study conducted by Boutard et al. [Bou06]. The identity here means the musical role of the part: chorus, verse, etc. Paulus and Klapuri [P7] propose modelling the sequential dependencies of musical part labels with N-grams. An example of a label bigram calculated from the *UPF Beatles* data set is illustrated in Fig. 3.20. Given a structural description with arbitrary tags as the input, the method

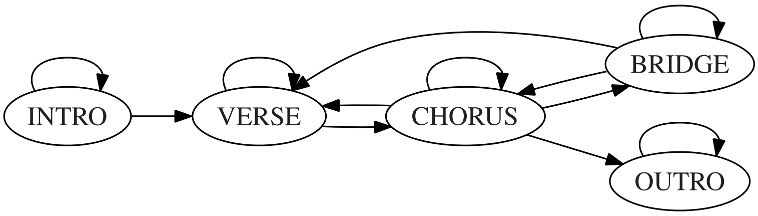


Figure 3.19: The musical part state model used by Shiu et al. [Shi05].

assigns a label to each of the group tags, e.g., input “p1, p2, p1, p3, p2” may be transformed into “verse, chorus, verse, bridge, chorus”. The method operates by evaluating the overall probability of different resulting label sequences over the description with

$$p(c_{1:J}) = \prod_{i=1}^{J} p(c_i | c_{(i-N+1):(i-1)}), \tag{3.23}$$

where $p(c_i|c_{(i-N+1):(i-1)})$ is the N-gram probability of observing the label c_i after the label sequence $c_{(i-N+1):(i-1)}$. An algorithm to solve the optimisation problem is presented in [P7].

If some additional information related to the piece lyrics is available, it can be utilised in the labelling, e.g., Zhu et al. [Zhu05] use lyrics to distinguish between verses and choruses, and Cheng et al. [Che09] label the parts after rule-based analysis of the lyrics. As a related topic, Mahedero et al. [Mah05] perform a full structure analysis based only on the lyrics.

Even though the N-gram model for musical part label sequences is very simple, it is able to assign musically meaningful labels to structural descriptions to some extent. Because of this Paulus and Klapuri [P8] propose to include a term modelling the label knowledge to the fitness function for structural descriptions (3.19), resulting into a new fitness function definition of

$$P(E) = \sum_{i=1}^{S}\sum_{j=1}^{S} W\left(s_i, s_j\right) l(s_i, s_j, g) + \frac{w_{\mathrm{L}} A}{S-1} \sum_{i=1}^{S} \log\left(p(M(g(s_i))|M(g(s_{1:(i-1)})))\right). \tag{3.24}$$

In the fitness function (3.24) above, $M(\cdot)$ denotes a mapping function from the arbitrary group identifiers to the musical part labels, and thus the added term only evaluates the resulting N-gram probability similar to (3.23). w_{L} denotes the relative weighting factor assigned for the labelling term, and

$$A = \sum_{i=1}^{S}\sum_{j=1}^{S} W\left(s_i, s_j\right). \tag{3.25}$$

The segment set $\mathbb{S}'$ is now ordered by the starting time of the segments to an ascending order $\mathbb{S}' = (s_1, s_2, \ldots, s_S)$ to enable the use of N-grams. The optimisation problem of maximising (3.24) can still be solved with the search algorithm presented in [P8].

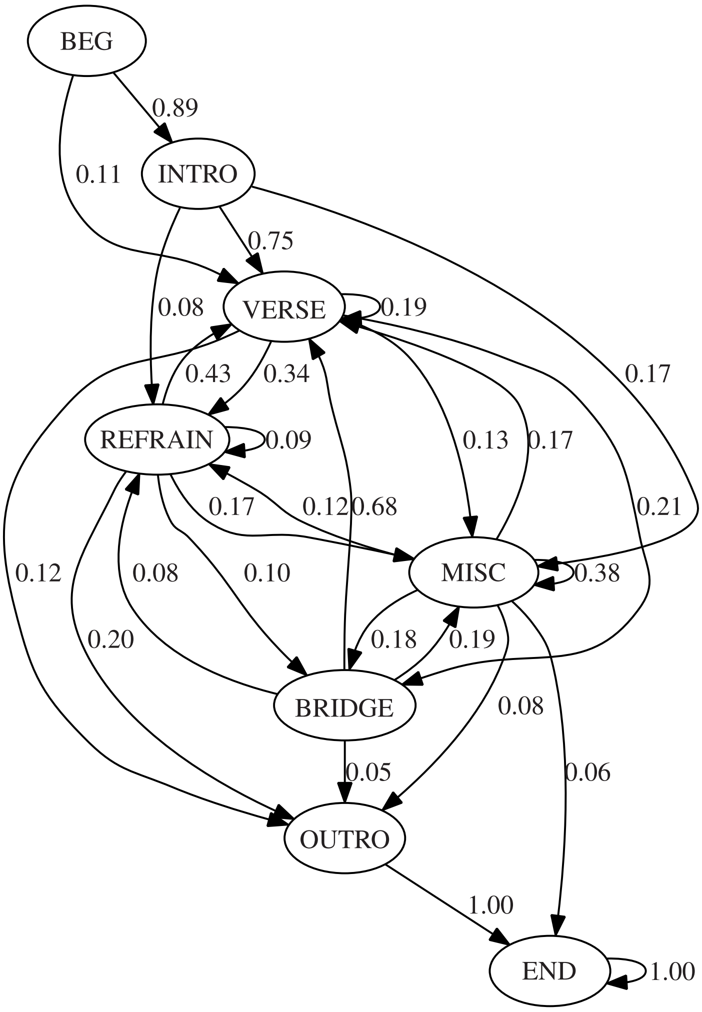


Figure 3.20: A bigram of the musical part transitions from the songs by The Beatles. The named parts contribute 85% of all the musical part occurrences in the songs, and the remaining ones are assigned the label "MISC". The edge weights denote the transition probabilities between nodes. Edges with less than 0.05 associated probability are removed for clarity. The nodes "BEG" and "END" are added to provide a unique beginning and ending point for a path through the graph.

However, it should be noted that the addition of the labelling information increases the computational complexity of the search considerably (see Appendix B), and the overall improvement in the system performance is relatively small. Because of this, it is advisable to adhere to a simpler fitness function definition, e.g., (3.19), and to assign the labels as a post-processing step instead.

Proposed Methods

From the methods proposed in this thesis [P7] uses an N-gram model to assign musically meaningful “role” labels to the groups in a structure description. The method in [P8] proposes to include the label assignment in the overall fitness function, so that the labelling would be done jointly with the structure description is searched.

3.5 Evaluating the Obtained Result

Evaluating the performance of a music piece structure analysis method is not as simple as it may initially seem. Some of the evaluation metrics proposed in the literature will now be briefly discussed.

A possible evaluation metric for segmentation matches the segmentation points in the annotation with the ones produced by the analysis accepting a small deviation. Recall rate, precision rate, and F-measure may be calculated from the matched points similar to the drum transcription evaluation on page 14. An alternative is to calculate the mean (or median) time between a claimed and annotated segmentation point and vice versa. [Tur07]

The evaluation of music thumbnailing is more difficult, since the quality of the output is measured mainly subjectively instead of an objective metric. For this reason, user studies are required to evaluate the result, as described by Chai [Cha05] and Ong [Ong06a].

Evaluating the result of a method producing a description of the full structure of a piece has proven to be less straightforward, as mentioned by Lukashevich [Luk08]. Many of the evaluation measures adopt an approach related to clustering result evaluation: pairs of frames are inspected if they belong to any occurrence of the same musical part, and if they do, they are considered to belong to the same cluster. Calculating an F-measure based on these statistics was proposed by Levy and Sandler [Lev08]. Another closely related metric is the Rand index [Hub85], that was used by Barrington et al. [Bar09].

Abdallah et al. [Abd05] proposed to match the segments in the analysis result and ground truth, and calculate a directional Hamming distance between frame sequences after the match. A similar approach with a differing background was proposed by Peeters [Pee07]. A second evaluation measure proposed by Abdallah et al. [Abd05] treats the structure descriptions as symbol sequences, and calculated the mutual information between the analysis result and ground truth. The mutual information concept was developed further by Lukashevich [Luk08] who proposed an over- and under-segmentation measures based on the conditional entropies of the sequential representations of structures.

A property that can be considered to be a weakness in the measures relying on pairs of frames, is that they disregard the order of the frames. In other words, they do not penalise hierarchical level differences between the found parts, e.g., after splitting an occurrence into two segments with the same group as the original one. Considering the importance of sequences for the perception of music, they should be noted in the evaluation. Chai [Cha05], and Paulus and Klapuri [P6] have proposed heuristics finding a common hierarchical level for the result and ground truth. However, the evaluation method is rather complicated and the results are still subject for discussion.

If the analysis method assigns the description with musically meaningful labels, it is possible to base the evaluation on them, as proposed by Paulus and Klapuri [P8]. The evaluation measure inspects the descriptions as a sequence of frames, each assigned with a label. Then a confusion matrix is calculated between the labels in the result and ground truth. It is possible to calculate further evaluation measures from the confusion matrix, e.g., the amount of the piece with correct label assignment.

Finally, it should be noted that any measure evaluates only with respect to the provided ground truth annotation. As the experiments by Bruderer et al. [Bru06] suggest, the perception of musical structures is not completely unambiguous. Thus the description provided by two persons on a same piece might be different. A small-scale comparison of descriptions made by two annotators was presented by Paulus and Klapuri [P8], and slight differences in the hierarchical levels as well as in the grouping were noted.

Proposed Methods

Among the publications included in this thesis, [P6] bases the evaluation on finding a common hierarchical level between the result and ground truth. The publication [P8] uses the pairwise F-measure calculated on analysis frames [Lev08], over- and user-segmentation measures [Luk08], and the label assignment measure proposed in the publication itself.

3.6 Summary

This chapter has provided an overview of the music structure analysis problem and various methods proposed for solving it. The definition of music structure analysis problem itself has varied from simple visualisation to segmentation, and further to providing a description for the full structure of the piece including labelling the found musical parts. Because different authors have targeted different aspects of the structure analysis problem, the employed methods vary considerably. This chapter has aimed to locate the methodological similarities between the methods and to organise them in a common context. The main methods proposed in the literature have been gathered to Table 3.1 with some notes on the targeted problem and employed method. Though several authors have proposed successful methods for solving some sub-tasks, such as chorus detection, there still is a need for a more general music structure analysis method. A publication included in this thesis [P8] has aimed for a more general approach, and the results seem promising.

Chapter 4

Conclusions and Future Work

THIS thesis has presented methods for analysing the content of music signal at two different time scales: transcription of drums, and analysis of the sectional form of a piece. Even though these two tasks are in quite opposite ends of the time scale, they share some signal representations and analysis methods.

4.1 Conclusions

Related to the drum transcription problem, this thesis has proposed various acoustic recognition methods, as well as the use of musicological modelling. Publication [P1] proposes recognising combinations of drums with a Gaussian mixture model (GMM) classifier. However, due to the large amount of different combination resulting from seven target drums, the recognition result was not very good. A considerable reduction in the error rate was obtained by including N-gram models describing the sequential and periodical dependencies between drum hits. The results suggest that the modelling of periodic dependencies produces more accurate predictions that the modelling of the directly preceding context.

As the findings with periodic N-grams showed, drums occur at relatively constant locations within rhythmic patterns. Publication [P2] proposes modelling the rhythmic roles of various drums as occurrence probabilities depending on the location within a pattern. The model was evaluated by using it to assign drum names to patterns performed with arbitrary sounds. The results suggest that the relatively simple model is able to predict correct labels for drums without acoustic cues. Analysing the results for various musical genres, large differences can be noted. This suggests that in some genres, e.g., “soft rock”, the drums are played in patterns with high consistency through the piece, while in others, e.g., “world”, the patterns are more complex and difficult to predict.

A different acoustic analysis approach is taken in [P3] where a supervised source separation method is proposed to separate each target drum from the input signal.

The method uses spectral templates for the target drums, and recovers time-varying gains for them. Since the goal was to transcribe the drums instead of acoustic source separation, a very coarse frequency resolution of only five bands was found to be adequate. Evaluations with acoustic recordings of drum sequences show that the proposed method has a near-perfect performance when the input signal consists mainly of the three target drums. The method is extended in [P4] from the single-channel input to multichannel signals available especially in recording studios. The evaluation results show that the use of multiple channels provides a performance increase over the single-channel analysis. In addition, the results show that the spectral analysis inherent in the source separation step of the method provides an improvement in performance compared to plain onset detection from close microphones. This is mainly due to the acoustic leakage between the microphone channels.

The "segment and classify" transcription methods often rely on onset detection for temporal segmentation of the signal. The onset detection is difficult in the case of polyphonic music as the input, and the result may contain spurious detections, or some drum onsets may be missed altogether. To avoid these problems, publication [P5] proposes to use a network of connected HMMs to perform the segmentation and recognition jointly. Detector-like modelling with separate "sound" and "silence" models for each target drum was found to outperform modelling the combinations of the target drums. The acoustic modelling uses "standard" speech recognition tools with MFCCs as features and GMMs to model the observation likelihoods. Feature dimensionality reduction with PCA and LDA was evaluated, and LDA was found to provide a considerable performance. Furthermore, unsupervised acoustic model adaptation with maximum likelihood linear regression was evaluated, but the obtained improvement was found to be relatively small. Despite the straightforward model, the method performs well even on polyphonic music.

Overall, the drum transcription performance from polyphonic music is still not perfect, but usable for some applications. With simpler inputs without other instruments and a limited set of drums, near-perfect transcription can be obtained. Even though drums occur in highly-predictable patterns, musicological models have not been able to provide considerable performance increase.

In addition to the drum transcription, this thesis has presented methods for analysing the sectional form of a music piece. The proposed analysis methods operate by defining a high-level fitness function specifying properties of a good structure description. The first method from [P6] proposes to include terms for musical part acoustic similarity, the amount of the piece not covered by the description, and the complexity of the description. Varying the relative weights of the terms allows the method to recover multiple descriptions for a single piece emphasising different aspects.

The structure analysis method proposed in [P8] takes a simpler approach in the fitness function definition: occurrences of a musical part should be similar and differ from occurrences of other parts. The main problem related to this is to define "similar". Three acoustic features: MFCCs, chroma, and rhythmogram are used to parametrise different perceptually important aspects of music signals. The necessity of the features was evaluated in [S4], and the results suggest that the features provide complementary information useful in the structure analysis. Even though the optimisation problem resulting from the definition of the fitness function is difficult, a search algorithm with intuitive control parameters is proposed for solving it. Despite the simple idea of the fitness function, the method is able to provide useful structure descriptions.

Almost all of the structure analysis methods describing entire pieces only indicate which segments are occurrences of the same musical part. That is, the segments are not provided with musically meaningful names. However, the musical parts may have certain roles in the piece, e.g., "intro", "chorus", or "verse". Publication [P7] proposes to model the sequential dependencies between these role labels with N-grams, and then use them in labelling the descriptions. The evaluation results show that labelling musical parts with this model is possible to some extent.

The obtained results suggest that using multiple features to parametrise different musical aspects produces better performance than using only single features. This is presumably because the perception of musical structures is a combination of several factors also in humans. Additionally, using higher-level information instead of simple low-level acoustic data appears to improve the analysis performance.

4.2 Future Work

Even though many aspects related to the drum transcription and music structure analysis have been addressed both in this thesis and in other publications, there still remains a large number of open questions for future work. One can even say that the methods developed to date are only the beginning. Most of the methods proposed so far rely heavily on the low-level acoustic processing and omit high-level modelling. A more general goal in the future work might be to include multiple information sources, especially high-level musical knowledge to the analysis process. Cross-utilisation of information between various music content analysis methods would be interesting, e.g., drum transcription result used to assist in music structure analysis, and musical structure to aid in drum transcription, cf. [Dan05].

An approach to be studied in drum transcription is to use more specific models in the recognition. The specificity could be achieved on several levels: localised models, more accurate generic models, and more accurate musicological modelling. The results of using localised models for transcription are promising [San04]. The idea is intuitive: when a person starts listening to a new song, only some generic mental models for different drums exist, but while listening the models are adapted to match the acoustic properties of the drums present in the piece. Nowadays the methods aim to transcribe everything with the same models. However, the drum sounds used in a jazz piece usually differ considerably from the ones in a death metal piece. The use of more accurate generic models refers to the idea of creating more specialised, e.g., genre-specific, global models. The same applies to musicological modelling: the played patterns differ with styles, and instead of generic models, more specific ones should be used.

The methods for music structure analysis are still relatively unsophisticated and most of them rely on various heuristics. Before investing more time on developing structure analysis methods, it would be best to create an accurate definition of the task. This would include more research on how humans perceive musical structures, the thesis of Bruderer [Bru08] is a good starting point for this. As with drum transcription, it is probable that using only low-level acoustic information does not produce the optimal result. Thus it would be interesting to develop different models for different musical styles based on the typical structures employed in them. Regarding the musical role labelling of the segments in a music structure description, the use of acoustic information should be considered in addition to the overly simple sequence model.

In general, research on algorithmic music content analysis has started to gain momentum in the last decade. As some of the methods are reaching the level of some maturity, i.e., they produce a result that actually can be used for something, more comprehensive music listening systems have become closer to reality.

Bibliography

[Abd05] S. Abdallah, K. Nolad, M. Sandler, M. Casey, and C. Rhodes, "Theory and evaluation of a Bayesian music structure extractor," in *Proc. of 6th International Conference on Music Information Retrieval*, London, England, UK, Sep. 2005, pp. 420–425.

[Abd06] S. Abdallah, M. Sandler, C. Rhodes, and M. Casey, "Using duration models to reduce fragmentation in audio segmentation," *Machine Learning*, vol. 65, no. 2–3, pp. 485–515, Dec. 2006.

[Auc01] J.-J. Aucouturier and M. Sandler, "Segmentation of musical signals using hidden Markov models," in *Proc. of 110th Audio Engineering Society Convention*, Amsterdam, The Netherlands, May 2001.

[Auc02] J.-J. Aucouturier and M. Sandler, "Finding repeating patterns in acoustic musical signals: Applications for audio thumbnailing," in *Proc. of Audio Engineering Society 22nd International Conference on Virtual, Synthetic and Entertainment Audio*, Espoo, Finland, 2002, pp. 412–421.

[Auc05] J.-J. Aucouturier, F. Pachet, and M. Sandler, "“The way it sounds”: Timbre models for analysis and retrieval of music signals," *IEEE Transactions on Multimedia*, vol. 7, no. 6, pp. 1028–1035, Dec. 2005.

[Bar05] M. A. Bartsch and G. H. Wakefield, "Audio thumbnailing of popular music using chroma-based representations," *IEEE Transactions on Multimedia*, vol. 7, no. 1, pp. 96–104, Feb. 2005.

[Bar09] L. Barrington, A. B. Chan, and G. Lanckriet, "Dynamic texture models of music," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Taipei, Taiwan, Apr. 2009, pp. 1589–1592.

[Bel05] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A tutorial on onset detection in music signals," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 1035–1047, Sep. 2005.

[Bel06] J. P. Bello, E. Ravelli, and M. B. Sandler, "Drum sound analysis for the manipulation of rhythm in drum loops," in *Proc. of IEEE International Confer-*

ence on Acoustics, Speech, and Signal Processing, Toulouse, France, May 2006, pp. 233–236.

[Bil93] J. A. Bilmes, “Timing is of the essence: Perceptual and computational techniques for representing, learning, and reproducing expressive timing in percussive rhythm,” Master’s thesis, Massachusetts Institute of Technology, Boston, Mass., USA, Sep. 1993.

[Bou06] G. Boutard, S. Goldszmidt, and G. Peeters, “Browsing inside a music track, the experimentation case study,” in *Proc. of 1st Workshop on Learning the Semantics of Audio Signals*, Athens, Greece, Dec. 2006, pp. 87–94.

[Bru06] M. J. Bruderer, M. McKinney, and A. Kohlrausch, “Structural boundary perception in popular music,” in *Proc. of 7th International Conference on Music Information Retrieval*, Victoria, B.C., Canada, Oct. 2006, pp. 198–201.

[Bru08] M. J. Bruderer, “Perception and modeling of segment boundaries in popular music,” Ph.D. dissertation, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, Apr. 2008.

[Bur98] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[Cas00] M. A. Casey and A. Westner, “Separation of mixed audio sources by independent subspace analysis,” in *Proc. of International Computer Music Conference*, Berlin, Germany, Aug. 2000, pp. 154–161.

[Cas06] M. Casey and M. Slaney, “The importance of sequences in musical similarity,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, May 2006, pp. 5–8.

[Cha05] W. Chai, “Automated analysis of musical structure,” Ph.D. dissertation, Massachusetts Institute of Technology, Boston, Mass., USA, Sep. 2005.

[Che09] H.-T. Cheng, Y.-H. Yang, Y.-C. Lin, and H. H. Chen, “Multimodal structure segmentation and analysis of music using audio and textual information,” in *Proc. of IEEE International Symposium on Circuits and Systems*, Taipei, Taiwan, May 2009, pp. 1677–1680.

[Chu91] K. W. Church and W. A. Gale, “A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English speech,” *Computer Speech and Language*, vol. 5, pp. 19–54, 1991.

[Cle66] C. W. Cleverdon, J. Mills, and M. Keen, “Aslib Cranfield research project - factors determining the performance of indexing systems, volume 1 - design, part 1 - text,” Cranfield University, Bedfordshire, UK, Tech. Rep., 1966.

[Com94] P. Comon, “Independent component analysis - a new concept?” *Signal Processing*, vol. 36, no. 3, pp. 287–314, Apr. 1994.

[Coo03] M. Cooper and J. Foote, “Summarizing popular music via structural similarity analysis,” in *Proc. of 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Platz, N.Y., USA, Oct. 2003, pp. 127–130.

[Cor90] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to algorithms*. Cambridge, Mass., USA: The MIT Press, 1990.

[Dan02] R. B. Dannenberg and N. Hu, “Pattern discovery techniques for music audio,” in *Proc. of 3rd International Conference on Music Information Retrieval*, Paris, France, Oct. 2002, pp. 63–70.

[Dan05] R. B. Dannenberg, “Toward automated holistic beat tracking, music analysis, and understanding,” in *Proc. of 6th International Conference on Music Information Retrieval*, London, England, UK, Sep. 2005, pp. 366–373.

[Dan08] R. B. Dannenberg and M. Goto, “Music structure analysis from acoustic signals,” in *Handbook of Signal Processing in Acoustics*, D. Havelock, S. Kuwano, and M. Vorländer, Eds. New York, N.Y., USA: Springer, 2008, vol. 1, pp. 305–331.

[Dav80] S. B. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug. 1980.

[dC06] A. de Cheveigné, “Pitch perception models,” in *Pitch*, C. J. Plack, A. J. Oxenham, R. R. Fay, and N. Popper, Arthur, Eds. Springer New York, 2006, pp. 169–233.

[Deg05] S. Degroeve, K. Tanghe, and B. De Baets, “A simulated annealing optimization of audio features for drum classification,” in *Proc. of 6th International Conference on Music Information Retrieval*, London, England, UK, Sep. 2005, pp. 482–487.

[Die02] T. G. Dietterich, “Machine learning for sequential data: A review,” in *Structural, Syntactic, and Statistical Pattern Recognition*, ser. Lecture Notes in Computer Science, T. Caelli, A. Amin, R. Duin, M. Kamel, and de Ridder D., Eds. Springer-Verlag, 2002, vol. 2396, pp. 15–30.

[Dit04] C. Dittmar and C. Uhle, “Further steps towards drum transcription of polyphonic music,” in *Proc. of 116th Audio Engineering Society Convention*, Berlin, Germany, May 2004.

[Dix07] S. Dixon, “Evaluation of the audio beat tracking system BeatRoot,” *Journal of New Music Research*, vol. 36, no. 8, pp. 39–50, 2007.

[Dow08] J. S. Downie, “The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research,” *Acoustical Science and Technology*, vol. 29, no. 4, pp. 247–255, 2008.

[Dud01] R. O. Duda, P. E. Hart, and G. Stork, David, *Pattern classification*, 2nd ed. New York, N.Y., USA: John Wiley & Sons, 2001.

[Eck87] J.-P. Eckmann, S. O. Kamphorst, and D. Ruelle, “Recurrence plots of dynamical systems,” *Europhysics Letters*, vol. 4, no. 9, pp. 973–977, Nov. 1987.

[Ell96] D. P. W. Ellis, “Prediction-driven computational auditory scene analysis,” Ph.D. dissertation, Massachusetts Institute of Technology, Boston, Mass., USA, Jun. 1996.

[Ell04] D. P. W. Ellis and J. Arroyo, “Eigenrhythms: drum pattern basis sets for classification and generation,” in *Proc. of 5th International Conference on Music Information Retrieval*, Barcelona, Spain, Oct. 2004, pp. 554–559.

[Ero07] A. Eronen, “Chorus detection with combined use of MFCC and chroma features and image processing filters,” in *Proc. of 10th International Conference on Digital Audio Effects*, Bordeaux, France, Sep. 2007, pp. 229–236.

[Ero09] A. Eronen, “Signal processing methods for audio classification and music content analysis,” Ph.D. dissertation, Tampere University of Technology, Tampere, Finland, Jun. 2009.

[Fit02] D. FitzGerald, E. Coyle, and B. Lawlor, “Sub-band independent subspace analysis for drum transcription,” in *Proc. of 5th Internationl Conference on Digital Audio Effects*, Hamburg, Germany, Sep. 2002, pp. 65–69.

[Fit03a] D. FitzGerald, R. Lawlor, and E. Coyle, “Drum transcription using automatic grouping of events and prior subspace analysis,” in *Proc. of 4th European Workshop on Image Analysis for Multimedia Interactive Services*, 2003, pp. 306–309.

[Fit03b] D. FitzGerald, B. Lawlor, and E. Coyle, “Drum transcription in the presence of pitched instruments using prior subspace analysis,” in *Proc. of Irish Signals & Systems Conference 2003*, Limerick, Ireland, Jul. 2003, pp. 202–206.

[Fit03c] D. FitzGerald, B. Lawlor, and E. Coyle, “Prior subspace analysis for drum transcription,” in *Proc. of 114th Audio Engineering Society Convention*, Amsterdam, The Netherlands, Mar. 2003.

[Fit04] D. FitzGerald, “Automatic drum transcription and source separation,” Ph.D. dissertation, Conservatory of Music and Drama, Dublin Institute of Technology, Dublin, Ireland, 2004.

[Fit05] D. FitzGerald, M. Cranitch, and E. Coyle, “Generalised prior subspace analysis for polyphonic pitch transcription,” in *Proc. of 8th International Conference on Digital Audio Effects*, Madrid, Spain, Sep. 2005, pp. 77–81.

[Fle98] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*, 2nd ed. New York, N.Y., USA: Springer-Verlag New York Inc., 1998.

[Foo99] J. Foote, “Visualizing music and audio using self-similarity,” in *Proc. of ACM Multimedia*, Orlando, Fla., USA, 1999, pp. 77–80.

[Foo00] J. Foote, “Automatic audio segmentation using a measure of audio novelty,” in *Proc. of IEEE International Conference on Multimedia and Expo*, New York, N.Y., USA, Aug. 2000, pp. 452–455.

[Gao03] S. Gao, N. C. Maddage, and C.-H. Lee, “A hidden Markov model based approach to music segmentation and identification,” in *Proc. of 4th Pacific Rim Conference on Multimedia*, Singapore, Dec. 2003, pp. 1576–1580.

[Gil04] O. Gillet and G. Richard, “Automatic transcription of drum loops,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Que., Canada, May 2004, pp. 269–272.

[Gil05a] O. Gillet and G. Richard, “Automatic transcription of drum sequences using audiovisual features,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, Philadelphia, Pa., USA, Mar. 2005, pp. 205–208.

[Gil05b] O. Gillet and G. Richard, “Drum loops retrieval from spoken queries,” *Journal of Intelligent Information Systems*, vol. 24, no. 2, pp. 159–177, 2005.

[Gil05c] O. Gillet and G. Richard, “Drum track transcription of polyphonic music using noise subspace projection,” in *Proc. of 6th International Conference on Music Information Retrieval*, London, England, UK, Sep. 2005, pp. 156–159.

[Gil06] O. Gillet and G. Richard, “ENST-Drums: an extensive audio-visual database for drum signal processing,” in *Proc. of 7th International Conference on Music Information Retrieval*, Victoria, B.C., Canada, Oct. 2006, pp. 156–159.

[Gil07] O. Gillet and G. Richard, “Supervised and unsupervised sequence modelling for drum transcription,” in *Proc. of 8th International Conference on Music Information Retrieval*, Vienna, Austria, Sep. 2007, pp. 219–224.

[Gil08] O. Gillet and G. Richard, “Transcription and separation of drum signals from polyphonic music,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 3, pp. 529–540, Mar. 2008.

[Gol72] K. Goldberg, M. Newman, and E. Haynsworth, “Combinatorial analysis,” in *Handbook of Mathematical Functions with Formulas, Graphs, and Mathe-*

matical Tables, 10th ed., M. Abramowitz and I. A. Stegun, Eds. Washington, D.C., USA: U.S. Department of Commerce, Dec. 1972, pp. 821–827.

[Gol03] J. Goldberger, S. Gordon, and H. Greenspan, "An efficient image similarity measure based on approximations of KL-divergence between two Gaussian mixtures," in *Proc. of Ninth IEEE International Conference on Computer Vision*, 2003, pp. 487–493.

[Gon92] R. C. Gonzales and R. E. Woods, *Digital image processing*. Addison-Wesley, 1992.

[Goo53] I. J. Good, "The population frequencies of species and the estimation of population parameters," *Biometrika*, vol. 40, no. 3, pp. 237–264, 1953.

[Goo04] M. M. Goodwin and J. Laroche, "A dynamic programming approach to audio segmentation and music / speech discrimination," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Que., Canada, May 2004, pp. 309–312.

[Got93] M. Goto, M. Tabuchi, and Y. Muraoka, "An automatic transcription system for percussion instruments," in *Proc. of the 46th Annual Convention IPS Japan, 7Q-2*, 1993, in Japanese.

[Got94a] M. Goto and Y. Muraoka, "A beat tracking system for acoustic signals of music," in *Proc. of ACM Multimedia*, 1994, pp. 365–372.

[Got94b] M. Goto and Y. Muraoka, "A sound source separation system for percussion instruments," *The Transactions of the Institute of Electronics, Information and Communication Engineers D-II*, vol. J77-D-II, no. 5, pp. 901–911, May 1994, in Japanese.

[Got01] M. Goto, "An audio-based real-time beat tracking system for music with or without drum-sounds," *Journal of New Music Research*, vol. 30, no. 2, pp. 159–171, 2001.

[Got02] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical, and jazz music databases," in *Proc. of 3rd International Conference on Music Information Retrieval*, Paris, France, Oct. 2002, pp. 287–288.

[Got03] M. Goto, "A chorus-section detecting method for musical audio signals," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, 2003, pp. 437–440.

[Got06] M. Goto, "AIST annotation for the RWC music database," in *Proc. of 7th International Conference on Music Information Retrieval*, Victoria, B.C., Canada, Oct. 2006, pp. 359–360.

[Gou01] F. Gouyon and P. Herrera, "Exploration of techniques for automatic labeling of audio drum tracks' instruments," in *Proc. of MOSART Workshop on Current Research Directions in Computer Music*, Barcelona, Spain, 2001.

[Gou02] F. Gouyon, P. Herrera, and P. Cano, “Pulse-dependent analysis of percussive music,” in *Proc. of Audio Engineering Society 22nd International Conference on Virtual, Synthetic and Entertainment Audio*, Espoo, Finland, 2002, pp. 396–401.

[Gou05] F. Gouyon, “A computational approach to rhythm description - audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing,” Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, Spain, 2005.

[Gru04] M. Gruhne, C. Uhle, C. Dittmar, and M. Cremer, “Extraction of drum patterns and their description within the MPEG-7 high-level-framework,” in *Proc. of 5th International Conference on Music Information Retrieval*, Barcelona, Spain, Oct. 2004, pp. 150–153.

[Góm06] E. Gómez, B. Ong, and P. Herrera, “Automatic tonal analysis from music summaries for version identification,” in *Proc. of 121st Audio Engineering Society Convention*, San Francisco, Calif., USA, Oct. 2006.

[Hai06] S. Hainsworth, “Beat tracking and musical metre analysis,” in *Signal Processing Methods for Music Transcription*, A. Klapuri and M. Davy, Eds. Springer, 2006, pp. 101–129.

[Haz05] A. Hazan, “Towards automatic transcription of expressive oral percussive performances,” in *Proc. of 10th International Conference on Intelligent User Interfaces*, San Diego, Calif., USA, Jan. 2005, pp. 296–298.

[HB06] P. Herrera-Boyer, A. Klapuri, and M. Davy, “Automatic classification of pitched musical instrument sounds,” in *Signal Processing Methods for Music Transcription*, A. Klapuri and M. Davy, Eds. Springer, 2006, pp. 163–200.

[Hel05] M. Helén and T. Virtanen, “Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine,” in *Proc. of 13th European Signal Processing Conference*, Antalya, Turkey, Sep. 2005.

[Her98] H. Hermansky and S. Sharma, “TRAPS - classifiers of temporal patterns,” in *Proc. of the International Conference on Spoken Language Processing*, Sydney, Australia, Nov. 1998, pp. 1003–1006.

[Her02] P. Herrera, A. Yeterian, and F. Gouyon, “Automatic classification of drum sounds: a comparison of feature selection methods and classification techniques,” in *Proc. of 2nd International Conference on Music and Artificial Intelligence*, Edinburgh, Scotland, UK, Sep. 2002, pp. 69–80.

[Her03] P. Herrera, A. Dehamel, and F. Gouyon, “Automatic labeling of unpitched percussion sounds,” in *Proc. of 114th Audio Engineering Society Convention*, Amsterdam, The Netherlands, Mar. 2003.

[Hub85] L. Hubert and P. Arabie, “Comparing partitions,” *Journal of Classification*, vol. 2, no. 1, pp. 193–218, Dec. 1985.

[Hyv00] A. Hyvärinen and P. Hoyer, “Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces,” *Neural Computation*, vol. 12, no. 7, pp. 1705–1720, 2000.

[Int02] International Organization for Standardization, *ISO/IEC 15938-4:2002 Information technology – Multimedia content description interface – Part 4: Audio*, Geneva, Switzerland, 2002.

[Jeh05] T. Jehan, “Creating music by listening,” Ph.D. dissertation, Massachusetts Institute of Technology, Boston, Mass., USA, Sep. 2005.

[Jel80] F. Jelinek and R. L. Mercer, “Interpolated estimation of Markov source parameters from sparse data,” in *Proc. of Internation Workshop on Pattern Recognition in Practice*, Amsterdam, The Netherlands, May 1980, pp. 381–397.

[Jen04] K. Jensen, “A causal rhythm grouping,” in *Computer Music Modeling and Retrieval*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2004, vol. 3310, pp. 83–95.

[Jen07] K. Jensen, “Multiple scale music segmentation using rhythm, timbre, and harmony,” *EURASIP Journal on Advances in Signal Processing*, 2007, article ID 73205, 11 pages.

[Jur00] D. Jurafsky and J. H. Martin, *Speech and language processing*. Upper Saddle River, N.J., USA: Prentice-Hall, 2000.

[Kap04] A. Kapur, M. Benning, and G. Tzanetakis, “Query-by-beat-boxing: Music retrieval for the DJ,” in *Proc. of 5th International Conference on Music Information Retrieval*, Barcelona, Spain, Oct. 2004, pp. 170–177.

[Kat87] S. M. Katz, “Estimation of probabilities from sparse data for the language model component of a speech recogniser,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, no. 3, pp. 400–401, Mar. 1987.

[Kla99] A. Klapuri, “Sound onset detection by applying psychoacoustic knowledge,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, Ariz., USA, 1999, pp. 3089–3092.

[Kla06a] A. Klapuri and M. Davy, Eds., *Signal Processing Methods for Music Transcription*. Springer, 2006.

[Kla06b] A. Klapuri, A. Eronen, and J. Astola, “Analysis of the meter of acoustic musical signals,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 342–355, Jan. 2006.

[Kla08] A. Klapuri, “Multipitch analysis of polyphonic music and speech signals using an auditory model,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 2, pp. 255–266, Feb. 2008.

[Lee01] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp, Eds. MIT Press, 2001, vol. 13, pp. 556–562.

[Lev06] M. Levy, M. Sandler, and M. Casey, “Extraction of high-level musical structure from audio data and its application to thumbnail generation,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, May 2006, pp. 13–16.

[Lev07] M. Levy, K. Noland, and M. Sandler, “A comparison of timbral and harmonic music segmentation algorithms,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Honolulu, Hawaii, USA, Apr. 2007, pp. 1433–1436.

[Lev08] M. Levy and M. Sandler, “Structural segmentation of musical audio by constrained clustering,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 318–326, Feb. 2008.

[Lid07] T. Lidy, A. Rauber, A. Pertusa, and J. M. Iñesta, “Improving genre classification by combination of audio and symbolic descriptors using a transcription system,” in *Proc. of 8th International Conference on Music Information Retrieval*, Vienna, Austria, Sep. 2007, pp. 61–66.

[Log00] B. Logan and S. Chu, “Music summarization using key phrases,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Istanbul, Turkey, Jun. 2000, pp. 749–752.

[Lu04] L. Lu, M. Wang, and H.-J. Zhang, “Repeating pattern discovery and structure analysis from acoustic music data,” in *Proc. of Workshop on Multimedia Information Retrieval*, New York, N.Y., USA, Oct. 2004, pp. 275–282.

[Luk08] H. Lukashevich, “Towards quantitative measures of evaluating song segmentation,” in *Proc. of 9th International Conference on Music Information Retrieval*, Philadelphia, Pa., USA, Sep. 2008, pp. 375–380.

[Mad04] N. C. Maddage, C. Xu, M. S. Kankanhalli, and X. Shao, “Content-based music structure analysis with applications to music semantics understanding,” in *Proc. of ACM Multimedia*, New York, N.Y., USA, Oct. 2004, pp. 112–119.

[Mad06] N. C. Maddage, “Automatic structure detection for popular music,” *IEEE Multimedia*, vol. 13, no. 1, pp. 65–77, Jan. 2006.

[Mah05] J. P. G. Mahedero, Á. Martinez, and P. Cano, “Natural language processing of lyrics,” in *Proc. of ACM Multimedia*, Singapore, Nov. 2005, pp. 475–478.

[Mai08] J. A. F. Maitland, Ed., *Grove’s Dictionary of Music and Musicians*, 2nd ed. New York, N.Y., USA: The Macmillan Company, Jun. 1908, vol. IV.

[Mar06] M. Marolt, “A mid-level melody-based representation for calculating audio similarity,” in *Proc. of 7th International Conference on Music Information Retrieval*, Victoria, B.C., Canada, Oct. 2006, pp. 280–285.

[McG07] K. McGuinness, O. Gillet, N. E. O’Connor, and G. Richard, “Visual analysis for drum sequence transcription,” in *Proc. of 15th European Signal Processing Conference*, Poznań, Poland, Sep. 2007, pp. 312–316.

[McK07] M. F. McKinney, D. Moelants, M. E. P. Davies, and A. Klapuri, “Evaluation of audio beat tracking and music tempo extraction algorithms,” *Journal of New Music Research*, vol. 36, no. 1, pp. 1–16, 2007.

[Mid99] R. Middleton, “Form,” in *Key terms in popular music and culture*, B. Horner and T. Swiss, Eds. Wiley-Blackwell, Sep. 1999, pp. 141–155.

[Mor07] A. Moreau and A. Flexer, “Drum transcription in polyphonic music using non-negative matrix factorisation,” in *Proc. of 8th International Conference on Music Information Retrieval*, Vienna, Austria, Sep. 2007, pp. 353–354.

[Mül07a] M. Müller, *Information Retrieval for Music and Motion*. Springer Berlin Heidelberg, 2007.

[Mül07b] M. Müller and F. Kurth, “Towards structural analysis of audio recordings in the presence of musical variations,” *EURASIP Journal on Advances in Signal Processing*, 2007, article ID 89686, 18 pages.

[Mül09] M. Müller, S. Ewert, and S. Kreuzer, “Making chroma features more robust to timbre changes,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Taipei, Taiwan, Apr. 2009, pp. 1877–1880.

[Nak04] T. Nakano, J. Ogata, M. Goto, and Y. Hiraga, “A drum pattern retrieval method by voice percussion,” in *Proc. of 5th International Conference on Music Information Retrieval*, Barcelona, Spain, Oct. 2004.

[Ong06a] B. S. Ong, “Structural analysis and segmentation of musical signals,” Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, Spain, 2006.

[Ong06b] B. S. Ong, E. Gómez, and S. Streich, “Automatic extraction of musical structure using pitch class distribution features,” in *Proc. of 1st Workshop on Learning the Semantics of Audio Signals*, Athens, Greece, Dec. 2006.

[Ori01] I. F. O. Orife, “Riddim: A rhythm analysis and decomposition tool based on independent subspace analysis,” Master’s thesis, Dartmouth College, Hanover, N.H., USA, May 2001.

[O’S87] D. O’Shaughnessy, *Speech Communication: Human and Machine*. Addison-Wesley Publishing Company, 1987.

[Pam08] E. Pampalk, P. Herrera, and M. Goto, “Computational models of similarity for drum samples,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 408–423, Feb. 2008.

[Pee02] G. Peeters, A. La Burthe, and X. Rodet, “Toward automatic music audio summary generation from signal analysis,” in *Proc. of 3rd International Conference on Music Information Retrieval*, Paris, France, Oct. 2002, pp. 94–100.

[Pee04a] G. Peeters, “Deriving musical structure from signal analysis for music audio summary generation: "sequence" and "state" approach,” in *Computer Music Modeling and Retrieval*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2004, vol. 2771, pp. 143–166.

[Pee04b] G. Peeters, “A large set of audio features for sound description (similarity and classification) in the CUIDADO project,” Ircam, Paris, France, Tech. Rep., Apr. 2004.

[Pee07] G. Peeters, “Sequence representation of music structure using higher-order similarity matrix and maximum-likelihood approach,” in *Proc. of 8th International Conference on Music Information Retrieval*, Vienna, Austria, Sep. 2007, pp. 35–40.

[Pee08] G. Peeters, “A generic training and classification system for MIREX08 classification tasks: Audio mood, audio genre, audio artist and audio tag,” in *Proc. of Music Information Retrieval Evaluation eXchange*, Philadelphia, Pa., USA, Sep. 2008, extended abstract.

[Pei07] E. Peiszer, “Automatic audio segmentation: Segment boundary and structure detection in popular music,” Master’s thesis, Vienna University of Technology, Vienna, Austria, Aug. 2007.

[Pla99] J. C. Platt, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” in *Advances in Large Margin Classifiers*, A. J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. MIT Press, 1999, pp. 61–74.

[Plu03] M. D. Plumbley, “Algorithms for non-negative independent component analysis,” *IEEE Transactions on Neural Networks*, vol. 14, no. 3, pp. 534–543, May 2003.

[Pol01] A. W. Pollack, “’Notes on...’ series,” The Official rec.music.beatles Home Page (http://www.recmusicbeatles.com), 1989–2001.

[Pol07] G. E. Poliner, D. P. W. Ellis, A. F. Ehmann, E. Gómez, S. Streich, and B. Ong, “Melody transcription from music audio: Approaches and evaluation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1247–1256, May 2007.

[Pöp97] E. Pöppel, “A hierarchical model of temporal perception,” *Trends in Cognitive Sciences*, vol. 1, no. 2, pp. 56–61, May 1997.

[Qui93] J. R. Quinlan, *C4.5: programs for machine learning*. San Francisco, Calif., USA: Morgan Kaufmann Publishers Inc., 1993.

[Rab89] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–289, Feb. 1989.

[Rab93] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Upper Saddle River, N.J., USA: Prentice-Hall, 1993.

[Rao04] V. M. Rao, "Audio compression using repetitive structures in music," Master's thesis, University of Miami, Miami, Fla., USA, May 2004.

[Rav07] E. Ravelli, J. P. Bello, and M. Sandler, "Automatic rhythm modification of drum loops," *IEEE Signal Processing Letters*, vol. 14, no. 4, pp. 228–231, Apr. 2007.

[Rho07] C. Rhodes and M. Casey, "Algorithms for determining and labelling approximate hierarchical self-similarity," in *Proc. of 8th International Conference on Music Information Retrieval*, Vienna, Austria, Sep. 2007, pp. 41–46.

[Roa96] C. Roads, *The computer music tutorial*. MIT Press, 1996.

[Ron96] D. Ron, Y. Singer, and N. Tishby, "The power of amnesia: Learning probabilistic automata with variable memory length," *Machine Learning*, vol. 25, no. 2–3, pp. 117–149, 1996.

[Rüp04] S. Rüping, "A simple method for estimating conditional probabilities for SVMs," CS Department, Dortmund University, Dortmund, Germany, Tech. Rep., Dec. 2004.

[Ryy06] M. Ryynänen, "Singing transcription," in *Signal Processing Methods for Music Transcription*, A. Klapuri and M. Davy, Eds. Springer, 2006, pp. 361–390.

[Ryy08a] M. Ryynänen, "Automatic transcription of pitch content in music and selected applications," Ph.D. dissertation, Tampere University of Technology, Tampere, Finland, Dec. 2008.

[Ryy08b] M. P. Ryynänen and A. P. Klapuri, "Automatic transcription of melody, bass line, and chords in polyphonic music," *Computer Music Journal*, vol. 32, no. 3, pp. 72–86, 2008.

[San04] V. Sandvold, F. Gouyon, and P. Herrera, "Percussion classification in polyphonic audio recordings using localized sound models," in *Proc. of 5th International Conference on Music Information Retrieval*, Barcelona, Spain, Oct. 2004, pp. 537–540.

[Sca06] N. Scaringella, G. Zoia, and D. Mlynek, "Automatic genre classification of music content," *IEEE Signal Processing Magazine*, vol. 22, no. 2, pp. 133–141, Mar. 2006.

[Sch85] W. A. Schloss, "On the automatic transcription of percussive music – from acoustic signal to high-level analysis," Ph.D. dissertation, Center for Computer Research in Music and Acoustics, Stanford University, Stanford, Calif., USA, May 1985.

[Shi05] Y. Shiu, H. Jeong, and C.-C. J. Kuo, "Musical structure analysis using similarity matrix and dynamic programming," in *Proc. of SPIE Vol. 6015 - Multimedia Systems and Applications VIII*, 2005, pp. 398–409.

[Shi06] Y. Shiu, H. Jeong, and C.-C. J. Kuo, "Similar segment detection for music structure analysis via Viterbi algorithm," in *Proc. of IEEE International Conference on Multimedia and Expo*, Toronto, Ont., Canada, Jul. 2006, pp. 789–792.

[Sil00] J. Sillanpää, A. Klapuri, J. Seppänen, and T. Virtanen, "Recognition of acoustic noise mixtures by combined bottom-up and top-down processing," in *Proc. of European Signal Processing Conference*, Tampere, Finland, 2000, pp. 335–338.

[Sin05] E. Sinyor, C. McKay, R. Fiebrink, D. McEnnis, and I. Fujinaga, "Beatbox classification using ACE," in *Proc. of 6th International Conference on Music Information Retrieval*, London, England, UK, Sep. 2005, pp. 672–675.

[Tan05] K. Tanghe, S. Dengroeve, and B. De Baets, "An algorithm for detecting and labeling drum events in polyphonic music," in *Proc. of First Annual Music Information Retrieval Evaluation eXchange*, London, England, UK, Sep. 2005, extended abstract.

[Tar04] T. Tarvainen, "Automatic drum track transcription from polyphonic music," Master's thesis, Department of Computer Science, University of Helsinki, Helsinki, Finland, May 2004.

[Tso04] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *Proc. of the 21st International Conference on Machine Learning*, Banff, Alta., Canada, 2004, pp. 104–112.

[Tur07] D. Turnbull, G. Lanckriet, E. Pampalk, and M. Goto, "A supervised approach for detecting boundaries in music using difference features and boosting," in *Proc. of 8th International Conference on Music Information Retrieval*, Vienna, Austria, Sep. 2007, pp. 51–54.

[Tza99] G. Tzanetakis and P. Cook, "Multifeature audio segmentation for browsing and annotation," in *Proc. of 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Platz, N.Y., USA, Oct. 1999, pp. 103–106.

[Väl06] V. Välimäki, J. Pakarinen, C. Erkut, and M. Karjalainen, "Discrete-time modelling of musical instruments," *Reports on Progress in Physics*, vol. 69, no. 1, pp. 1–78, 2006.

[Vam] Vamtech Enterprises Inc., "Drumtrax 3.0."

[Vin05] H. Vinet, “The SemanticHIFI project,” in *Proc. of International Computer Music Conference*, Barcelona, Spain, Sep. 2005, pp. 503–506.

[Vir03] T. Virtanen, “Sound source separation using sparse coding with temporal continuity objective,” in *Proc. of International Computer Music Conference*, Singapore, Oct. 2003, pp. 231–234.

[Vir06] T. Virtanen, “Sound source separation in monaural music signals,” Ph.D. dissertation, Tampere University of Technology, Tampere, Finland, Nov. 2006.

[vR79] C. J. van Rijsbergen, *Information Retrieval*, 2nd ed. London, England, UK: Butterworths, 1979.

[vS04] D. van Steelant, K. Tanghe, S. Degroeve, B. De Baets, M. Leman, and J.-P. Martens, “Classification of percussive sounds using support vector machines,” in *Proc. of The Annual Machine Learning Conference of Belgium and The Netherlands*, Brussels, Belgium, Jan. 2004.

[Wei99] Y. Weiss, “Segmentation using eigenvectors: a unifying view,” in *Proc. of Seventh IEEE International Conference on Computer Vision*, Kerkyra, Greece, Sep. 1999, pp. 975–982.

[Wel03] J. Wellhausen and M. Höynck, “Audio thumbnailing using MPEG-7 low-level audio descriptors,” in *Proc. of The SPIE Internet Multimedia Management Systems IV*, vol. 5242, Nov. 2003, pp. 65–73.

[Wit91] I. H. Witten and T. C. Bell, “The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression,” *IEEE Transcations on Information Theory*, vol. 37, no. 4, pp. 1085–1094, 1991.

[Xu04] C. Xu, X. Shao, N. C. Maddage, M. S. Kankanhalli, and T. Qi, “Automatically summarize musical audio using adaptive clustering,” in *Proc. of IEEE International Conference on Multimedia and Expo*, Taipei, Taiwan, Jun. 2004, pp. 2063–2066.

[Yos05] K. Yoshii, M. Goto, and H. G. Okuno, “INTER:D: A drum sound equalizer for controlling volume and timbre of drums,” in *Proc. of 2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies*, London, England, UK, Nov. 2005, pp. 205–212.

[Yos06] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, “An error correction framework based on drum pattern periodicity for improving drum sound detection,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, May 2006, pp. 237–240.

[Yos07a] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, “Drumix: An audio player with real-time drum-part rearrangement functions for active music listening,” *Journal of Information Processing Society of Japan*, vol. 48, no. 3, pp. 1229–1239, 2007.

[Yos07b] K. Yoshii, M. Goto, and H. G. Okuno, “Drum sound recognition for polyphonic audio signals by adaptation and matching of spectrogram templates with harmonic structure suppression,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 333–345, Jan. 2007.

[You89] S. J. Young, N. H. Russell, and J. H. S. Thornton, “Token passing: a simple conceptual model for connected speech recognition systems,” Cambridge University Engineering Department, Cambridge, UK, Tech. Rep. CUED/F-INFENG/TR38, Jul. 1989.

[Zha07] T. Zhang and R. Samadani, “Automatic generation of music thumbnails,” in *Proc. of IEEE International Conference on Multimedia and Expo*, Beijing, China, Jul. 2007, pp. 228–231.

[Zhu05] Y. Zhu, K. Chen, and Q. Sun, “Multimodal content-based structure analysis of karaoke music,” in *Proc. of ACM Multimedia*, Singapore, Nov. 2005, pp. 638–647.

[Zil02] A. Zils, F. Pachet, O. Delerue, and F. Gouyon, “Automatic extraction of drum tracks from polyphonic music signals,” in *Proc. of 2nd International Conference on Web Delivering of Music*, Darmstadt, Germany, Dec. 2002, pp. 179–183.

Total number of entries is 174.

Errata and Clarifications for the Publications

- Publication P1: Equation (6) should be:

$$e = \frac{\sum_i \aleph(w_i^R) - \aleph(w_i^T \cap w_i^R) + max(0, \aleph(w_i^T) - \aleph(w_i^R))}{\sum_i \aleph(w_i^R)} \tag{4.1}$$

- Publication P8: Equation (25) should be:

$$R_R = \frac{|\mathbb{F}_{\mathbb{A}} \cap \mathbb{F}_{\mathbb{E}}|}{|\mathbb{F}_{\mathbb{A}}]}. \tag{4.2}$$

Publication P1

J. K. Paulus and A. P. Klapuri, “Conventional and periodic N-grams in the transcription of drum sequences,” in *Proc. of IEEE International Conference on Multimedia and Expo*, vol. 2, Baltimore, Md., USA, Jul. 2003, pp. 737–740.
doi: 10.1109/ICME.2003.1221722

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the Tampere University of Technology’s products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org.

By choosing to view this material, you agree to all provisions of the copyright laws protecting it.

©2003 IEEE. Reprinted, with permission, from Proceedings of 2003 IEEE International Conference on Multimedia and Expo.

CONVENTIONAL AND PERIODIC *N*-GRAMS IN THE TRANSCRIPTION OF DRUM SEQUENCES

Jouni K. Paulus, Anssi P. Klapuri

Tampere University of Technology, P.O.Box 553, FIN-33101 Tampere, Finland
{paulus,klap}@cs.tut.fi

ABSTRACT

In this paper, we describe a system for transcribing polyphonic drum sequences from an acoustic signal to a symbolic representation. Low-level signal analysis is done with an acoustic model consisting of a Gaussian mixture model and a support vector machine. For higher-level modelling, periodic *N*-grams are proposed to construct a "language model" for music, based on the repetitive nature of musical structure. Also, a technique for estimating relatively long *N*-grams is introduced. The performance of *N*-grams in the transcription was evaluated using a database of realistic drum sequences from different genres and yielded a performance increase of 7.6 % compared to a the use of only prior (unigram) probabilities with the acoustic model.

1. INTRODUCTION

Drums and percussive instruments are an essential part of contemporary music and especially of popular music. As a consequence, the recognition and transcription of rhythm sequences from acoustic signals to MIDI has become a topic of interest. Applications of this comprise e.g. automatic music transcription, light effects control, and music information retrieval in general. However, there has been relatively little previous research in this area (for an overview, see [2]). In most of the work in this field, e.g. in [2], the research has focused on the recognition of individual drum sounds without considering mixtures of simultaneous sounds. Also, typically no higher-level modelling of temporal dependencies in rhythm sequences has been attempted.

The purpose in this paper is to transcribe drum sound mixtures that appear in real rhythm sequences. The task has proven to be very difficult with low-level signal processing only. We propose to improve the recognition ability by using higher-level musicological "language models", based on conventional and periodic *N*-grams. In the periodic *N*-gram, the units that are used to predict the probability of a "word" at time *n* are picked at multiples of interval *L* before the word to be predicted, i.e., at time instances $k-(N-1)L, \ldots, k-2L, k-L$. The conventional *N*-gram is a special case of this where $L = 1$. Many basic elements of music exhibit periodically repeating patterns that vary over time. This observation can be made for harmonic and rhythmic elements at a wide range of different time scales. Conventional *N*-grams can be quite successfully used to predict melodies. However, when it comes to the other parts of music, such as accompaniment and rhythm section, instruments usually follows a periodically regular rather than a locally predictable pattern. The applicability of such models is evaluated and a technique which allows the estimation for relatively large values of *N* is proposed.

2. PROBABILISTIC MODEL FOR RHYTHM SEQUENCES

2.1. Notation

Each individual drum sound is associated with a code which uniquely identifies it. For example, the General MIDI standard defines codes for 47 different drum sounds. Let Ψ be a set of drum codes and *Y* the size of this set. Drum sounds are further classified into broader categories by defining a mapping from the set Ψ to a finite alphabet of symbols Σ which represents the drum categories. The size of the alphabet Σ is typically significantly smaller than that of Ψ. Although the size of Σ could be equal to that of Ψ, this kind of very extensive alphabet would limit the ability of the system to generalize or it would require huge amounts of training data in later steps. In this paper, we chose to use an alphabet size *S*=7, where the symbols represent bass drums, snare drums, hi-hats, cymbals, ride cymbals, tom toms, and percussion instruments, respectively. The percussion symbol class operates as a kind of left-over class which contains all the sounds that could not be fitted to any other category.

Words are defined to be unordered subsets of Σ, where each symbol may occur only once. A "word" is interpreted to represent a set of drum categories that are played simultaneously at a given instant of time, the term "simultaneously" to be defined more exactly in Sec. 2.2. A word can be written as a string of symbols $w_i=\{s_1,s_2,...,s_l\}$, where $l \le S$. For example, $\{s_1,s_2\}$ is a word where sounds from two different categories play simultaneously. An empty word which does not contain any symbols is called *silence*.

V is the total number of word types in the language, that is, the *vocabulary size*. This can be calculated as $V = 2^S$. I.e. each word can be represented as a binary number, where one bit per one symbol indicates whether on not the symbol belongs to the word.

2.2. Rhythm sequences

A percussive music performance is modelled as follows. First, musical time is discretized by finding the *tatum* of the incoming musical performance. The term *tatum*, or, time quantum, refers to the shortest durational value in a musical composition that is still more than incidentally encountered. The other durational values (with few exceptions) are integer multiples of the tatum. Tatum is relative to tempo, and its value may gradually change over time, reflecting tempo fluctuations.

After the tatum of a performance has been estimated, a grid of equidistant tatum pulses is aligned with the performance, and each drum event is associated with the nearest grid point. The events that are associated to a same tatum grid point are considered as simultaneous. Following a common notation [4], the rhythm sequence can then be written as a sequence of words

0-7803-7965-9/03/$17.00 ©2003 IEEE

$w_1 w_2 ... w_K$ as w_1^K, where exactly one word is generated per each grid point. If no drum events occur in the vicinity of a grid point, an empty word is generated.

2.3. Prior probabilities for words

Given a representative database of rhythm sequences, prior probabilities for different word types, $P(w_k)$, $w_k = 0, ..., 127$ can be estimated by performing tatum estimation for each performance in the database and by counting the number of occurrences of each word type in the whole database in relation to the total number of all word occurrences in the database. From the point of view of *N*-grams these can be called *unigram* probabilities.

2.4. Conventional *N*-grams

N-grams have been found to be a convenient way of modelling the sequential dependencies in natural languages. An *N*-gram uses $N - 1$ previous words to predict what the next word would be. This involves an $(N–1)^{\text{th}}$ order *Markov assumption*, i.e., that the probability of the next word depends only on the *N*–1 immediately preceding words. Applying the *N*–gram model, the probability of a word sequence can be calculated as

$$P(w_1^K) = \prod_{k=1}^{K} P(w_k | w_{k-N+1}^{k-1}) . \tag{1}$$

Given a representative database, the *N*-gram probabilities $P(w_k | w_{k-N+1}^{k-1})$ can be estimated by counting the number of times a certain word occurs after a certain prefix, and dividing this by the count how many times the prefix occurs in the database:

$$P(w_k | w_{k-N+1}^{k-1}) = \frac{C(w_{k-N+1}^{k})}{C(w_{k-N+1}^{k-1})} . \tag{2}$$

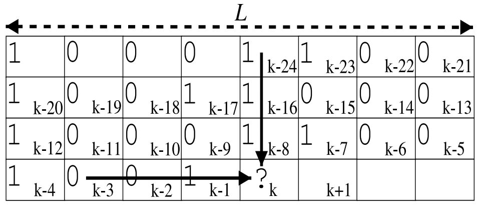
A typical problem with *N*-grams is that, even for a modest vocabulary size *V*, the size of the database does not suffice to estimate the probabilities for large *N*. The number of probabilities to be estimated for a given *V* and *N* is V^N, which requires a rather large database already for *N*=3. Because the training set is usually not extensive enough to estimate all possible *N*-gram probabilities reliably, some smoothing method is applied to the word counts before transforming them to probabilities. Here Witten-Bell smoothing was used [9].

In the case of rhythm sequences, a potential solution to the problem of insufficient data is to estimate *N*-gram probabilities for each individual symbol separately, instead of estimating them for words. Equations are the same as those given above, but with w_k getting only binary values (symbol does or does not occur in the word). Now the size of the vocabulary in the *N*-gram modelling shrinks to *V*=2, with “1” indicating that the symbol occurs at a given point, and “0” vice versa. In this case, *N*-gram probabilities for large values of *N* can be estimated. In our example, the number of probabilities to be estimated for word bigrams is 128^2, which is equal to 2^{14}, i.e., 14-grams for binary data.

Symbol *N*-grams can be used to predict words by combining the symbol-by-symbol predictions as

$$P(w_k | w_1^{k-1}) = \prod_{s_n \in w_k} P(s_n | w_1^{k-1}) \prod_{s_m \notin w_k} (1 - P(s_m | w_1^{k-1})) , \tag{3}$$

where $s_n \cup s_m = \Sigma$. In this way, the training corpus can be utilized more efficiently. However, it is not memory-efficient to store long word *N*-grams, but only the symbol *N*-grams which are then combined according to Eq. (3) when processing files.



1	0	0	0	1 k-24	1 k-23	0 k-22	0 k-21
1 k-20	0 k-19	0 k-18	1 k-17	1 k-16	0 k-15	0 k-14	0 k-13
1 k-12	0 k-11	0 k-10	0 k-9	1 k-8	1 k-7	0 k-6	0 k-5
1 k-4	0 k-3	0 k-2	1 k-1	? k	k+1		

Fig. 1. The idea of periodic *N*-grams illustrated. Quantized time indices are represented with the smaller font. Horizontal arrow represents conventional quadrigram prediction and the vertical arrow periodic quadrigram prediction with *L* = 8.

Constructing *N*-grams separately for each symbol may lead to a situation where the predicted symbols at a given point of time together constitute a very improbable word. This problem can be solved by combining the use of symbol *N*-grams with word priors.

2.5. Periodic *N*-grams

Percussive rhythms exhibit periodicity at different time scales. This observation can be utilized by constructing periodic *N*-grams, as illustrated in Figure 1. Instead of the conventional *N*-gram model given in Eq. (1), the events that are used to predict the word w_k are taken at multiples of interval *L* earlier, at $w_{k-(N-1)L}^{k-L}$. The conventional *N*-gram is a special case of this where $L = 1$. Preferably, *L* is set to be the length of a relatively prominent repetition period, such as the rhythmic pattern, or the musical measure length. In Figure 1, *L* is set to be the length of the musical measure, and the bass drum is being predicted. The horizontal arrow represents a conventional quadrigram prediction, and the vertical arrow a periodic quadrigram prediction with period *L* = 8.

To apply periodic *N*-grams efficiently, a musical meter estimation process is required to find a suitable *L*. Musical meter estimation is a difficult problem in itself, and is above the scope of this paper. We have earlier presented a model which estimates the tatum, tactus (foot tapping rate), and the musical measure length from an acoustic musical signal [6]. We suggest setting *L* to correspond to the musical measure length.

3. ACOUSTIC MODELING

The foundation of signal analysis is in reliable low-level observations. Without being able to reliable extract information at the lowest level, no amount of higher level modelling is going to lead to a correct analysis.

Two independent acoustic models were constructed which together constitute the overall model which provides low-level observations for further statistical processing. The first model calculates the probability that a given audio segment (word) represents an empty word, i.e., silence. The second model assumes that the given segment is not silence and computes the probabilities for each of the 127 non-empty words to have generated the acoustic signal. The latter model is considered first.

3.1. Word recognition

The acoustic model for recognizing non-empty words is based on a Gaussian mixture model (GMM) classifier and Mel-frequency cepstral coefficients (MFCCs) and ΔMFCCs as features [8]. The model was constructed using the following steps:

- 100 random instances of each of the 127 non-empty words were synthesized by allotting each constituent sound (symbol) from a drum sound database and by mixing the sounds. This acoustic database was *not* used in the subsequent testing stage.
- For each sample, six MFCCs and six ΔMFCCs (after discarding the zeroth coefficient) were calculated in successive 20 ms frames with 75 % overlap up to 150 ms after the onset of the sounds. Feature vectors from all the 100 samples were catenated to a single matrix. These matrices were then mean and variance normalized.
- A GMM with two components was used to model the distribution of the feature values for each of the 127 non-empty signals.

3.2. Silence detection

In tests it proved out that the concept of an empty word or the silence was difficult to model, and preceding sounds which continue ringing at an empty grid point are easily confused with hihat sounds. Since silence and hi-hat are the two most often appearing word types (see Fig. 2.), this was a serious problem. It was approached with support vector machines, normally used in binary classification. For a detailed description, refer to [1]. The used implementation was *SVMlight*-toolkit [3] and a radial basis function (RBF) kernel was used.

The output of the SVM is a distance from the decision surface. Normally just the sign of the output matters. In our case, a probability value was needed because the *N*-grams are based on probabilistic computation. In [5] a sigmoid function is used for modifying the SVM output. As the distance from the hyperplane increases, the more probable it is that the sample is classified correctly. When choosing the “silence” class to be positive and the “sound” negative, the probability of this word being empty is then acquired from manipulating the SVM output $f(x)$ with sigmoid function:

$$P(w = \{\varnothing\}) = \frac{1}{1 + e^{-Kf(x)}} \qquad (4)$$

where K is a constant determining the steepness of the sigmoid. Finally the result is scaled as

$$P'(w = \{\varnothing\}) = \max\{P(w \neq \{\varnothing\})\}P(w = \{\varnothing\}) \qquad (5)$$

and catenated to the vector of non-empty word probabilities, which are scaled with $(1 - P(w = \{\varnothing\}))$.

The features that are used for silence detection are crest factor, kurtosis, skewness, zero-crossing rate, RMS-value and temporal centroid [2]. They are extracted from the whole length of the grid point. In the SVM training the samples for class “sound” are collected from extracting the features from 150 ms part of all generated word samples, whereas the samples for class “silence” are collected by extracting the same features at the time range 150-300 ms from all of the generated words. This mimics the situation that some sounds are still echoing in the background at the moments of silence.

4. VALIDATION EXPERIMENTS

The proposed methods were validated by applying it to the automatic transcription of drum sequences. Input to the system was presented as an acoustic signal, and the output consisted of a sequence of recognized words, i.e. a list of drum categories playing at each time instance. This can be written to a MIDI file or displayed to the user in a symbolic form.

We assume that the temporal framework, i.e. the tatum lengths T_0 and musical measure length LT_0, are given along with the acoustic input signals. A method which can automatically estimate these in acoustic input signals has been presented in [6]. However, in order to make the results in this paper as unambiguous as possible, we use manually determined tatum and musical measure length values in the following simulations. Thus the only remaining task is to decide what drum sounds play at each given time instant.

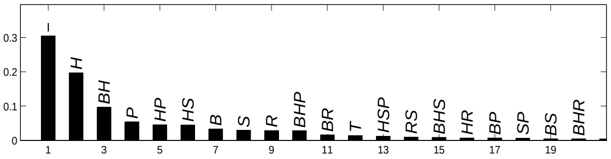


Fig. 2. Occurrence frequencies of the 20 most probable words in the Drumtrax database.

4.1. Estimation of the prior and *N*-gram probabilities

A commercial database, *Drumtrax Library* 3.0, was used to estimate the *N*-gram probabilities and prior probabilities for words. The database consists of 359 drum performances recorded in real time by studio drummers and organized into 14 different categories (genres). The performances are stored as MIDI files, the average length of which is 140 seconds. From each category five performances were taken to test set and the rest were used in calculating the *N*-grams. The test sequences were synthesized using *Timidity* program from MIDI files into monophonic audio files with sample rate of 44100 Hz.

Word *N*-gram probabilities were estimated by converting MIDI performances into a sequence of words as described in Sec. 2.2, and then by using Eq. (2). *N*-grams for each of the seven individual symbols were calculated by converting word sequences into binary sequences (see Sec. 2.2) before applying Eq. (2). For symbols, *N*-grams for *N*={5,10} were estimated, and for words, unigrams (*a priori* probabilities), bigrams and trigrams could be meaningfully estimated. Witten-Bell smoothing was applied to the probabilities to account for data sparseness [9]. Even though the training corpus was quite extensive not even all bigram probabilities were able to be approximated reliably, even such words exist that were not at all present at the training corpus. This is yet another motivation to use symbol *N*-grams and calculate word probabilities from them.

Figure 2 shows the prior probabilities of the 20 most frequently occurring words in the Drumtrax database. In the figure, characters, *B, S, H, C, T, R, P* refer to bass drum, snare drum, hihat, cymbal, tom tom, ride cymbal, and percussion, respectively, and *BH*, for example, means a bass drum and a hi-hat occurring simultaneously. The most probable word is the empty word (silence). As can be seen, the distribution is heavily concentrated to the few most probable words.

4.2. Models in comparison

The simplest recognition experiment was done using only the acoustic models. More exactly, the events at each tatum point of an incoming acoustic signal were classified to the most likely word model. When the word classification has been performed, all the symbols (drum sounds) at each time point are known.

In the second experiment, only the word priors (unigrams)

Table 1: Performance of different *N*-gram systems.

method	symbol error rate (%)	improvement from baseline (%)
acoustic model	76.1	
baseline	49.5	
1. conv. word bigram	47.2	4.7
2. per. word bigram	46.6	5.8
3. conv+per word bigram	47.1	4.9
4. conv. word trigram	46.9	5.2
5. per. word trigram	46.8	5.5
6. conv+per word trigram	46.5	6.0
7. conv. sym. quintagram	46.8	5.5
8. per. sym. quintagram	45.9	7.3
9. conv. sym decagram	45.7	7.6
10. per. sym. decagram	46.0	7.1

were used along with the acoustic model. An as could be predicted from the highly compacted distribution of Fig. 2, this makes a significant improvement to the acoustical model performance. This result operates also as the baseline result to which the *N*-gram enhancements are then added.

Several different *N*-gram types were added to the baseline: (1) conventional bigrams for words, (2) periodic bigrams for words, (3) both conventional and periodic bigrams for words, (4) conventional trigrams for words, (5) periodic trigrams for words, (6) both conventional and periodic trigrams for words, (7) conventional quintagrams (*N*=5) for symbols, (8) periodic quintagrams for symbols, (9) conventional decagrams (*N*=10) for symbols and (10) periodic decagrams for symbols.

Integration of the *N*-grams to the acoustic model was done simply by multiplying the probabilities of the acoustic model and the *N*-gram prediction. In the systems where conventional and periodic *N*-grams were used simultaneously, the periods-behind words were taken from along the path which was decoded up to that point. The final words to the transcription were chosen with greedy decoding. Globally optimizing Viterbi decoding was also tested for methods 1-6, but it had no major effect on the results.

4.3. Results

To be able to evaluate system transcription results automatically, a symbol error rate calculation formula was introduced. There exists three different error types that are omission (certain symbol should be present at word, but it is not), insertion (certain symbol should not be present at word, but it is) and substitution (combination of two previous error types, but counted only as one error). The final error rate is defined by

$$e = \sum_i \frac{(\aleph(w_i^R \cap w_i^T) + \max(0, \aleph(w_i^T) - \aleph(w_i^R)))}{\sum_i \aleph(w_i^R)} \qquad (6)$$

where i goes through all the test grid points, w_i^R is the set of symbols which are present in the word at that point in reference transcription, w_i^T is similarly the set of symbols in the word found by the system and $\aleph(\ )$ means the cardinality of the set.

The final error rates for all of the systems and improvement of *N*-gram methods from the baseline can be seen in Table 1. An overview of the errors for each of the symbols can be seen in Fig. 3. A large portion of the snare drum omissions are related to bass drum insertions. Another significant source for bass drum insertions is the omission of percussions. Also about half of the hi-hat omissions are related to insertion of silence. These are the main sources for errors and should be noted in future work.

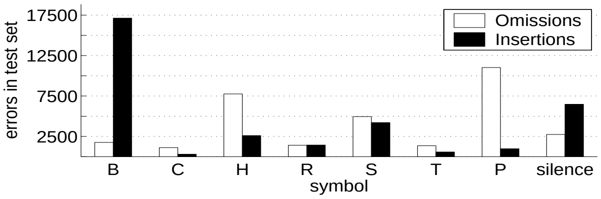


Fig. 3. Omission and insertion errors for each symbol.

5. CONCLUSIONS

We have presented a system for transcribing long sequences of drum sound mixtures. Higher-level statistical modelling improved the performance of acoustic models. Both periodic *N*-grams for words and *N*-grams for symbols have the ability to use information from a longer portion of the past events when predicting the words. The presented statistical models are not limited to percussive part only but can be used for other musical structures, too.

As suggested in [7], there are cases where increased memory length does not increase the prediction accuracy in same ratio. In those cases it would be useful to estimate the gained performance improvement when increasing the *N* and if it is below some predetermined threshold, adhere to the smaller one.

6. REFERENCES

[1] C. Cortes, V. Vapnik, “Support-Vector Networks,” Machine Learning, vol. 20 no. 3, pp. 273-297, 1995.

[2] P. Herrera, A. Yeterian, F. Gouyon, “Automatic classification of drum sounds: a comparison of feature selection methods and classification techniques,” Proc of 2nd International Conference on Music and Artificial Intelligence, pp. 69-80, 2002.

[3] T. Joachims, “Making Large-Scale SVM Learning Practical,” In B. Schölkopf, C. Burges, A. Smola, (ed), “Advances in Kernel Methods-Support Vector Learning,” MIT-Press, 1999.

[4] D. S. Jurafsky, J. H. Martin, “Speech and Language Processing,” Prentice-Hall, 2000.

[5] A. Madevska-Bogdanova, D. Nikolic, “A new approach of modifying SVM outputs,” Proc.of IEEE-INNS-ENNS International Joint Conference on Neural Networks 2000, vol. 6, pp. 395-398, 2000.

[6] J. Paulus, A. Klapuri, “Measuring the similarity of rhythmic patterns,” Proc. of 3rd International Conference on Music Information Retrieval 2002, pp. 150-156, 2002.

[7] D. Ron, Y. Singer, N. Tishby, “The power of amnesia: learning probabilistic automata with variable memory length,” Machine Learning, vol. 25, no. 2-3, pp. 117-149, 1996.

[8] L. Rabiner, B. H. Juang, “Fundamentals of Speech Recognition,” Prentice-Hall, New Jersey, 1993.

[9] I. H. Witten, T. C. Bell, “The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression,” IEEE Trans. on Information Theory, vol. 37, no. 4, pp. 1085-1094, 1991.

Publication P2

J. Paulus and A. Klapuri, “Model-based event labeling in the transcription of percussive audio signals,” in *Proc. of 6th International Conference on Digital Audio Effects*, London, UK, Sep. 2003, pp. 73–77.

©2003 J. Paulus and A. Klapuri.

MODEL-BASED EVENT LABELING IN THE TRANSCRIPTION OF PERCUSSIVE AUDIO SIGNALS

Jouni Paulus, Anssi Klapuri

Institute of Signal Processing, Tampere University of Technology
P.O.Box 553, FIN-33101 Tampere, Finland
{paulus,klap}@cs.tut.fi

ABSTRACT

In this paper we describe a method for the transcription of percussive audio signals which have been performed with arbitrary non-drum sounds. The system locates sound events from the input signal using an onset detector. Then a set of features is extracted from the onset times. Feature vectors are clustered and the clusters are assigned with labels which describe the rhythmic role of each event. For the labeling, a novel method is proposed which is based on metrical (temporal) positions of the sound events within the measures. The system is evaluated using monophonic percussive tracks consisting of non-drum sounds. In simulations, the system achieved a total error rate of 33.7%. Demo signals are available at URL:<http://www.cs.tut.fi/~paulus/demo/>.

1. INTRODUCTION

The aim of this paper is to propose a method for transcribing percussive rhythms from audio signals into a symbolic representation. In particular, the idea is that the input rhythms can be performed using an arbitrary set of percussive sounds, for example by hand-tapping or pencil-clicking on different materials, or even by scat singing (imitating drum instruments by speech sounds). The output of the system consists of a sequence of time-stamped labels which can be further converted e.g. to a MIDI file. Due to the degrees of freedom in regard to the sound sets allowed, only three rhythmically different sounds are attempted to be recognized. These are referred to as *rhythmic role labels*, and are denoted as *B* (bass drum), *S* (snare drum) and *H* (hi-hat), according to the instruments that are usually used to play the roles of these labels in real world music.

A system of the described kind acts as a *user interface* for presenting musical rhythms to a computer. Specific musical instruments or musical education (e.g. score writing ability) is not necessary. Such a user interface has several applications. For example, it can be used to enter a query string to a music information retrieval system. A musician may use it to input a drum track to a score-writing program. A non-musician may want to create music by tapping a rhythm (and simultaneously singing a melody), and to have a computer program which listens and accompanies according to a particular musical style. The most intuitive method for presenting the rhythms would be tapping them with e.g. fingers and record the produced sound with a microphone. If some special hardware is needed for presenting the desired rhythmical patterns to the computer, the usability of the system degrades. Also, the extra hardware may be intimidating for the user. The task for the system is to somehow recognize and label the percussive sound events.

A rather constraining model with only three rhythmically different labels is proposed. Although this drops all timbral nuances of the audio signals, the basic rhythmic percept can be retained for a large body of music from different genres. As has been shown by Zils *et al.*, a drum track of popular music can be presented with very few actual elements occurring and still it will produce the same rhythmic percept as the original track [1]. Their system attempts to extract a drum track consisting of bass and snare drum occurrences. Here, the label *H* was added, because even though the rhythmic percept generated by only bass and snare drums are close to the original one, still something is missing. This missing part is in popular music often played by hi-hats, hence the label. The initial intuition predicts that this kind of model for rhythms suits for the genres pop, rock, blues and hip hop, and to some degree for electronic music.

To our knowledge, transcription of percussive tracks performed with arbitrary non-drum sounds has not been attempted before. Transcription of percussive tracks performed with actual drum instruments has been attempted e.g. by FitzGerald *et al.* who used sub-band independent subspace analysis in transcribing drum tracks consisting of kick drums, snare drums and hi-hats [2]. Their system used manually set rules in determining the correct naming of the found components. Performance of their system was quite reasonable (total success rate 89.5%) considering that the material was polyphonic. The weak point is that their system needed human interference and so the system can be used with one kind of a drum set only, not with arbitrary sounds. Virtanen used a data-adaptive sparse coding approach, where the cost function took temporal continuity into account in [3] when trying to separate different sound sources from a mixture. He tested the system in automatic drum transcription, trying to separate kick and snare drums from polyphonic mixtures. The total error rate for his system was 34%. The weakness in that system was that it needed spectral templates to identify the separated sources. Also hi-hats were not separated due to their relatively weak energy.

When normal drum sounds are used, an acoustic-model based approach can be used. Earlier we presented a system for transcribing polyphonic drum tracks of real drums with the aid of simple

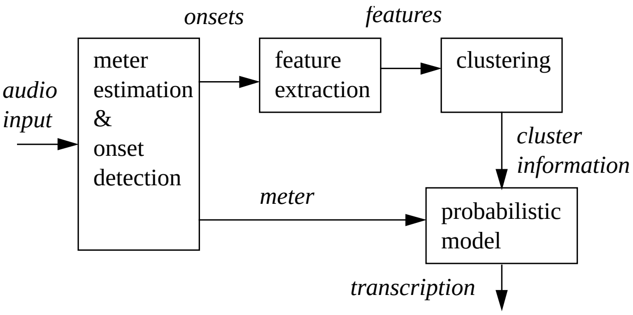


Figure 1: *System overview.*

acoustic models and *N*-gram based language models in [4]. But when using arbitrary sounds in the input, acoustic information it is not enough to identify the sounds. In [5] Patel and Iversen have studied the North Indian tabla drumming tradition in which a system of nonsense syllables is used to name drum sounds. They wanted to know if there exists an acoustic and perceptual basis for the mapping from drum sounds to these syllables. As a result they found that there exists acoustic features, such as spectral centroid and decay time, that can be used in the mapping when used in relation to each other. In their perceptual tests, people unfamiliar with tabla drumming were able to create the mapping quite well when sets of two syllable sounds and corresponding tabla sounds were presented to them. This method works when individual sounds need to be labeled, but the metrical information information tends to overrule the acoustic information when dealing with rhythmical patterns. This is partly due to the diversity of the possible sound sets.

2. PROPOSED METHOD

Overview of the proposed system is shown in Figure 1. The bottom-up clustering part of the system resembles that of Herrera *et al.* where drum tracks consisting of hi-hats, bass and snare drums are automatically labeled. Their system analyses the signal using a constant temporal grid, extracts features at each grid point and finally clusters the extracted features [6].

Another system which uses percussive sound clustering is that of Wang *et al.* which detects percussive sounds in a musical piece and then clusters them into as many clusters as needed according to their perceptual similarity. The obtained cluster information is then used in reconstructing acoustic signal in the case of a packet loss in the transmission of an encoded signal [7].

In our system, features are extracted only at the beginnings of detected sound events, similarly to [7]. For this purpose, onset detection is performed using the mid-level representation of the system presented in [8]. At each onset location a small frame of the signal is extracted and analysed with a method similar to [6]. The analysis result, i.e. the clustering information, is then inserted to a grid of tatum pulses according to the timing information resulting to a symbolic representation. As the crucial step, this information is fed to the labeling system, which then uses a simple probabilistic model in determination the labeling. The term *labeling* is used to refer to the naming of the created clusters using the limited set of rhythmic role labels available. If the clustering was not accomplished totally correct, a simple algorithm for post-labeling cluster assignment changes is applied.

2.1. Meter estimation and sound onset detection

Temporal segmentation in the proposed system is done using the musical meter estimator described in [8]. *Meter* refers to the temporal regularity of music signals, consisting of pulse sensations at different levels. The applied meter estimator analyses meter at three different time scales. *Beat* (foot tapping rate) is the most prominent level. *Tatum* (time quantum) refers to the shortest durational values that are still more than incidentally encountered. The other durational values, with few exceptions, are integer multiples of the tatum. Musical *measure* is related to the harmonic change rate and to the length of rhythmical patterns in music. The accuracy of the meter estimator has been evaluated in [8] and it was found to be applicable in music from different genres.

Here, information about the temporal structure is used for two purposes. First, the bottom-up feature extraction takes place only at the instants of detected *onsets*. The feature vectors are then clustered further as will be described in Sec 2.2 and 2.3. Secondly, a subsequent probabilistic model uses the metrical positions of the sound events belonging to each cluster to infer the rhythmical role (label) of each cluster. This is described in more detail in Sec. 2.4. A discrete time grid of equidistant tatum pulses is created using the information extracted from the signal.

2.2. Feature extraction

From the location of each detected sound onset, a part of the signal is extracted using a Hanning window. Since the sound events are limited in time by their nature, a rectangular window could also be used. The length of the window is to the next detected onset, but 100 ms at maximum. From each frame temporal centroid, signal crest factor, signal energy, spectral kurtosis and six MFCCs (Mel-frequency cepstral coefficients) are extracted. The zeroth MFCC is not used. Finally the features are normalized to have zero mean and unity variance over time.

2.3. Clustering of sound events

The normalized feature vectors are clustered with fuzzy K-means algorithm. Like with its crisp version, the number of desired clusters is set manually. The term *crisp* is used here as the opposite of fuzzy. In addition to the normal cluster information, the algorithm also calculates for each data point degrees of membership to each cluster. The data point is assigned to the cluster to which it has the largest membership value. When left to this state, the clustering result is similar to the one produced by the crisp version. The membership values are used in the post-labeling enhancement.

The result after clustering is a sequence of time stamped cluster numbers $c_t \in \{1, 2, \ldots, K\}$, where K denotes the total number of clusters and t is the continuous time index over the signal. Each cluster number c_t is assigned to the nearest grid point. If there is no cluster number assigned to a certain grid point, it will contain the number 0. The resulting grid contains the cluster numbers $c_i \in \{0, 1, 2, \ldots, K\}$, where i is discrete time index over the signal in steps of one tatum. In further steps, when using

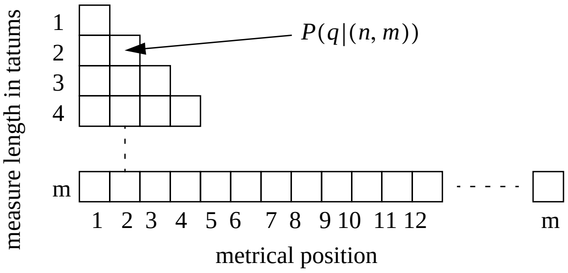


Figure 2: *Data representation of the probabilistic model. q denotes the label, n metrical position, m length of the measure in tatums, and P the probability of the label q to be present at the position* (n, m).

the term cluster numbers, the set $\{0, 1, 2, \ldots, K\}$ is referred.

The degrees of cluster memberships after time quantization, are store in a matrix U, where the element $(U)_{i,k}$ is the degree of membership of the data point at the time *i* in the cluster number *k*. On the locations *i* where no onsets was assigned to, the matrix U contains the value 1 in location corresponding to the cluster number $k = 0$, and 0 on the locations corresponding to the other clusters. Similarly, the value 0 is set to the location corresponding to the cluster number $k = 0$ on the locations *i* where any other cluster number was assigned.

2.4. Probabilistic model

The remaining problem is to find a way for labeling the clusters, i.e. to find a mapping $L:\{0, 1, 2, \ldots, K\} \rightarrow \{\varnothing, B, H, S\}$ from cluster numbers k to labels $q \in \{\varnothing, B, H, S\}$. The label $\varnothing$ is directly mapped to the cluster number representing silence, i.e. $k = 0$. The rest of the mapping could be created with the methods of acoustic pattern recognition, e.g. a Gaussian mixture model. The main problem with acoustic models is that they cannot generalize to extreme cases like the sound set variation considered here. If the input signals are performed with drum instruments, an acoustic model based recognition system can be constructed, as we demonstrated in [4]. As the aim was to be able to handle and label any percussive tracks, independently of the used instruments, acoustic models need to be set aside. Instead, a method relying on the timing of sound events within patterns is introduced.

The model estimates the probability of a certain rhythmic label q to be present at a certain time index within a measure, when the time is discretized to steps of one tatum. An illustration of the data structure of the matrix of probability values is in Figure 2. The measure length in tatum units is m and $n \in \{1, 2, \ldots, m\}$ denotes the position of the sound event within the measure. Since there exists numerous different musical time signatures, the measure lengths from 1 to 48 were modeled.

2.4.1. Model estimation

Probabilities for each label q to occur at different metrical positions (n, m) were estimated using a commercially available MIDI database Drumtrax 3.0. The database covers most of the western musical genres containing 359 performances in total. The pieces are organized in 14 different categories, of which 13 are different genres and the last one is a kind of a tool box. Varied subset of 26 pieces (two from each genre) was left for the test set and the rest were used in training of the model. The division was done for ten times and the presented results are averaged over all tests. In probability estimation the musical grid of songs was annotated by hand and notes are distributed to this discrete time grid. Then each MIDI drum instrument is assigned to belong to one label q. The grid is divided to measures and they are handled individually. The number of occurrences of each label in each metrical position are calculated to a data structure seen in Figure 2, and the resulting probabilities estimated.

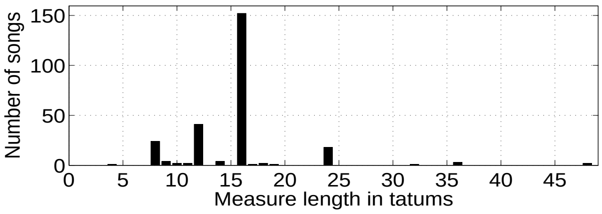


Figure 3: *Number of songs in the training set having a certain length of measure in tatums.*

Though the used training set contained quite an extensive range of pieces, not all probabilities could be determined due to the lack of proper data. The number of songs in the training subset of the database, having a certain measure length in tatums can be seen in Figure 3. This zero occurrence problem was handled by applying Witten-Bell smoothing. An example of the produced probabilities can be seen in Figure 4.

2.4.2. Model usage

Each of the cluster numbers *k* is mapped to one of the rhythmic role labels *q*. After assigning a mapping, the resulting probability can be calculated with

$$P(L) = \prod_i P(q_i|(n_i, m)), \qquad (1)$$

where *i* denotes the discrete time index, q_i the label assigned to that position using the mapping *L*, n_i the metrical position of the time index *i* within the measure of the length *m*. Because of the small number of the clusters and possible labels, every possible mapping permutation can be calculated in *brute force* and the optimal one found. The optimal labeling is defined to be the one having the largest total probability. This labeling strategy is based on the assumption that a majority of events are correctly clustered.

2.5. Post-labeling cluster changes

Since it is more than likely that not all of the data points are clustered correctly in a realistic situation, a simple method for enhancing the final result is presented. The basis for the changes is the cluster membership values from the fuzzy K-means. It is assumed that most of the data points is assigned to a correct cluster and hence only small changes are allowed. Also, it is assumed that the chosen mapping L' is the correct one and it is not changed in the algorithm. It should be noted that the algorithm can change only the data points that contained an onset, i.e. not

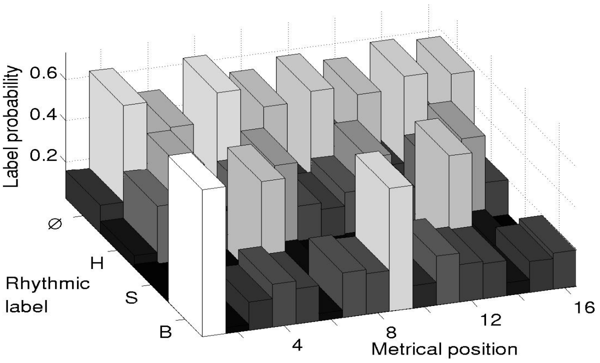


Figure 4: *An example of calculated label probabilities. The measure length is 16.*

the ones containing the label $\varnothing$. It proceeds as follows:

```
calculate base line probability $P_B(L')$ with Eq. (1)
for $i \in \{$grid points in piece$\}$
  for $k \in \{0, 1, \ldots, K\}$
    if $(U)_{i,k} > u_{min}$
      assign label $q_i$ to be the one mapped from cluster $k$
      calculate total probability $P_T(L')$ using the Eq. (1)
      if $P_T(L') > P_B(L')$
        retain the change
      else
        discard the change
      endif
    endif
  endfor
endfor
```

The membership value limit u_{min} is in the range $0\ldots1$. It has the effect that the smaller it is, the less the system trusts the clustering result. If it set to 0, the system can change any cluster assignment. When the total number of clusters is 4, including the silence, the best working value for u_{min} was found to be ca. 0.10. Value this small allows the algorithm to change ca. 40% of label assignments, where a non-silence label was set. The number of actually changed labels is very small, less than 5% in most cases.

3. SYSTEM EVALUATION

The system was evaluated with simulations. The input to the system, audio tracks with percussive sounds was produced by synthesizing 30 seconds from each of the MIDI pieces left to the test set. The test and train set division was done randomly for ten times and the presented results are the average of all divisions. The evaluation could be done automatically since the reference was obtained from the MIDI piece.

In all simulations, the metrical information was annotated by hand instead of using the musical meter estimator. This was done because of the need for automatic transcription evaluation. Since the reference data was annotated in MIDI files, it was decided to use the annotated metrical information. The accuracy of the musical meter estimator was evaluated in [8].

3.1. Audio synthesis

Since the aim was away from the more traditional drum tracks towards tracks performed with arbitrary percussive sounds, a specific synthesizer is needed. It was constructed by recording total of 68 different sounds that can be thought to be used in a real situation. For each sound 15 repetitions were recorded to guarantee some degree of variation in the synthesis. The sounds were distributed so that 48 of them were produced by tapping with hands or pen to tables, books, coffee mugs etc., or by foot tapping with different footwear. The remaining 20 sounds were speech sounds by two persons, both performing the same 10 sounds. From the recorded samples, five sets of percussive sounds were constructed. Two of these set consist only sounds produced with speech and the rest consist of clicks and tapping sounds. It should be noted that since no acoustic modeling was done, there was no need to do any division to train and test sets.

The synthesis produces monophonic signal, i.e. only one sound is playing at a time. In a case where more than one sound was to be played simultaneously, the one to be played was chosen by a simple priority scheme where the sound mapped to from MIDI notes to label *S* have the highest priority, *B* the second highest, and *H* the lowest.

3.2. Simulation setups

Since the total system performance depends heavily on the successive sub-blocks, it was decided to test it in four individual steps:

1. The performance of *onset detection and clustering accuracy.* In this setup, the system performs onset detection, feature extraction and clustering. The clusters are assigned with labels manually, so that the error rate is minimized.
2. Test if the *rhythmic role labeling* is theoretically possible based on the metrical position of events. In this setup, the system was given the reference transcription except that the sound labels were hidden and had to be inferred using the method described in Sec. 2.4.
3. The performance of the *whole system without post-labeling cluster changes*. Here, the system is given only acoustic signal and the metrical information and it needed to detect onsets, extract features, perform clustering and infer the labeling.
4. The performance of the *whole system with post-labeling cluster changes*. This setup is similar to the step 3, but the post-labeling cluster change algorithm is also used.

Each of these cases are evaluated using the same test material and it can be determined which parts of the system may need further development. The used error rate measure was

$$e = \frac{\sum_i f(q_i^{\text{ref}}, q_i^{\text{trans}})}{\sum_i 1}, \tag{2}$$

where *i* is the discrete time index over the whole signal and

$$f(q^1, q^2) = \begin{cases} 1, \text{if } q^1 \neq q^2 \\ 0, \text{otherwise} \end{cases}. \tag{3}$$

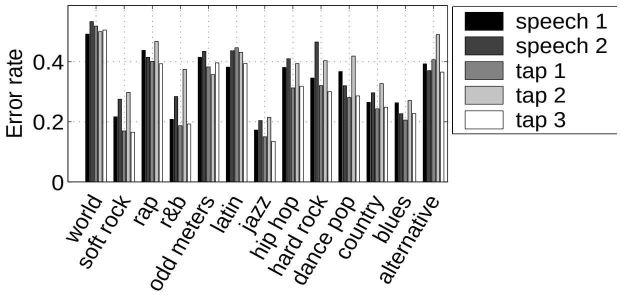


Figure 5: *Genre error rate averages for each of the used five sound sets.*

3.3. Simulation results

The simulation results for the four individual steps are in the Table 1. In addition to the total error rate over all sound sets, there is also the average error rates of speech based sound sets and tapping based sound sets. More detailed error rate statistics for each of the 13 genres and five sets can be seen on Figure 5.

As from the results can be seen, it seems that sounds produced with speech are more difficult for the system to cope with. When inspecting the error rates for each genre, seems that soft rock, r'n'b and jazz are the easiest ones for the system while alternative and world genres are the most difficult. The within genre variations were large, mostly caused by the random choice of the part of the pieces to be analysed. The post-labeling cluster change algorithm turned out to have a very small effect to the total performance. This is due to the fact that errors in labeling step cause high total error rate and minor changes can not fix enough errors.

Some demo signals from the simulations are available at URL:<http://www.cs.tut.fi/~paulus/demo/>.

Table 1: *Error rates for different test steps for speech sounds (average of two sets), tapping sounds (average of three sets) and the average of all systems.*

test step	speech ER	tapping ER	total avg ER
1. clustering	15.28%	13.12%	13.98%
2. role labeling	27.91%	27.91%	27.91%
3. whole w/o post fix	34.91%	33.15%	33.85%
4. whole w/ post fix	35.68%	33.01%	33.67%

4. DISCUSSION

In generation of the *tatum grid*, the tempo is assumed to be constant over the whole signal. Similarly, it is assumed that the length of the measure is constant. However, it is very likely that these assumptions do not hold when operating with real world signals. Musicians may vary the tempo when playing, creating their own versions of the piece and the time signature may be different in the verse and chorus of the piece.

When *constructing the models*, fixed assignment from MIDI notes to the rhythmic role labels turned out to generate problems. Though in many cases the rhythmic roles are operated by the specified drum instruments, this was not the case with all genres. Especially with jazz, latin and world music genres, more exotic percussions were used or the instrument was in a different role (e.g. cymbal in rock is usually in the "snare" role whilst in jazz it may be in the "bass" role). This caused problems in automatic handling of the test data. It turned out that on most of the cases where labeling after "perfect clustering" was erroneous, the time signature was such that there was only few other pieces available for training material.

5. CONCLUSIONS

This paper has introduced a method for transcribing percussive audio signals consisting of arbitrary sounds. It uses the statistical dependencies of metrical positions of rhythmical elements in labeling of the events. The efficiency of the method was evaluated with simulations. The transcription for arbitrary sound sets is a difficult task, and especially errors in label assignment increase the error rate significantly compared to the errors in the clustering step. Based on the results, using metrical positions in the rhythmic role labeling is possible to some degree.

6. REFERENCES

[1] Zils A., Pachet F., Delerue O., Gouyon F., "Automatic Extraction of Drum Tracks from Polyphonic Music Signals", in *Proc. 2nd Int. Conference on Web Delivering of Music (WedelMusic2002),* Darmstadt, Germany, pp. 179-183, 2002.

[2] FitzGerald D., Coyle E., Lawlor B., "Sub-band Independent Subspace Analysis for Drum Transcription", in *Proc. 5th Int. Conference on Digital Audio Effects (DAFX-02),* Hamburg, Germany, pp. 65-69, 2002.

[3] Virtanen T., "Sound Source Separation Using Sparse Coding with Temporal Continuity Objective", in *Proc. of International Computer Music Conference (ICMC2003),* Singapore, 2003, to appear.

[4] Paulus J., Klapuri A., "Conventional and Periodic N-grams in the Transcription of Drum Sequences", in *Proc. of IEEE International Conference on Multimedia and Expo (ICME03),* Baltimore, USA, pp. 737-740, 2003.

[5] Patel A. D., Iversen J. R., "Acoustic and Perceptual Comparison of Speech and Drum Sounds in the North Indian Tabla Tradition: An Empirical Study of Sound Symbolism", in *Proc. 15th International Congress of Phonetic Sciences*, Barcelona, Spain, 2003, to appear

[6] Gouyon F., Herrera P., "Exploration of techniques for automatic labeling of audio drum tracks' instruments", in *Proc. MOSART: Workshop on Current Directions in Computer Music,* Barcelona, Spain, 2001.

[7] Wang Y., Tang J., Ahmaniemi A., Vaalgama M., "Parametric Vector Quantization for Coding Percussive Sounds in Music", in *Proc. IEEE International Conference on Acoustics, Speech & Signal Processing (ICASSP03),* Hong Kong, pp. 652-655, 2003.

[8] Klapuri A., "Musical Meter Estimation and Music Transcription", Paper presented at the Cambridge Music Processing Colloquium, Cambridge University, UK, 2003.

Publication P3

J. Paulus and T. Virtanen, “Drum transcription with non-negative spectrogram factorisation,” in *Proc. of 13th European Signal Processing Conference*, Antalya, Turkey, Sep. 2005.

©2005 European Association for Signal Processing. Reprinted with permission. First published in the Proceedings of the 13th European Signal Processing Conference (EUSIPCO-2005) in 2005, published by EURASIP.

DRUM TRANSCRIPTION WITH NON-NEGATIVE SPECTROGRAM FACTORISATION

Jouni Paulus, Tuomas Virtanen

Institute of Signal Processing, Tampere University of Technology
Korkeakoulunkatu 1, FI-33720, Tampere, Finland
phone: + (358) 3 3115 4790, fax: + (358) 3115 3857, email: jouni.paulus@tut.fi, tuomas.virtanen@tut.fi
web: http://www.cs.tut.fi/˜paulus/research.html

ABSTRACT

This paper describes a novel method for the automatic transcription of drum sequences. The method is based on separating the target drum sounds from the input signal using non-negative matrix factorisation, and on detecting sound onsets from the separated signals. The separation algorithm factorises the spectrogram of the input signal into a sum of instrument spectrograms, each having a fixed spectrum and a time-varying gain. The spectra are calculated from a set of training signals, and the time-varying gains are estimated with an algorithm stemming from non-negative matrix factorisation. Onset times of the instruments are detected from the estimated time-varying gains. The system gave better results than two state-of-the-art methods in simulations with acoustic signals containing polyphonic drum sequences, and overall hit rate of 96% was accomplished. Demonstrational signals are available at http://www.cs.tut.fi/˜paulus/demo/.

1. INTRODUCTION

Automatic music transcription and music information retrieval have recently become more popular as the needed computational power has become available. In general, automatic music transcription can be divided into separate tasks of transcribing the tonal parts and the percussive parts (drums). This paper will concentrate on the drum transcription task, which can be defined as the task of estimating the temporal locations of percussive sound events and recognising the instruments which have been used to produce them. In many systems, like in the one proposed here, the task is modified to detecting the temporal locations of pre-defined percussive sound events.

One of the earliest works on automatic drum transcription was by Schloss, whose system transcribed percussion-only music in which only one instrument is present at a time [12]. The system located the sound event onsets based on rapid increases on amplitude envelope. Each located sound event was classified to one of the groups trained for the system based on subband-energy related features.

Another early work was introduced by Goto and Muraoka [5]. In their system, drum transcription was used as an aid in a beat tracking polyphonic music signals. The onset detection in their system was done by locating power increases in frequency domain. Bass drums and snare drums were sought from the located onsets by inspecting peaks in the spectral content of the onsets. This work was continued by Yoshii et al. in [17], where the event recognition was done by matching template spectrograms of individual bass drum and snare drum events to the detected sound onset locations in polyphonic music. The templates were automatically adapted to the target signal, because the drum sounds used in the signal to be analysed may differ from the template sounds.

Traditional pattern recognition approaches have also been utilised in several ways. Herrera et al. made a thorough comparison of different features and classification techniques for analysing individual drum sound events [6]. In drum transcription, these methods generally first locate possible sound onsets using, e.g., the method suggested by Klapuri [8]. Then a set of features is extracted from the signal at the locations of the detected onsets. The detected onsets are labelled using standard pattern recognition techniques, for example, k-nearest neighbours [11], support vector machines (SVM) [4, 14], or Gaussian mixture models [4, 10]. None of these methods seem to perform clearly better than the others, so some advanced techniques and higher-level processing have been developed to increase the performance, such as, language modelling with explicit N-grams [10] or hidden Markov models [4], or choosing best feature subset dynamically [11].

1.1 Separation of drum sounds

Even though individual drum sound events can be recognised quite reliably [6], the recognition from polyphonic music is a difficult task, because of other simultaneously occurring sounds [4]. Separation of sound sources has been used to address the problem, e.g., by using methods based on independent subspace analysis (ISA) [1, 2], and sparse coding [16].

In the case of music signals, ISA and sparse coding have been used to separate the input signal into a sum of sources, each of which has a fixed spectrum and a time-varying gain. This model suits quite well for representing drum signals. The signal model for spectrum $X_t(f)$ in frame t can be written as a weighted sum of source spectra $S_n(f)$:

$$X_t(f) \approx \sum_{n=1}^{N} a_{n,t} S_n(f), \tag{1}$$

where N is the number of sources, n is the source index, $a_{n,t}$ is the gain of the n^{th} source in frame t, and f is the discrete frequency index.

There are several different criteria for estimating $a_{n,t}$ and $S_n(f)$, including the independence of the sources [1], non-negativity [13], or sparseness of the sources [16]. In some systems, the sources are estimated blindly, i.e., there is no prior knowledge of the parameters of the sources. Also, some proposals for the use of pre-trained sources have been made [15].

Prior subspace analysis (PSA) proposed by FitzGerald simplifies the decomposition by initialising the spectral subspaces $S_n(f)$ with values calculated from a large sample set [3]. Then the time-varying gains $a_{n,t}$ are calculated using matrix inverse, passed through independent component analysis (ICA) and finally subjected to onset detection. The main problem with PSA is that $a_{n,t}$ can have also negative values which do not have a reasonable physical counterpart.

Recently, non-negative matrix factorisation (NMF) has been successfully used in several unsupervised learning tasks [9] and also in the analysis of music signals, e.g., by Smaragdis and Brown [13]. In NMF, both the spectra $S_n(f)$ and gains $a_{n,t}$ are restricted to be non-negative. In the case of audio source separation, this can be interpreted so that the spectrograms are purely additive. It has turned out that the non-negativity constraint alone is sufficient for separating sources, to some degree.

1.2 Improvements in the proposed method

The proposed method combines the ideas of PSA and NMF. The spectrogram of the mixture signal is decomposed into spectrograms of target drum instruments using pre-defined fixed spectra $S_n(f)$ and non-negativity constraints in the estimation of the gains $a_{n,t}$.

Natural drum sounds do not have an exactly fixed spectrum over time. When examined with high frequency resolution, the spectrograms exhibit stochastic nature within an individual sound event. In addition, there are differences between the occurrences of a same drum instrument sound events. The variation of the spectrum is reduced by using a coarse frequency resolution, and the signal model of Equation (1) can be used. The spectrum of a drum instrument is approximately fixed on a coarse frequency grid, e.g., bass drums have low-frequency energy and hi-hats have high-frequency energy.

Some publications have discussed the matter of recognising drum patterns from polyphonic music [5, 11, 17]. Before aiming directly to that level, the transcription task in this paper is restricted to material consisting only of a limited number of different drum instruments. Namely, only bass drum, snare drum, and hi-hat occurrences are transcribed.

2. PROPOSED METHOD

The proposed method consists of three stages: at first, source spectra $S_n(f)$ are estimated for each instrument using training material, as will be described in Section 2.1. At the second stage, each drum instrument is separated from the input signal using the trained spectra and the method that will be described in Section 2.2. Finally, the temporal locations of sound events are sought from the separated signals with the method that will be described in Section 2.3.

The magnitude spectrogram is used as a mid-level signal representation. Since drum transcription requires a good temporal resolution, the length of the analysis frame is 24 ms with 75 % overlap between consecutive frames, leading into temporal resolution of 6 ms.

The segregation between different drum classes can be made using a coarse frequency resolution. Only five bands (20-180 Hz, 180-400 Hz, 400-1000 Hz, 1-10 kHz and 10-20 kHz) were used in the simulations. The number and locations of the bands were not specifically optimised for the transcription, but these yielded the best result from the ones tested (e.g., linearly spaced 512 frequency bins or 25 critical bands). The magnitude spectrogram $X_t(f)$ is obtained by using short-time Fourier transform, summing the squared magnitudes within each band to obtain bandwise energies, and by taking the square root.

2.1 Estimation of the source spectra

There are several possibilities for obtaining the instrument spectra $S_n(f)$ from the training data. In our simulations the best results were obtained by using the following procedure. A set of recordings of individual examples of a certain drum instrument n is taken. NMF [9, 13] is used to dismantle the magnitude spectrogram $Y_t^i(f)$ of each example event i into a product of non-negative spectrum $W^i(f)$ and non-negative time-varying gain h_t^i, so that $Y_t^i(f) \approx W^i(f)h_t^i$. The spectral basis vectors $W^i(f)$ are then averaged over i to produce the instrument spectra $S_n(f)$ of drum instrument n. The procedure is repeated for all instruments $n \in [1,N]$.

2.2 Estimation of the time-varying gains

The separation algorithm estimates time-varying gains $a_{n,t}$ for each drum instrument n in each frame t, so that the magnitude spectrum $X_t(f)$ of the input signal is presented as a weighted sum of the fixed spectra $S_n(f)$, as represented in Equation (1). The estimation is done by minimising a cost function between the observed spectrum $X_t(f)$ and the model $M_t(f) = \sum_{n=1}^{N} a_{n,t}S_n(f)$. The gains $a_{n,t}$ are restricted to be non-negative. The method does not make any explicit assumptions of the independence or sparseness of the gains.

The best transcription result was obtained using the divergence proposed by Lee and Seung [9] as the cost function. The divergence D between $X_t(f)$ and $M_t(f)$ is defined as

$$D(X_t(f)||M_t(f)) = \sum_{t,f} d(X_t(f), M_t(f)), \quad (2)$$

where the function d is defined as

$$d(p,q) = p\log(\frac{p}{q}) - p + q. \quad (3)$$

The divergence is minimised by an iterative algorithm, which uses multiplicative updates, given as

$$a_{n,t} \leftarrow a_{n,t} \frac{\sum_f X_t(f)S_n(f)/M_t(f)}{\sum_f S_n(f)}. \quad (4)$$

The iterative estimation algorithm is given by the following procedure:

1. Initialise each $a_{n,t}$ to unity.
2. Set $M_t(f) = \sum_{n=1}^{N} a_{n,t}S_n(f)$.
3. Update each $a_{n,t}$ using the update rule (4)
4. Evaluate cost function 2 and repeat steps 2 to 4 until the value of the cost function converges.

In our experiments with three sources and a five-band spectral representation, the algorithm took approximately 20-30 iterations to converge.

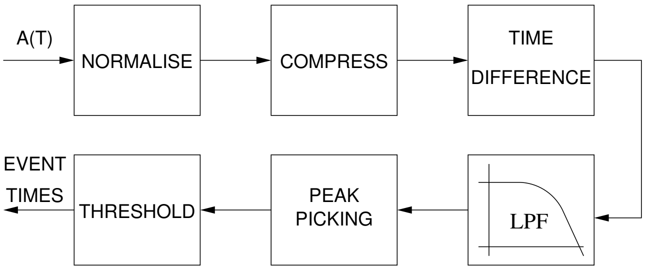


Figure 1: Block diagram of procedure for detecting onsets from an estimated time-varying gain $a_{n,t}$.

2.3 Onset detection

Onset detection of the instrument n is done from the corresponding time-varying gain $a_{n,t}$ with a procedure motivated by the one proposed by Klapuri [8]. Only the time-varying gain is used instead of several sub-band amplitude envelope signals used in the reference. This simplification can be done since the spectrum associated with each gain is fixed, causing all sub-band amplitude envelopes to be of identical form. The algorithm is motivated by the human auditory system, which is sensitive for relative changes in signal level.

The block diagram of the onset detection procedure is illustrated in Figure 1. First, the gain is normalised to range $[0,1]$ to obtain a better control of subsequent steps of the onset detection procedure. The normalised gain $\tilde{a}_{n,t}$ is compressed with $\hat{a}_{n,t} = \log(1 + J\tilde{a}_{n,t})$, where J is a fixed compression factor. The algorithm is not sensitive for the exact value of J; a value of 100 was found to be suitable. The compressed gain is differentiated with $a'_{n,t} = \hat{a}_{n,t} - \hat{a}_{n,t-1}$.

The difference signal $a'_{n,t}$ contains low-amplitude ripple, which is reduced by low-pass filtering. The system is not sensitive to the exact filter characteristics; in our implementation a fourth order Butterworth filter with cut-off frequency 0.25π[1] was used. Finally, the filtered signal is subjected to peak picking. Peaks in the signal represent perceptually salient onsets. Thresholding is used to pick

[1] sampling rate being 167 Hz

only the most prominent peaks. The threshold value can be different for each instrument.

The thresholds needed in the onset detection are estimated automatically from training material with the following procedure. The training signals are separated with the proposed method, and onset are located. By comparing the located onsets to the reference onsets, the threshold value is chosen so that the number of undetected onsets and extraneous detections is minimised. The threshold is calculated for each drum instrument independently.

3. EVALUATION

The performance of the proposed transcription method was evaluated and compared to two other systems using acoustic signals. We used a four-fold cross-validation setup for acoustic material from 4 recording sets, so that three sets were used for training and one set for testing at a time.

3.1 Acoustic material

The simulation database consists of acoustic drum sequences and individual drum samples. Three different drum kits and three different recording locations were used. One of the kits was recorded in two different locations, resulting to total of four recording sets:

1. an entry level kit recorded in a medium sized room,
2. a studio grade kit recorded in a medium sized room,
3. a heavy metal kit recorded in an acoustically damped hall, and
4. an entry level kit recorded in an anechoic chamber.

The acoustic information was recorded using close microphones for bass drums and snare drums, and overhead microphones for hi-hats. Recorded signals were mixed to yield two mix-downs: an unprocessed one, and a "production-grade" processed one where multiband compression, equalisation, and reverberation were used. The reference onsets were acquired by using piezo triggers on bass drums and snare drums. The hi-hats were annotated by hand. The temporal accuracy of the annotated onsets was estimated to be better than 10 ms.

The drum sequences in the evaluation database are fairly simple, consisting only of bass drums, snare drums and hi-hats. Different playing styles are not discriminated, e.g., open, closed and pedal hi-hats are treated as equal. The sequences used were 8-beat, 16-beat, stomp, shuffle and triole, resulting in total of 20 signals. The sequences do not contain only several repetitions of the same pattern, but the players were encouraged to make some variations while playing. Only 15-second excerpts of the sequences were used in the evaluation.

In addition to the sequences, individual drum hits were recorded with 20 repetitions of each. These were used to obtain the spectra $S_n(f)$, as explained in Section 2.1.

3.2 Performance metrics

For each drum instrument, the performance was measured by comparing the transcribed onsets with the reference onsets. A transcribed onset was judged to be correct if it deviated less than 30 ms from a reference onset. The transcribed and reference onsets were matched using the following procedure. At first, the algorithm calculates a $V \times L$ matrix Z of absolute time differences between all transcribed and reference events $Z_{v,l} = |(t_v - t_l)|, v = 1 \ldots V, l = 1 \ldots L$, where V is the number of transcribed events and L is the number of events in reference data. Then the events v and l having the smallest time difference are paired and removed from the distance matrix. This pairing is continued until all remaining time differences are larger than 30 ms or either event set runs out of available events. The b remaining unmatched transcribed events are insertions and the c remaining unmatched reference events are deletions leading to *instrument hit rate* of $R_h = 1-(b+c)/L$. Also, *precision rate* $R_p = (V-b)/V$ and *recall rate* $R_r = (L-c)/L$ were calculated. Precision rate is the ratio of correct detections to all detections, and recall rate is the ratio of correct detections to number of events in the reference annotation. The overall hit rate was calculated as the mean of individual instrument hit rates. The presented performance measures are calculated over the four cross validation iterations.

unprocessed		B	S	H	avg
SVM	R_p %	99	93	92	94
	R_r %	99	93	86	89
	R_h %	98	86	77	**87**
PSA	R_p %	91	77	80	82
	R_r %	95	91	70	78
	R_h %	86	70	46	**67**
NSF	R_p %	100	100	98	99
	R_r%	100	94	96	96
	R_h %	100	93	94	**96**

processed		B	S	H	avg
SVM	R_p %	99	100	95	97
	R_r %	99	93	91	93
	R_h %	98	93	86	**92**
PSA	R_p %	77	83	80	80
	R_r %	92	84	73	78
	R_h %	71	67	51	**63**
NSF	R_p %	98	100	96	97
	R_r %	98	94	96	96
	R_h %	95	94	93	**94**

Table 1: Results for the unprocessed (upper table) and "production-grade" processed (lower table) test signals. *B* denotes bass drums, *S* snare drums, *H* hi-hats, and *avg* the average of *B*, *S* and *H*. *SVM* is the method described in [4], *PSA* the method described in [3], and *NSF* the proposed method.

3.3 Comparisons to other systems

Two other transcription systems were used for comparison with similar evaluation setup. The systems by Gillet et al. [4] and FitzGerald et al. [3] were tested. The method by Gillet et al. initially detects all sound event onsets from the signal, then extracts a set of features from the locations of the detected onsets, and finally uses an SVM classifier for recognising the events. The presence of each drum instrument in the event is detected with a binary classifier, and no sequence modeling is used. The classifiers were trained with the acoustic sequences in the training set. The algorithm implementation was based on the information given in the reference, and the SVM implementation by Joachims was used [7].

In PSA, initially, the spectral basis vectors are calculated from the individual drum hits in the training set. Then, the corresponding time-varying gains are estimated with a matrix inverse and independent component analysis. Finally, the sound onsets are detected from the estimated time-varying gains. The implementation of the original authors was used.

3.4 Results and discussion

The performance evaluation results are presented in Table 1. The proposed method performed better in total than both comparison methods with unprocessed and processed drum signals, though the difference is smaller with processed signals. This is most likely due to the fact that when handling the processed signals, the features used by the SVM method were trained with processed signals, while the spectral models used by the PSA and the proposed method were trained with individual unprocessed hits in both scenarios, because there were no processed individual hits available.

Some preliminary experiments were made to utilise the proposed method also for more complex signals, i.e., signals containing also other drums or melodic instruments. It was noted that the performance of the proposed system degrades if the analysed signal does not fit the model, i.e., other interfering sounds are present in addition to the modelled instruments. Similar performance degradation was noted also with the two reference systems. It is possible that the SVM method may be able to handle more complex input signals, as it's operation does not rely on direct assumptions on the structure of the signal. This problem of method generalisation remains as a subject of further development.

4. CONCLUSIONS

In this paper, we have presented a method for drum transcription. It uses pre-calculated spectra and non-negativity constraints for the gains of the spectra for separating different instruments. The proposed method has been evaluated with simulations and the performance of the presented method has been compared with two reference methods. The proposed method performed better than the two reference methods in simulations with polyphonic drum sequences.

Acknowledgements

The drum database was recorded and annotated by Teemu Karjalainen at TUT.

REFERENCES

[1] M. A. Casey and A. Westner. Separation of mixed audio sources by independent subspace analysis. In *Proc. International Computer Music Conference*, pages 154–161, Berlin, Germany, Aug. 2000.

[2] D. FitzGerald, E. Coyle, and B. Lawlor. Sub-band independent subspace analysis for drum transcription. In *Proc. 5th Internationl Conference on Digital Audio Effects*, pages 65–69, Hamburg, Germany, Sept. 2002.

[3] D. FitzGerald, B. Lawlor, and E. Coyle. Prior subspace analysis for drum transcription. In *Proc. 114th Audio Engineering Society Convention*, Amsterdam, The Netherlands, Mar. 2003.

[4] O. Gillet and G. Richard. Automatic transcription of drum loops. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Canada, May 2004.

[5] M. Goto and Y. Muraoka. A beat tracking system for acoustic signals of music. In *Proc. ACM Multimedia*, pages 365–372, 1994.

[6] P. Herrera, A. Yeterian, and F. Gouyon. Automatic classification of drum sounds: a comparison of feature selection methods and classification techniques. In *Proc. 2nd International Conference on Music and Artificial Intelligence*, pages 69–80, Edinburgh, Scotland, UK, Sept. 2002.

[7] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT-Press, 1999.

[8] A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, Arizona, USA, 1999.

[9] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 556–562. MIT Press, 2001.

[10] J. K. Paulus and A. P. Klapuri. Conventional and periodic N-grams in the transcription of drum sequences. In *Proc. IEEE International Conference on Multimedia and Expo*, volume 2, pages 737–740, Baltimore, Maryland, USA, July 2003.

[11] V. Sandvold, F. Gouyon, and P. Herrera. Percussion classification in polyphonic audio recordings using localized sound models. In *Proc. 5th International Conference on Music Information Retrieval*, Barcelona, Spain, Oct. 2004.

[12] W. A. Schloss. *On the automatic transcription of percussive music – from acoustic signal to high-level analysis*. PhD thesis, Center for Computer Research in Music and Acoustics, Stanford University, Stanford, California, USA, May 1985.

[13] P. Smaragdis and J. C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *Proc. 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 177–180, New Platz, New York, USA, Oct. 2003.

[14] D. Van Steelant, K. Tanghe, S. Degroeve, B. De Baets, M. Leman, and J.-P. Martens. Classification of percussive sounds using support vector machines. In *Proc. The Annual Machine Learning Conference of Belgium and The Netherlands*, Brussels, Belgium, Jan. 2004.

[15] E. Vincent and X. Rodet. Music transcription with ISA and HMM. In *Proc. Independent Component Analysis and Blind Signal Separation*, pages 1197–1204, Granada, Spain, Sept. 2004.

[16] T. Virtanen. Sound source separation using sparse coding with temporal continuity objective. In *Proc. International Computer Music Conference*, Singapore, Oct. 2003.

[17] K. Yoshii, M. Goto, and H. G. Okuno. Automatic drum sound description for real-world music using template adaptation and matching methods. In *Proc. 5th International Conference on Music Information Retrieval*, Barcelona, Spain, Oct. 2004.

Publication P4

D. Alves, J. Paulus, and J. Fonseca, "Drum transcription from multichannel recordings with non-negative matrix factorization," in *Proc. of 17th European Signal Processing Conference*, Glasgow, Scotland, UK, Aug. 2009, pp. 894–898.

©2009 European Association for Signal Processing. Reprinted with permission. First published in the Proceedings of the 17th European Signal Processing Conference (EUSIPCO-2009) in 2009, published by EURASIP.

DRUM TRANSCRIPTION FROM MULTICHANNEL RECORDINGS WITH NON-NEGATIVE MATRIX FACTORIZATION

David S. Alves[1,2], *Jouni Paulus*[2], *and José Fonseca*[1]

[1]Department of Electrical Engineering
Faculty of Science and Technology
New University of Lisbon, Portugal
email:davsantosster@gmail.com

[2]Department of Signal Processing
Tampere University of Technology
Tampere, Finland
email:jouni.paulus@tut.fi

ABSTRACT

Automatic drum transcription enables handling symbolic data instead of plain acoustic information in music information retrieval applications. Usually the input to the transcription system is single-channel audio, and as a result the proposed solutions are designed for this kind of input. However, in studio environment the multichannel recording of the drums is often available. This paper proposes an extension to a non-negative matrix factorization drum transcription method to handle multichannel data. The method creates spectral templates for all target drums from all available channels, and in transcription estimates time-varying gains for each of them so that the sum approximates the recorded signal. Sound event onsets are detected from the estimated gains. The system is evaluated with multichannel data from a publicly available data set, and compared with other methods. The results suggest that the use of multiple channels instead of a single-channel mix improves the transcription result.

1. INTRODUCTION

Drum transcription provides a method to transition from acoustic signal to a symbolic representation of the drum sound content. Drum transcription means the process of detecting the occurrences of drum sound events in a musical performance and recognizing the instruments used. The input performance is usually a single-channel or a stereo recording. Several methods for solving the transcription from material containing only drum hits or from polyphonic music have been proposed in literature, see [5] for a review. The methods are often divided into two categories: event-based and source separation -based methods. The method in the first category locate sound events in the input signal and then recognize the content. In the source separation approaches, the individual target drums are considered to be the sound sources that are separated from the mixture signal, and the sound event onsets are searched from the separated drum signals. The method proposed in this paper belongs to the separation-based approaches. Other recent drum transcription methods include using spectrogram template matching and adaptation [14], using support vector machine classifiers to recognize the sound event content [7], and using continuous hidden Markov models to perform the signal segmentation and classification simultaneously [11].

The input signal is represented as a magnitude spectrogram $\boldsymbol{X}$ with F rows corresponding to frequencies and T columns corresponding to frames. In this representation, the mixture signal can be considered to be a sum of the spectrograms of the N source signals

$$\boldsymbol{X} = \sum_{n=1}^{N} \boldsymbol{X}_n + \varepsilon, \tag{1}$$

where $\boldsymbol{X}_n$ is the spectrogram of the nth source, and ε represents the approximation error. Each of the sources $\boldsymbol{X}_n$ is considered to be a product of two basis vectors

$$\boldsymbol{X}_n = \boldsymbol{s}_n \boldsymbol{a}_n^T. \tag{2}$$

With this approximation and omitting the approximation error term, (1) can be rewritten as a matrix product

$$\boldsymbol{X} = \boldsymbol{S}\boldsymbol{A}, \tag{3}$$

where the component matrices $\boldsymbol{S}$ and $\boldsymbol{A}$ are defined as

$$\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_N] \tag{4}$$

and

$$\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_N]^T. \tag{5}$$

Several approaches have been proposed for solving the decomposition of $\boldsymbol{X}$ into the components $\boldsymbol{S}$ and $\boldsymbol{A}$. One of the first was independent subspace analysis (ISA) [1]. ISA was proposed to solve the problem of applying independent component analysis on a single-channel signal by representing the signal as a spectrogram and considering each frequency band as a channel signal. In this context, each vector $\boldsymbol{s}_n$ can be considered as a frequency basis function and $\boldsymbol{a}_n$ the corresponding time basis function.

ISA performs the separation without any prior knowledge of the sources despite that in many cases some information is available. In [4] prior subspace analysis (PSA) was proposed to utilize prior knowledge of the sources by calculating spectral templates $\boldsymbol{s}_n$ for each target drum in advance and then calculating the corresponding time-varying gains $\boldsymbol{a}_n$. The sound events are then detected from the temporal basis functions.

Another way to perform the decomposition of (3) is to use non-negative matrix factorization (NMF) [13], which assumes that every element in the matrices $\boldsymbol{X}$, $\boldsymbol{A}$, and $\boldsymbol{S}$ are non-negative. This non-negativity constraint fits the magnitude spectrogram representation as it has only non-negative values. Similar to the idea of PSA, use of spectral templates with NMF for drum transcription was proposed in [12]. This paper extends that work.

When a drum kit is recorded in a studio environment, usually each membranophone (drums with membranes, e.g.,

This work was supported by the Academy of Finland, project no. 5213462 (Finnish Centre of Excellence program 2006 - 2011).

kick drum, snare drum, tom-toms) has at least one close microphone and all the cymbals are recorded with two or more overhead microphones. The knowledge of the content and timing of each hit in the recorded signals would enable more flexible editing of the recordings. Since these multichannel signals are available, the transcription is done using them instead of the monophonic or stereophonic mix-down available in the later stages of record production.

The rest of this paper is organized as follows. Section 2 describes the use of NMF for drum transcription starting from the single-channel method and then extending it for multichannel data. Section 3 details the evaluations of the system performance, including the material used, the performance metrics, and the evaluation results. Finally, Section 4 provides the conclusions of the paper.

2. DRUM TRANSCRIPTION USING NON-NEGATIVE MATRIX FACTORIZATION

The factorization of non-negative matrix $\boldsymbol{X}$ to two non-negative matrices $\boldsymbol{S}$ and $\boldsymbol{A}$ can be done using the Lee and Seung algorithm [10] which minimizes the Kullback-Leibler divergence -like distance measure

$$D(\boldsymbol{X}||\boldsymbol{SA}) = \sum_{f,t} [\boldsymbol{X}]_{f,t} \log \frac{[\boldsymbol{X}]_{f,t}}{[\boldsymbol{SA}]_{f,t}} - [\boldsymbol{X}]_{f,t} + [\boldsymbol{SA}]_{f,t} \qquad (6)$$

between the input $\boldsymbol{X}$ and the approximation $\boldsymbol{SA}$. The matrices $\boldsymbol{S}$ and $\boldsymbol{A}$ are initialized with non-negative random values and then iteratively updated with the multiplicative rules

$$\boldsymbol{A} \leftarrow \boldsymbol{A}.* \frac{\boldsymbol{S}^T(\boldsymbol{X}./(\boldsymbol{SA}))}{\boldsymbol{S}^T\boldsymbol{1}} \qquad (7)$$

and

$$\boldsymbol{S} \leftarrow \boldsymbol{S}.* \frac{(\boldsymbol{X}./(\boldsymbol{SA}))\boldsymbol{A}^T}{\boldsymbol{1}\boldsymbol{A}^T}. \qquad (8)$$

In the equations above, $\boldsymbol{1}$ is a all-one matrix of size $F \times T$. Element wise multiplication and division operations are represented by $.*$ and $./$ after [8]. For a tutorial-like overview of other NMF algorithms and the relationship of NMF with its generalization non-negative tensor factorization (NTF), refer to [2]. Using NTF for sound source separation is demonstrated, e.g., in [3].

2.1 Single Channel Input Data

The use of NMF and templates in drum transcription was proposed in [12]. The spectral basis functions (or templates) $\boldsymbol{s}_n$ are calculated from signals containing only the target drum. This signal is factorized into one source and the resulting $\boldsymbol{S}$ containing only one column is assumed to represent the main characteristics of the target drum, and it is taken as the spectral template for that drum. If several training signals are available for one drum, the spectral template is calculated separately for each of them and then averaged. The templates of all target drums are combined with (4) to create the spectral template matrix $\boldsymbol{S}$.

The time-varying gains are obtained from the spectrogram of the input signal by applying only the update formula (7) with the spectral templates until the result has converged. The detection of the drum hits from the resulting time-varying gains $\boldsymbol{a}_n$ is done by applying a psychoacoustically motivated onset detection on the gain curves. The process consists of μ-law compression, low-pass filtering, differentiation, half-wave rectification, detecting peaks, and thresholding the located peaks. The optimal threshold value is determined for each target drum using a set of training signals by minimizing the sum of extraneous and missed detections on them. The onset detection method is motivated by [9].

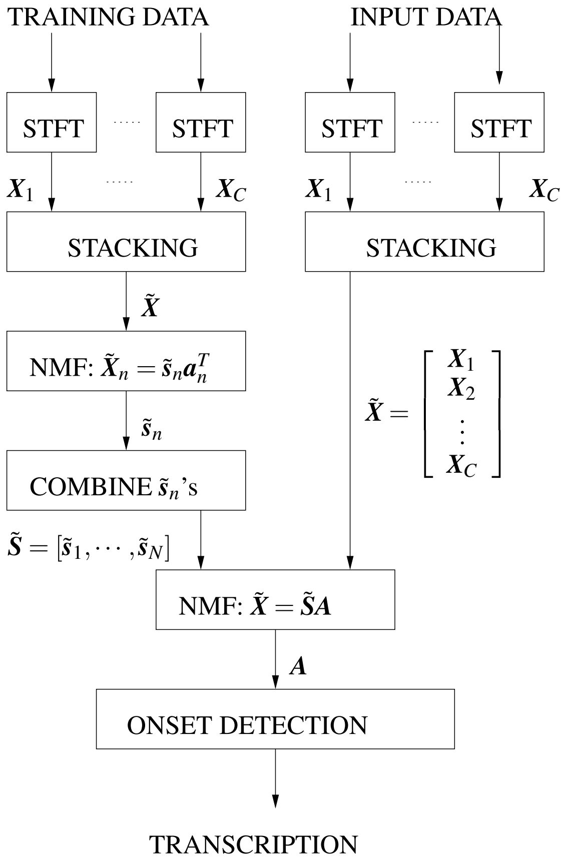


Figure 1: Functional overview of the proposed method. It calculates the spectral template $\tilde{\boldsymbol{s}}_n$ over all channels from signals containing only the target drum. The process is repeated for all target drums, and the resulting templates are combined to create the spectral template matrix $\tilde{\boldsymbol{S}}$. The templates are used to estimate the time-varying gains of the target drums played in the multichannel input. Finally, the sound event onsets are detected from the gains.

The single-channel NMF method has proven to perform well when the input data match the model well, i.e., there are only target drums in the mixture [12]. Additional drums and other sound sources caused the performance to degrade quickly. Furthermore, if the target drums overlap considerably in the frequency domain, the factorization may produce an undesired result.

2.2 Multichannel Input Data

We propose extending the single-channel method so that the factorized spectrogram is calculated not only from a single-channel mixture, but from the multiple microphone signals available in the studio environment. The spectrograms of C individual channels $\boldsymbol{X}_c$, $c \in 1 \ldots C$ are stacked to form the matrix to be factorized

$$\tilde{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \\ \vdots \\ \boldsymbol{X}_C \end{bmatrix}. \tag{9}$$

Now the factorization (3) is extended to

$$\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{S}}\boldsymbol{A}, \tag{10}$$

where the spectral templates are similarly stacked into

$$\tilde{\boldsymbol{S}} = \begin{bmatrix} \boldsymbol{S}_1 \\ \boldsymbol{S}_2 \\ \vdots \\ \boldsymbol{S}_C \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_{1,1}, \boldsymbol{s}_{1,2}, \cdots, \boldsymbol{s}_{1,N} \\ \boldsymbol{s}_{2,1}, \boldsymbol{s}_{2,2}, \cdots, \boldsymbol{s}_{2,N} \\ \vdots \cdots \ddots \vdots \\ \boldsymbol{s}_{C,1}, \boldsymbol{s}_{C,2}, \cdots, \boldsymbol{s}_{C,N} \end{bmatrix} = [\tilde{\boldsymbol{s}}_1, \tilde{\boldsymbol{s}}_2, \cdots, \tilde{\boldsymbol{s}}_N]. \tag{11}$$

In the matrix above, $\boldsymbol{s}_{c,n}$ is the spectral template of nth source in the microphone channel c. The main motivation for this stacking is that it allows to separate drums that have similar spectral content, e.g., tom-toms, if their intensity in different microphone channels differs. A functional overview of the proposed method is presented in Figure 1.

The training and use of the model is similar to the one of the single-channel method with small exceptions. The main difference is that the template $\tilde{\boldsymbol{s}}_n$ has to be calculated over all channels at once instead of constructing it by stacking the templates from individual channels. This is required to ensure that the relative level differences between the channels would be correct in the resulting template. The use of the template in the factorization and the determination of the onset instants is similar to the single-channel case.

3. EVALUATIONS

The performance of the proposed method was tested using a data set of multichannel drum recordings with manually made annotations [6]. The performance is compared with two other methods.

3.1 Acoustic Material

The material used was from the public audio subset of "ENST drums" database [6]. The database contains recordings for three different drummers and drum kits. Each drummer has a different number of instruments and a different microphone setup. The material consists of individual drum hits, traditional short drum sequences, and drum tracks played along accompaniment that are provided separately from the drum audio. These "minus one" tracks are further divided into tracks with acoustic and with MIDI accompaniment. For all of the material, the database contains individual microphone signals (7 or 8, depending on the kit), a "dry" mix-down with only level adjustments, and a "wet" mix-down with compressor and other effects. The evaluations use the individual drum hits for template calculation, and the "minus one" tracks for testing. In total, there are 64 "minus one" tracks in the data set with duration ranging from 30 s to 75 s and mean duration of 55 s.

3.2 Evaluation Process

The target drum set in the evaluations consists of bass drum (BD), snare drum (SD), and hi-hat (HH). Other drums in the tracks are not attempted to be transcribed. The target drum set is limited in this way since the three drums for the main rhythmic background on a large body of the pop/rock songs, while the other drums provide mainly accentuations. The input to the system consists of the close microphone signals and contains only drums sounds. The evaluations are done using leave-one-out cross-validation scheme on each of the three drum kits separately, and the presented results are calculated over all cross-validation folds. The cross-validation has to be made for each drummer subset separately because the method relies on the channel setup to remain identical across training and testing phases.

The spectrogram signal representation uses the frequency resolution of 24 Bark bands, and the analysis window length is 24 ms with 75% overlap.

3.3 Performance Metrics

The hits in the obtained result and in the ground truth are matched. Two hits were accepted as a match if they differ less than 50 ms in their onset times. Recall rate R_R is the ratio of correct hits to the hits in the ground truth, precision rate R_P the ratio of correct hits to the hits in the result, and F-measure is the harmonic mean of these $F = 2R_R R_P/(R_R + R_P)$.

3.4 Comparison to Other Systems

The performance of the proposed system is compared to other methods. First, the proposed multichannel version is compared with the single-channel NMF transcription method [12]. The single-channel data used in the experiments are the "dry-mix" versions. This comparison allows to determine if the use of multiple channels provides any additional information that the method can use compared to the mix-down.

Secondly, a naive multichannel transcription method is implemented. It assumes that each close microphone channel contains mostly the sound of that drum. The transcription can then be made by detecting sound event onsets on the channel signal. The onset detection is made with the method proposed in [9], and the detection threshold is determined using training data in a manner similar to the proposed method. The main weakness of this naive approach is that it relies on every target drum to have a close microphone.

Finally, a second single-channel comparison method is the one proposed in [7] attempting to enhance the drum sound content when there is also accompaniment involved. Even though the system was designed to transcribe drums with accompaniment, the original publication also provides evaluation results in the case when the accompaniment is not present. The evaluations in the original publication use the same data set and evaluation metric as this paper. These results are gathered in Table 2. When comparing the result, it should be remembered that the results presented in [7] were obtained using a cross-validation scheme differing from the one we are using, and used only single-channel input.

3.5 Results and Discussion

The evaluation results are presented in Tables 1-3. Table 3 presents the detailed results for the three different drum kit subsets for all of the "minus one" tracks. The results in Table 1 show that the naive approach works surprisingly well on the multichannel recordings. However, the multichannel approach performs even better. Both the single-channel

		BD	SD	HH	Total
1-channel [12]	R_R	93.7%	51.4%	77.1%	74.0%
	R_P	95.2%	80.0%	67.8%	78.5%
	F	94.4%	62.6%	72.1%	76.2%
onsets	R_R	85.7%	54.4%	83.5%	75.3%
	R_P	85.5%	76.8%	76.7%	79.4%
	F	85.6%	63.6%	80.0%	77.3%
m-channel	R_R	95.9%	77.8%	87.9%	87.1%
	R_P	93.5%	77.5%	78.7%	82.5%
	F	94.7%	77.6%	83.0%	84.7%

Table 1: Evaluation results for the proposed method (m-channel), the single-channel NMF (1-channel), and the naive method (onsets), on the "minus one" tracks of the ENST public dataset.

		BD	SD	HH
reference [7]	R_R	70.0%	64.2%	86.5%
	R_P	79.8%	71.0%	73.6%
	F	74.6%	67.4%	79.5%
1-channel [12]	R_R	94.1%	47.9%	70.6%
	R_P	95.1%	71.0%	63.4%
	F	94.6%	57.2%	66.8%
m-channel	R_R	96.0%	72.0%	84.0%
	R_P	93.6%	74.3%	75.5%
	F	95.0%	73.0%	79.4%

Table 2: Evaluation results for the proposed method, the single-channel NMF, and the reference method, on the "minus one" tracks excluding the ones recorded with MIDI accompaniment (the results for the reference are from [7]).

and naive multichannel method have a relatively low recall rate on snare drum. We assume that this may be caused by the "ghost hits", which are very light hits producing more full-bodied rhythmic feel. The lightness of these hits causes difficulties in setting the onset detection threshold. The difference in the recall and precision rates for hi-hat may be partially caused by the presence of the other cymbals in the tracks: since they do not belong to the target set and their spectral properties overlap with hi-hat, their presence causes some extra hits to be detected. More detailed inspection on the material revealed that a large body of the hi-hat insertions were caused by the cow bell instrument played in latin tracks in the place of hi-hat.

Based on the total F-measure of each individual target piece, the performance difference between the single-channel NMF and naive onset detection based method is not statistically significant. However, the overall performance increase obtained with the multichannel NMF method over the comparison methods is statistically significant with the level $p > 99\%$.

When the evaluations are done on the subset of "minus one" tracks played on a real accompaniment, the performance of both single-channel and multichannel NMF decrease, as can be seen in Table 2. Only the bass drum performance remains high. Comparing to the reference method [7], the proposed system is more accurate on bass drum, slightly more accurate on snare drum, and hi-hat performance is approximately even. Some of the performance degradation may be caused by the small amount of training data (only nine tracks for each drummer).

The per-drummer results in Table 3 show that there are some performance difference between the three subsets. The main improvement of the proposed method is visible in the snare drum results, where both recall and precision rates are greatly improved.

4. CONCLUSION

This paper has presented an extension of an NMF based drum transcription method to multichannel data. The multichannel data are available from the recording setup in a studio environment. The proposed method creates spectral templates for each target drum to each input channel, and uses NMF to recover the time-varying gains for the drums. Finally, onsets are searched from the recovered gains. The proposed system has been evaluated on multichannel data and compared to other methods in the task of transcribing bass drum, snare drum, and hi-hat. The results show that the use of multiple channels increases the system performance over the single-channel method.

REFERENCES

[1] M. A. Casey and A. Westner. Separation of mixed audio sources by independent subspace analysis. In *Proc. of International Computer Music Conference*, pages 154–161, Berlin, Aug. 2000.

[2] A. Cichocki, R. Zdunek, and S. Amari. Nonnegative matrix and tensor factorization. *IEEE Signal Processing Magazine*, pages 142–145, Jan. 2008.

[3] D. FitzGerald, M. Cranitch, and E. Coyle. Sound source separation using shifted non-negative tensor factorisation. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, May 2006.

[4] D. FitzGerald, B. Lawlor, and E. Coyle. Prior subspace analysis for drum transcription. In *Proc. of 114th Audio Engineering Society Convention*, Amsterdam, Mar. 2003.

[5] D. FitzGerald and J. Paulus. Unpitched percussion transcription. In A. Klapuri and M. Davy, editors, *Signal Processing Methods for Music Transcription*, pages 131–162. Springer, 2006.

[6] O. Gillet and G. Richard. ENST-Drums: an extensive audio-visual database for drum signal processing. In *Proc. of 7th International Conference on Music Information Retrieval*, Victoria, B.C., Canada, Oct. 2006.

[7] O. Gillet and G. Richard. Transcription and separation of drum signals from polyphonic music. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(3):529–540, Mar. 2008.

[8] M. Helén and T. Virtanen. Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine. In *Proc. of 13th European Signal Processing Conference*, Antalya, Turkey, Sept. 2005.

[9] A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, Ariz., USA, 1999.

[10] D. D. Lee and H. S. Seung. Algorithms for non-

		BD			SD			HH			Total		
		D1	D2	D3	D1	D2	D3	D1	D2	D3	D1	D2	D3
1-channel [12]	R_R	87.0%	97.5%	97.5%	47.7%	65.0%	41.4%	67.1%	80.4%	82.0%	67.1%	80.0%	74.9%
	R_P	94.6%	95.7%	95.2%	76.7%	82.3%	80.7%	55.9%	71.5%	74.6%	72.4%	80.5%	82.3%
	F	90.6%	96.6%	96.4%	58.8%	72.6%	54.7%	61.0%	75.7%	78.2%	69.7%	80.3%	78.4%
onsets	R_R	83.0%	86.4%	88.0%	73.5%	44.9%	43.8%	75.9%	82.8%	90.6%	77.4%	72.0%	76.4%
	R_P	75.6%	94.6%	89.7%	74.3%	73.8%	85.3%	77.0%	73.4%	79.7%	75.7%	78.9%	83.7%
	F	79.1%	90.3%	88.8%	73.9%	55.8%	57.9%	76.4%	77.8%	84.8%	76.5%	75.3%	79.9%
m-channel	R_R	93.2%	97.4%	97.4%	80.0%	77.8%	75.4%	86.0%	86.9%	90.4%	86.4%	86.8%	88.1%
	R_P	93.8%	95.3%	91.7%	77.4%	82.4%	73.0%	76.6%	77.1%	82.1%	81.9%	83.2%	82.3%
	F	93.5%	96.4%	94.5%	78.7%	80.1%	74.2%	81.1%	81.7%	86.0%	84.1%	85.0%	85.1%

Table 3: Evaluation results for the individual drummers (different drummers are denoted with D1, D2, and D3) for the same methods and material as presented in Table 1.

negative matrix factorization. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 556–562. MIT Press, 2001.

[11] J. Paulus and A. Klapuri. Combining temporal and spectral features in HMM-based drum transcription. In *Proc. of 8th International Conference on Music Information Retrieval*, pages 225–228, Vienna, Sept. 2007.

[12] J. Paulus and T. Virtanen. Drum transcription with non-negative spectrogram factorisation. In *Proc. of 13th European Signal Processing Conference*, Antalya, Turkey, Sept. 2005.

[13] P. Smaragdis and J. C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *Proc. of 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 177–180, New Platz, N.Y., USA, Oct. 2003.

[14] K. Yoshii, M. Goto, and H. G. Okuno. Drum sound recognition for polyphonic audio signals by adaptation and matching of spectrogram templates with harmonic structure suppression. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(1):333–345, Jan. 2007.

Publication P5

J. Paulus and A. Klapuri, “Drum sound detection in polyphonic music with hidden Markov models,” *EURASIP Journal on Audio, Speech, and Music Processing*. In press.

©2009 J. Paulus and A. Klapuri. Published under Creative Commons Attribution license.

Drum sound detection in polyphonic music with hidden Markov models

Jouni Paulus and Anssi Klapuri*
Department of Signal Processing
Tampere University of Technology
Korkeakoulunkatu 1, Tampere, Finland

Abstract

This paper proposes a method for transcribing drums from polyphonic music using a network of connected hidden Markov models (HMMs). The task is to detect the temporal locations of unpitched percussive sounds (such as bass drum or hi-hat) and recognise the instruments played. Contrary to many earlier methods, a separate sound event segmentation is not done, but connected HMMs are used to perform the segmentation and recognition jointly. Two ways of using HMMs are studied: modelling combinations of the target drums, and a detector-like modelling of each target drum.

Acoustic feature parametrisation is done with mel-frequency cepstral coefficients and their first order temporal derivatives. The effect of lowering the feature dimensionality with principal component analysis and linear discriminant analysis is evaluated. Because the acoustic properties of drum sounds may vary between the training and target signals, unsupervised acoustic model parameter adaptation with maximum likelihood linear regression is evaluated. The performance of the proposed method is evaluated on a publicly available data set containing signals with and without accompaniment (non-drum instruments) and compared with two reference methods. The results suggest that the transcription is possible using connected HMMs, and that using detector-like models for each target drum provides a better performance than modelling drum combinations.

Keywords: hidden Markov model, maximum likelihood linear regression, mel-frequency cepstral coefficient, linear discriminant analysis

1 Introduction

This paper applies connected hidden Markov models (HMMs) to the transcription of drums from polyphonic musical audio. For brevity, the word “drum” is here used to refer to all the unpitched percussions met in Western pop/rock music, such as bass drum, snare drum, and cymbals. The word “transcription” is used to refer to the process of locating drum sound onset instants and recognising the drums played. The analysis result enables several applications, such as using the transcription to assist beat tracking [11], drum track modification in the audio [30], reusing the drum patterns from existing audio, or musical studies on the played patterns.

Several methods have been proposed in the literature to solve the drum transcription problem. Following the categorisation made in [5] and [10], majority of the methods can be viewed to be either *segment and classify* or *separate and detect* approaches. The methods in the first category operate by segmenting the input audio into meaningful events, and then attempt to recognise the content of the segments. The segmentation can be done by detecting candidate sound onsets or by creating an isochronous temporal grid coinciding with most of the onsets. After the segmentation a set of features is extracted from each segment, and a classifier is employed to recognise the contents. The classification method varies from a naive Bayes classifier with Gaussian mixture models (GMMs) [19] to support vector machines (SVMs) [24, 10] and decision trees [21].

The methods in the second category aim at segregating each target drum into a separate stream and to detect sound onsets within the streams. The separation can be done with unsupervised methods like sparse coding [26] or independent subspace analysis (ISA) [25], but these

[*] This work was supported by the Academy of Finland, (application number 129657, Finnish Programme for Centres of Excellence in Research 2006–2011).

require recognising the instruments from the resulting streams. The recognition step can be avoided by utilising prior knowledge of the target drums in the form of templates, and applying a supervised source separation method. Combining ISA with drum templates produces a method called prior subspace analysis (PSA) [4]. PSA represents the templates as magnitude spectrograms and estimates the gains of each template over time. The possible negative values in the gains do not have a physical interpretation and require a heuristic post-processing. This problem was solved using non-negative matrix factorisation (NMF) restricting the component spectra and gains to be non-negative. This approach was shown to perform well when the target signal matches the model (signals containing only target drums) [18].

Some methods cannot be assigned to either of the categories above. These include *template matching and adaptation* methods operating with time-domain signals [33], or with a spectrogram representation [31].

The main weakness with the "segment and classify" methods is the segmentation. The classification phase is not able to recover any events missed in the segmentation without an explicit error correction scheme, e.g., [29]. If a temporal grid is used instead of onset detection, most of the events will be found, but the expressivity lying in the small temporal deviations from the grid is lost, and problems with the grid generation will be propagated to subsequent analysis stages.

To avoid making any decisions in the segmentation, this paper proposes to use a network of connected HMMs in the transcription in order to locate sound onsets and recognise the contents jointly. The target classes for recognition can be either combinations of drums or detectors for each drum. In the first approach, the recognition dictionary consists of combinations of target drums with one model to serve as the background model when no combination is played, and the task is to cover the input signal with these models. In the detector approach, each individual target drum is associated with two models: a "sound" model and a "silence" model, and the input signal is covered with these two models for each target drum independently from the others.

In addition to the HMM baseline system, the use of model adaptation with maximum likelihood linear regression (MLLR) will be evaluated. MLLR adapts the acoustic models from training to better match the specific input.

The rest of this article is organised as follows: Section 2 describes the proposed HMM-based transcription method, Section 3 details the evaluation setup and presents the obtained results, and finally Section 4 presents the conclusions of the paper. Parts of this work have been published earlier in [16] and [17].

2 Proposed method

Figure 1 shows an overview of the proposed method. The input audio is subjected to sinusoids-plus-residual modelling to suppress the effect of non-drum instruments by using only the residual. Then the signal is subdivided into short frames from which a set of features is extracted. The features serve as observations in HMMs that have been constructed in the training phase. The trained models are adapted with unsupervised maximum likelihood linear regression [12] to match the transcribed signal more closely. Finally, the transcription is done by searching an optimal path through the HMMs with Viterbi algorithm. The steps are described in more detail in the following.

2.1 Feature extraction and transformation

It has been noted, e.g., in [8] and [31], that suppression of tonal spectral components improves the accuracy of drum transcription. This is no surprise, as the common drums in pop/rock drum kit contain a notable stochastic component and relatively little tonal energy. Especially the idiophones (e.g., cymbals) produce mostly noise-like signal, while the membranophones (skinned drums) may contain also tonal components [6]. The harmonic suppression is here done with simple sinusoids-plus-residual modelling [15, 22]. The signal is subdivided into 92.9 ms frames, the spectrum is calculated with discrete Fourier transform, and 30 sinusoids with the largest magnitude are selected by locating the 30 largest local maxima in the magnitude spectrum. The sinusoids are then synthesised and the resulting signal is subtracted from the original signal. The residual serves as the input to the following analysis stages. Even though the processing may remove some of the tonal components of the membranophones, the remaining ones and the stochastic components are enough for the recognition. Preliminary experiments also suggest that the exact number of removed components is not important, even doubling the number to 60 caused only an insignificant drop in the performance.

The feature extraction calculates 13 mel-frequency cepstral coefficients (MFCCs) in 46.4 ms frames with

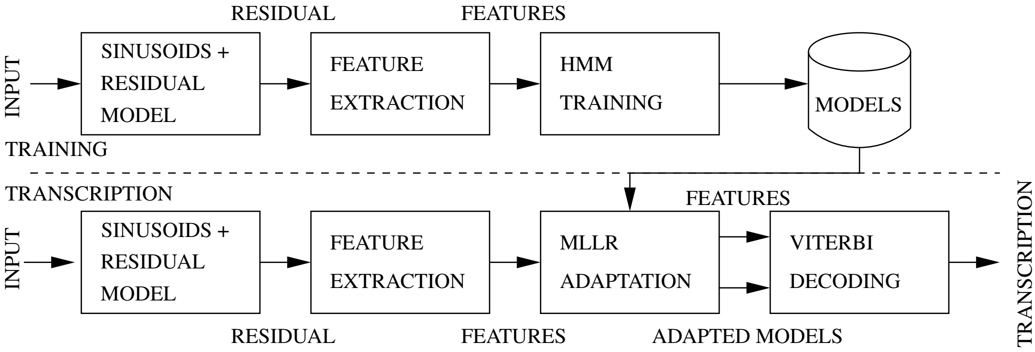


Figure 1: A block diagram of the proposed HMM transcription method including acoustic model adaptation.

75% overlap [1]. In addition to the MFCCs their first order temporal derivatives are estimated. The zeroth coefficient which is often discarded is also used. MFCCs have proven to work well in a variety of acoustic signal content analysis tasks including instrument recognition [2]. In addition to the MFCCs and their temporal derivatives, other spectral features, such as band energy ratios, spectral kurtosis, skewness, flatness, and slope used, e.g., in [24] were considered for the feature set. However, preliminary experiments suggested that their inclusion reduces the overall performance slightly and they are not used in the presented results. The reason for this degradation is an open question to be addressed in the future work, but is assumed that the features do not contain enough additional information compared to the original set to compensate the increased modelling requirements.

The resulting 26-dimensional feature vectors are normalised to have zero mean and unity variance in each feature dimension over the training data. Then the feature matrix is subjected to dimensionality reduction. Though unsupervised transformation with principal component analysis (PCA) has been successfully used in some earlier publications, e.g., [23], it did not perform well in our experiments. It is assumed that this is because PCA attempts only to describe the variance of the data without class information, and it may be distracted by the amount of noise present in the data.

The feature transformation used here is calculated with linear discriminant analysis (LDA). LDA is a class-aware transformation attempting to minimise intra-class scatter while maximising inter-class separation. If there are N different classes, LDA produces a transformation to $N-1$ feature dimensions.

2.2 HMM topologies

Two different ways to utilise connected HMMs for drum transcription are considered: drum sound combination modelling and detector models for each target drum. In the first case, each of the 2^M combinations of M target drums is modelled with a separate HMM. In the latter case, each target drum has two separate models: a "sound" model and a "silence" model. In both approaches the recognition aims to find a sequence of the models providing the optimal description of the input signal. Fig. 2 illustrates the decoding with combination modelling, while Fig. 3 illustrates the decoding with drumwise detectors.

The main motivation for the combination modelling is that in popular music multiple drums are often hit simultaneously. However, the main weakness is that as the number of target drums increases, the number of combinations to be modelled also increases rapidly. Since only the few most frequent combinations cover most of the occurrences, as illustrated in Fig. 4, there is very little training data for the more rare combinations. Furthermore, it may be difficult to determine whether or not some softer sound is present in a combination (e.g., when kick and snare drums are played, the presence of hi-hat may be difficult to detect from the acoustic information) and a wrong combination may be recognised.

With detector models, the training data can be utilised more efficiently than with combination models, because all combinations containing the target drum can be used to train the model. Another difference in the training phase is that each drum has a separate silence (or background) model.

As will be shown in Sec. 3, the detector topology generally outperforms the combination modelling which was found to have problems with overfitting the

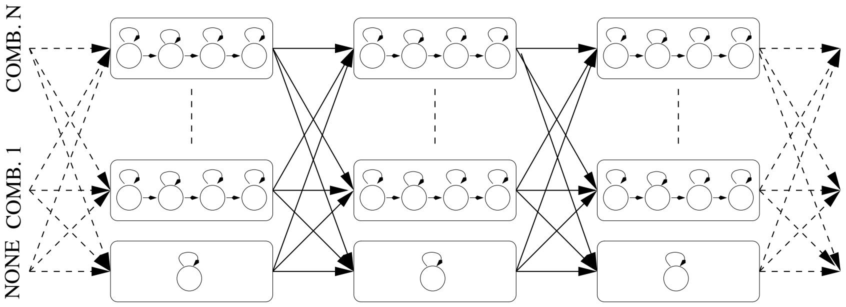


Figure 2: Illustration of the basic idea of drum transcription with connected HMMs for drum combinations. The decoding aims to find the optimal path through the models given the observed acoustic information.

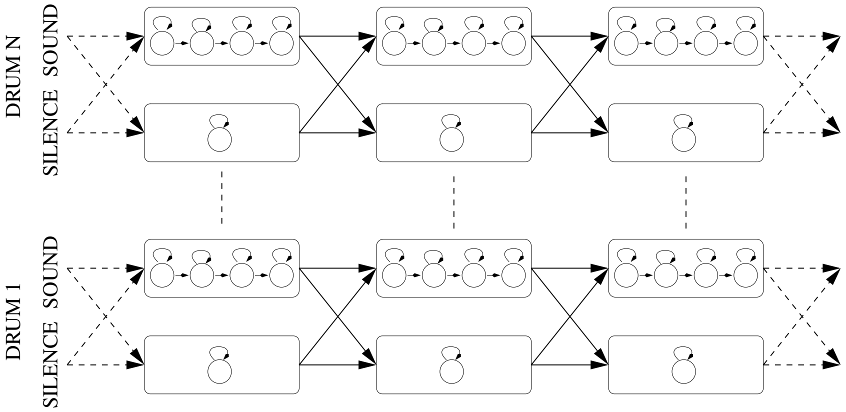


Figure 3: Illustration of the basic idea of drum transcription with HMM-based drum detectors. Each target drum is associated with two models, “sound” and “silence”, and the decoding is done for each drum separately.

limited amount of training data. This was indicated by the following observations: performance degradation with increasing the number of HMM training iterations and acoustic adaptation, and slight improvement in the performance with simpler models and reduced feature dimensions. Because of this, the results on acoustic model adaptation and feature transformations is presented only for the detector topology (similar choice has been done, e.g., in [10]). For the sake of comparison, however, results are reported also for the combination modelling baseline.

The sound models consist of a four-state left-to-right HMM where a transition is allowed to the state itself and to the following state. The observation likelihoods are modelled with single Gaussian distributions. The silence model is a single-state HMM with a 5-component GMM for the observation likelihoods. This topology was chosen because the background sound does not have a clear sequential form. The number of states and GMM components were empirically determined.

The models are trained with expectation maximisation algorithm [20] using segmented training examples. The segments are extracted after annotated event onsets using a maximum duration of 10 frames. If there is another onset closer than the set limit, the segment is truncated accordingly. In detector modelling, the training instances for the “sound” model are generated from the segments containing the target drum, and the remaining frames are used to train the “silence” model. In combination modelling, the training instances for each combination are collected from the data, and the remaining frames are used to train the background model.

2.3 Acoustic adaptation

Unsupervised acoustic adaptation with maximum likelihood linear regression (MLLR) [12] has been successfully used to adapt the HMM observation density parameters, e.g., in adapting speaker independent models to speaker dependent models in speech recogni-

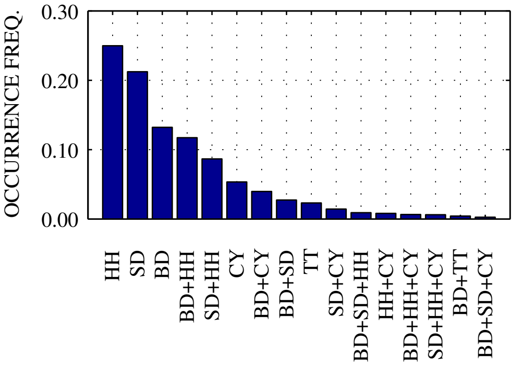


Figure 4: Relative occurrence frequencies of various drum combinations in "ENST drums" [9] data set. Different drums are denoted with BD (bass drum), CY (all cymbals), HH (all hi-hats), SD (snare drum), and TT (all tom-toms). Two drum hits were defined to be simultaneous if their annotated onset times differ less than 10 ms. Only the 16 most frequent combinations are shown.

tion [12], language adaptation from Spanish to Valencian [13], or to utilise a recognition database trained for phone speech to recognise speech in car conditions [3]. The motivation for using MLLR here is that it is assumed that the acoustic properties of the target signal always differ from those of the training data, and the match between the model and the observations can be improved with adaptation. The adaptation is done for each target signal independently to provide models that fit the specific signal better. The adaptation is evaluated only for the detector topology, because for drum combinations, the adaptation was not successful, most likely due to the limited amount of observations.

In single variable MLLR for the mean parameter, a transformation matrix

$$\boldsymbol{W} = \begin{pmatrix} w_{1,1} & w_{1,2} & 0 & \dots & 0 \\ w_{2,1} & 0 & w_{2,3} & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ w_{n,1} & 0 & \dots & 0 & w_{n,n+1} \end{pmatrix} \tag{1}$$

is used to apply a linear transformation to the GMM mean vector μ so that the likelihood of the adaptation data is maximised. The mean vector μ with the length n is transformed by

$$\mu' = \boldsymbol{W}[\omega, \mu^{\mathsf{T}}]^{\mathsf{T}}, \tag{2}$$

where the transformation matrix has the dimensions of $n \times (n+1)$, and $\omega = 1$ is a bias parameter. The non-zero elements of $\boldsymbol{W}$ can be organised into a vector

$$\hat{\mathbf{w}} = [w_{1,1}, \dots, w_{n,1}, w_{1,2}, \dots, w_{n,n+1}]^{\mathsf{T}}. \tag{3}$$

The value of the vector can be calculated by

$$\hat{\mathbf{w}} = \left[\sum_{s=1}^{S} \sum_{t=1}^{T} \gamma_s(t) \boldsymbol{D}_s^{\mathsf{T}} \boldsymbol{C}_s^{-1} \boldsymbol{D}_s \right]^{-1} \left[\sum_{s=1}^{S} \sum_{t=1}^{T} \gamma_s(t) \boldsymbol{D}_s^{\mathsf{T}} \boldsymbol{C}_s^{-1} \mathbf{o}(t) \right], \tag{4}$$

where t is frame index, $\mathbf{o}(t)$ is the observation vector from frame t, s is an index of GMM components in the HMM, $\boldsymbol{C}_s$ is the covariance matrix of GMM component s, $\gamma_s(t)$ the occupation probability of s^{th} component in frame t (calculated, e.g., with the forward-backward algorithm), and matrix $\boldsymbol{D}_s$ is defined as a concatenation of two diagonal matrices

$$\boldsymbol{D}_s = [\boldsymbol{I}\omega, \operatorname{diag}(\mu_s)], \tag{5}$$

where μ_s is the mean vector of the s^{th} component and $\boldsymbol{I}$ is a $n \times n$ identity matrix [12]. In addition to the single variable mean transformation, also full matrix mean transformation [12] and variance transformation [7] were tested. In the evaluations, the single variable adaptation performed better than the full matrix mean transformation, and therefore the results are presented only for it. Variance transformation reduced performance in all cases.

The adaptation is done so that the signal is first analysed with the original models. Then it is segmented to examples of either class ("sound" / "silence") based on the recognition result, and the segments are used to adapt the corresponding models. The adaptation can be repeated using the models from the previous adaptation iteration for segmentation. It was found in the evaluations that applying the adaptation repeatedly for three times produced the best result even though the obtained improvement after the first adaptation was usually very small. Further increment of the number of adaptation iterations from this started to degrade the results.

2.4 Recognition

In the recognition phase, the (adapted) HMM models are combined into a larger compound model, see Figs. 2 and 3. This is done by concatenating the state transition matrices of the individual HMMs and incorporating the inter-model transition probabilities in the same matrix. The transition probabilities between the models are estimated from the same material that is used for training the acoustic models, and the bigram probabilities are smoothed with Witten-Bell smoothing [28]. The compound model is then used to decode the sequence with Viterbi algorithm. Another alternative would be to use token passing algorithm [32], but since the model satisfies the first order Markov assumption (only bigrams are used), Viterbi is still a viable alternative.

3 Results

The performance of the proposed method is evaluated using the publicly available data set "ENST drums" [9]. The data set allows adjusting the accompaniment (everything else but the drums) level in relation to the drum signal, and two different levels are used in the evaluations: a balanced mix and a drums-only signal. The performance of the proposed method is compared with two reference systems: a "segment and classify" method by Tanghe et al. [24], and a supervised "separate and detect" method using non-negative matrix factorisation [18].

3.1 Acoustic data

The data set "ENST drums" contains multichannel recordings of three drummers playing with different drum kits. In addition to the original multichannel recordings, also two downmixes are provided: "dry" with minimal effects, mainly having only the levels of different drums balanced, and "wet" resembling the drum tracks on commercial recordings, containing some effects and compression. The material in the data set ranges from individual hits to stereotypical phrases, and finally to longer tracks played along with an accompaniment. These "minus one" tracks played on accompaniment have the synchronised accompaniment available as a separate signal allowing to create polyphonic signals with custom mixing levels. The ground truth for the data set contains the onset times for the different drums, and was provided with the data set.

The "minus one" tracks are used as the evaluation data. They are naturally split into three subsets based on the player and kit, each having approximately the same number of tracks (two with 21 tracks and one with 22). The lengths of the tracks range from 30 s to 75 s with mean duration of 55 s. The mixing ratios of drums and accompaniment used in the evaluations are drums-only and a "balanced" mix. The former is used to obtain a baseline result for the system with no accompaniment. The latter, corresponding to applying scaling factors of $2/3$ for the drum signal and $1/3$ for the accompaniment, is used then to evaluate the system performance in realistic conditions met in polyphonic music.[1]

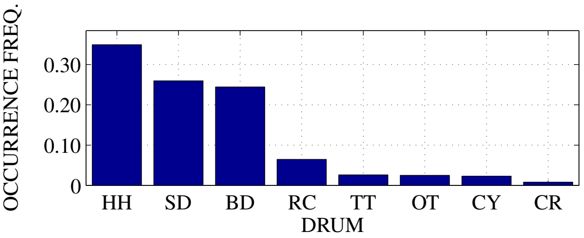


Figure 5: Occurrence frequencies of different drums in "ENST drums" data set. The instruments are denoted by: BD (bass drum), CR (all crash cymbals), CY (other cymbals), HH (open and closed hi-hat), RC (all ride cymbals), SD (snare drum), TT (all tom-toms), and OT (other unpitched percussion instruments, e.g., cow bell).

3.2 Evaluation setup

Evaluations are run using a three-fold cross-validation scheme. Data from two drummers are used to train the system and the data from the third are used for testing, and the division is repeated three times. This setup guarantees that the acoustic models have not seen the test data and their generalisation capability will be tested. In fact, the sounds of the corresponding drums in different kits may differ considerably (for example, depending on the tension of the skin, the use of muffling in case of kick drum, or the instrument used to hit the drum that can be a mallet, a stick, rods, or brushes) and using only two examples of a certain drum category to recognise a third one is a difficult problem. Hence, in real applications the training should be done with as diverse data as possible.

The target drums in the evaluations are bass drum (BD), snare drum (SD), and hi-hat (HH). The target set is limited to these three for two main reasons. First, they are found practically in every track in the evaluation data and they cover a large portion of all the drum sound events, as can be seen from Fig. 5. Secondly, and more importantly, these three instruments convey the

[1] The mixing levels are based on personal communication with O. Gillet, and result into an average of -1.25 dB drums-to-accompaniment ratio over the whole data set.

main rhythmic feel of most of the popular music songs, and occur in a relatively similar way in all the kits.

In the evaluation of the transcription result, the found target drum onset locations are compared with the locations given in the ground truth annotation. The hits are matched to the closest hit in the other set so that each hit has at most one hit associated to it. A transcribed onset is accepted as correct if the absolute time difference to the ground truth onset is less than 30 ms.[2] When the number of events is G in the ground truth and E in the transcription result, and the number of missed ground truth events and inserted events are m and i respectively, the transcription performance can be described with precision rate

$$P = (E - i)/E \tag{6}$$

and recall rate

$$R = (G - m)/G. \tag{7}$$

These two metrics can be further summarised by their harmonic mean, F-measure

$$F = (2PR)/(P + R). \tag{8}$$

3.3 Reference methods

The system performance is compared with two earlier methods: a "segment and classify" method by Tanghe et al. [24], and a "separate and detect" method by Paulus and Virtanen [18]. The former, referred to as *SVM* in the results, was designed for transcribing drums from polyphonic music by detecting sound onsets and then classifying the sounds with binary SVMs for each target drum. An implementation of the original author is used [14]. The latter, referred to as *NMF-PSA*, was designed for transcribing drums from a signal without accompaniment. The method uses spectral templates for each target drum and estimates their time-varying gains using NMF. Onsets are detected from the recovered gains. Also here the original implementation is used. The models for the SVM method are not trained specifically for the data used, but the generic models provided are used instead. The spectral templates for NMF-PSA are calculated from the individual drum hits in the data set used here. In the original publication the mid-level representation used spectral resolution of five bands. Here they are replaced with 24 Bark bands for improved frequency resolution.

Table 1: Evaluation results for the tested methods using the balanced drums and accompaniment mixture as input.

Method	Metric	BD	SD	HH	Total
HMM	$P(\%)$	84.7	65.3	84.9	80.0
	$R(\%)$	77.4	44.9	78.5	68.0
	$F(\%)$	80.9	53.2	81.6	**73.5**
HMM+MLLR	$P(\%)$	80.2	66.3	84.7	79.0
	$R(\%)$	81.5	45.3	82.6	70.9
	$F(\%)$	80.8	53.9	83.6	**74.7**
HMM comb	$P(\%)$	54.9	38.8	73.0	55.0
	$R(\%)$	66.4	47.0	58.7	57.4
	$F(\%)$	60.1	42.5	65.1	**56.1**
NMF-PSA [18]	$P(\%)$	69.9	57.0	58.2	62.0
	$R(\%)$	57.9	16.7	53.5	43.6
	$F(\%)$	63.4	25.9	55.8	**51.2**
SVM [24]	$P(\%)$	80.9	65.9	47.1	54.3
	$R(\%)$	38.4	14.2	69.5	43.8
	$F(\%)$	51.1	23.4	56.1	**48.5**

3.4 Results

The evaluation results are given in Tables 1 and 2. The former contains the evaluation results in the case of the "balanced" mixture as the input, while the latter contains the results for signals without accompaniment. The methods are referred to as

- *HMM*: The proposed HMM method with detectors for each target drum without acoustic adaptation.
- *HMM+MLLR*: The proposed detector-like HMM method including the acoustic model adaptation with MLLR.
- *HMM comb*: The proposed HMM method with drum combinations without acoustic adaptation.
- *NMF-PSA*: A "separate and detect" method using NMF for the source separation, proposed in [18].
- *SVM*: A "segment and classify" method proposed in [24] using SVMs for detecting the presence of each target drum in the located segments.

The results show that the proposed method performs best among the evaluated methods. In addition, it can be seen that the acoustic adaptation slightly improves the recognition result. All the evaluated methods seem to have problems in transcribing the snare drum (SD),

[2] When comparing the results obtained with the same data set in [10], it should be noted that there the allowed deviation was 50 ms.

Table 2: Evaluation results for the tested methods using signals without any accompaniment as input.

Method	Metric	BD	SD	HH	Total
HMM	P(%)	95.7	68.7	82.7	82.5
	R(%)	88.1	57.7	80.9	75.9
	F(%)	91.8	62.7	81.8	**79.1**
HMM+MLLR	P(%)	94.1	75.0	83.8	84.8
	R(%)	92.1	56.7	84.9	78.4
	F(%)	93.1	64.6	84.4	**81.5**
HMM comb	P(%)	71.5	41.3	63.8	57.5
	R(%)	74.2	54.3	55.3	60.4
	F(%)	72.8	46.9	59.3	**58.9**
NMF-PSA [18]	P(%)	85.0	75.6	57.1	68.5
	R(%)	80.1	38.1	67.7	62.2
	F(%)	82.5	50.7	61.9	**65.2**
SVM [24]	P(%)	95.4	62.9	61.1	68.2
	R(%)	54.0	37.9	72.3	56.6
	F(%)	69.0	47.3	66.2	**61.9**

even without the presence of accompaniment. One reason for this is that the snare drum is often played in more diverse ways than, e.g., the bass drum. Examples of these include producing the excitation with sticks or brushes, or playing with and without the snare belt, or by producing barely audible "ghost hits".

When analysing the results of "segment and classify" methods it is possible to distinguish between errors in segmentation and classification. However, since the proposed method aims to perform these tasks jointly, acting as a specialised onset detection method for each target drum, this distinction cannot be made.

An earlier evaluation with the same data set was presented in [10, Table II]. The table section "Accompaniment $+0$ dB" in there corresponds to the results presented in Table 1 and section "Accompaniment $-\infty$ dB" corresponds to the results in Table 2. In both cases, the proposed method clearly outperforms the earlier method in bass drum and hi-hat transcription accuracy. However, the performance of the proposed method on snare drum is slightly worse.

The improvement obtained using the acoustic model adaptation is relatively small. Measuring the statistical significance with two-tailed unequal variance Welch's t-test [27] on the F-measures for individual test signals produces p-value of approximately 0.64 for the balanced mix test data and 0.18 for the data without accompaniment suggesting that the difference in the results is not statistically significant. However, the adaptation seems to provide a better balance on precision and recall rates. The performance differences between the proposed detector-like HMMs and the other methods are clearly in favour of the proposed method.

Table 3: Effect of feature transformation on overall F-measure (%) of detector HMMs without acoustic model adaptation.

	none	PCA 90%	LDA
Plain drums	63.6	66.0	79.1
Balanced mix	59.6	60.9	73.5

Table 3 provides the evaluation results with different feature transformation methods while using detector-like HMMs without acoustic adaptation. The results show that PCA has a very small effect on the overall performance while LDA provides a considerable improvement.

4 Conclusions

This paper has studied and evaluated different ways of using connected HMMs for transcribing drums from polyphonic music. The proposed detector-type approach is relatively simple with only two models for each target drum: a "sound" and a "silence" model. In addition, modelling of drum combinations instead of detectors for individual drums was investigated, but found not to work very well. It is likely that the problems with the combination models are caused by overfitting the training data. The acoustic front-end extracts mel-frequency cepstral coefficients (MFCCs) and their first order derivatives to be used as the acoustic feature. Comparison of feature transformations suggests that LDA provides a considerable performance increase with the proposed method. Acoustic model adaptation with MLLR is tested, but the obtained improvement is relatively small. The proposed method produces a relatively good transcription of bass drum and hi-hat, but snare drum recognition has some problems that need to be addressed in future work. The main finding is that it is not necessary to have a separate segmentation step in a drum transcriber, but the segmentation and recognition can be performed jointly with an HMM even in the presence of accompaniment and with bad signal-to-noise ratios.

References

[1] S. B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):357–366, Aug. 1980.

[2] A. Eronen. Comparison of features for musical instrument recognition. In *Proc. of 2001 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 19–22, New Platz, N.Y., USA, Oct. 2001.

[3] A. Fischer and V. Stahl. Database and online adapatation for improved speech recognition in car environments. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 445–448, Phoenix, Ariz., USA, 1999.

[4] D. FitzGerald, B. Lawlor, and E. Coyle. Prior subspace analysis for drum transcription. In *Proc. of 114th Audio Engineering Society Convention*, Amsterdam, The Netherlands, Mar. 2003.

[5] D. FitzGerald and J. Paulus. Unpitched percussion transcription. In A. Klapuri and M. Davy, editors, *Signal Processing Methods for Music Transcription*, pages 131–162. Springer, 2006.

[6] N. H. Fletcher and T. D. Rossing. *The Physics of Musical Instruments*. Springer-Verlag New York Inc., New York, N.Y., USA, second edition, 1998.

[7] M. J. F. Gales, D. Pye, and P. C. Woodland. Variance compensation within the MLLR framework for robust speech recognition and speaker adaptation. In *Proc. of the Fourth International Conference on Spoken Language Processing*, pages 1832–1835, Philadelphia, Pa., USA, Oct. 1996.

[8] O. Gillet and G. Richard. Drum track transcription of polyphonic music using noise subspace projection. In *Proc. of 6th International Conference on Music Information Retrieval*, pages 156–159, London, England, UK, Sept. 2005.

[9] O. Gillet and G. Richard. ENST-Drums: an extensive audio-visual database for drum signal processing. In *Proc. of 7th International Conference on Music Information Retrieval*, pages 156–159, Victoria, B.C., Canada, Oct. 2006.

[10] O. Gillet and G. Richard. Transcription and separation of drum signals from polyphonic music. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(3):529–540, Mar. 2008.

[11] M. Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30(2):159–171, 2001.

[12] J. Leggetter, C and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, 9:171–185, 1995.

[13] M. Luján, C. D. Martínez, and V. Alabau. Evaluation of several maximum likelihood linear regression variants for language adaptation. In *Proc. of Sixth International Language Resources and Evaluation Conference*, Marrakech, Morocco, May 2008.

[14] MAMI. Musical audio-mining, drum detection console applications, 2005. `http://www.ipem.ugent.be/MAMI/`.

[15] R. J. McAulay and F. Quatieri, Thomas. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34(4):744–754, Aug. 1986.

[16] J. Paulus. Acoustic modelling of drum sounds with hidden Markov models for music transcription. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 241–244, Toulouse, France, May 2006.

[17] J. Paulus and A. Klapuri. Combining temporal and spectral features in HMM-based drum transcription. In *Proc. of 8th International Conference on Music Information Retrieval*, pages 225–228, Vienna, Austria, Sept. 2007.

[18] J. Paulus and T. Virtanen. Drum transcription with non-negative spectrogram factorisation. In *Proc. of 13th European Signal Processing Conference*, Antalya, Turkey, Sept. 2005.

[19] J. K. Paulus and A. P. Klapuri. Conventional and periodic N-grams in the transcription of drum sequences. In *Proc. of IEEE International Confer-*

ence on Multimedia and Expo, volume 2, pages 737–740, Baltimore, Md., USA, July 2003.

[20] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–289, Feb. 1989.

[21] V. Sandvold, F. Gouyon, and P. Herrera. Percussion classification in polyphonic audio recordings using localized sound models. In *Proc. of 5th International Conference on Music Information Retrieval*, pages 537–540, Barcelona, Spain, Oct. 2004.

[22] X. Serra. Musical sound modeling with sinusoids plus noise. In C. Roads, S. Pope, A. Picialli, and G. De Poli, editors, *Musical Signal Processing*, pages 91–122. Swets & Zeitlinger, 1997.

[23] P. Somervuo. Experiments with linear and nonlinear feature transformations in HMM based phone recognition. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume I, pages 52–55, Hong Kong, 2003.

[24] K. Tanghe, S. Dengroeve, and B. De Baets. An algorithm for detecting and labeling drum events in polyphonic music. In *Proc. of First Annual Music Information Retrieval Evaluation eXchange*, London, England, UK, Sept. 2005. Extended abstract.

[25] C. Uhle, C. Dittmar, and T. Sporer. Extraction of drum tracks from polyphonic music using independent subspace analysis. In *Proc. of 4th International Symposium on Independent Component Analysis and Blind Signal Separation*, pages 843–848, Nara, Japan, Apr. 2003.

[26] T. Virtanen. Sound source separation using sparse coding with temporal continuity objective. In *Proc. of International Computer Music Conference*, pages 231–234, Singapore, Oct. 2003.

[27] B. A. Welch. The generalization of "Student's" problem when several different population variances are involved. *Biometrika*, 34(1/2):28–35, Jan. 1947.

[28] I. H. Witten and T. C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transcations on Information Theory*, 37(4):1085–1094, 1991.

[29] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno. An error correction framework based on drum pattern periodicity for improving drum sound detection. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 237–240, Toulouse, France, May 2006.

[30] K. Yoshii, M. Goto, and H. G. Okuno. INTER:D: A drum sound equalizer for controlling volume and timbre of drums. In *Proc. of 2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies*, pages 205–212, London, England, UK, Nov. 2005.

[31] K. Yoshii, M. Goto, and H. G. Okuno. Drum sound recognition for polyphonic audio signals by adaptation and matching of spectrogram templates with harmonic structure suppression. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(1):333–345, Jan. 2007.

[32] S. J. Young, N. H. Russell, and J. H. S. Thornton. Token passing: a simple conceptual model for connected speech recognition systems. Technical Report CUED/F-INFENG/TR38, Cambridge University Engineering Department, Cambridge, UK, July 1989.

[33] A. Zils, F. Pachet, O. Delerue, and F. Gouyon. Automatic extraction of drum tracks from polyphonic music signals. In *Proc. of 2nd International Conference on Web Delivering of Music*, pages 179–183, Darmstadt, Germany, Dec. 2002.

Publication P6

J. Paulus and A. Klapuri, “Music structure analysis by finding repeated parts,” in *Proc. of 1st ACM Audio and Music Computing Multimedia Workshop*, Santa Barbara, Calif., USA, Oct. 2006, pp. 59–68. doi: 10.1145/1178723.1178733

©ACM, (2006) This is the author’s version of the work. it is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in *Proc. of 1st ACM Audio and Music Computing Multimedia Workshop*, Santa Barbara, Calif., USA, Oct. 2006, pp. 59–68. http://doi.acm.org/10.1145/1178723.1178733

Music Structure Analysis by Finding Repeated Parts

Jouni Paulus
jouni.paulus@tut.fi

Anssi Klapuri
anssi.klapuri@tut.fi

Institute of Signal Processing
Tampere University of Technology
Korkeakoulunkatu 1, Tampere, Finland

ABSTRACT

The structure of a musical piece can be described with segments having a certain time range and a label. Segments having the same label are considered as occurrences of a certain structural part. Here, a system for finding structural descriptions is presented. The problem is formulated in terms of a cost function for structural descriptions. A method for creating multiple candidate descriptions from acoustic input signal is presented, and an efficient algorithm is presented to find the optimal description with regard to the cost function from the candidate set. The analysis system is evaluated with simulations on a database of 50 popular music pieces.

Categories and Subject Descriptors

H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing—*Signal analysis, synthesis, and processing*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Abstracting methods*

General Terms

Algorithms, Theory

Keywords

music structure, structure analysis, segmentation, cost function, search algorithm, structure comparison

1. INTRODUCTION

Many musical pieces, especially in the popular music genre, consist of distinguishable parts that may repeat. The structure can be, e.g., "intro, verse, chorus, verse, chorus, chorus". Structure analysis aims to determine this kind of a description for a musical piece. Depending on the method applied, the result may consist of the temporal locations the parts and arbitrary labels assigned to them (e.g., "A, B, C, B, C, C"), whereas some systems also try to label the parts meaningfully. The information about the temporal locations of these parts and their labels form the *structural description* of the piece.

Automatic analysis of the structure has been studied mainly for the application of creating a meaningful summary of a musical piece. One of the first works operating on acoustic signals was by Logan and Chu, describing an agglomerative clustering and hidden Markov model (HMM) based approaches for key phrase generation [19]. They used mel-frequency cepstral coefficients (MFCCs) from short (26 ms), overlapping frames. The clustering method grouped the frames together iteratively until a level of stability had been reached. In the HMM method they trained an ergodic HMM with only few states, hoping that each state would represent a musical part, and used the Viterbi decoded state sequence as the description of the musical structure. The HMM approach was taken further by Aucouturier and Sandler using spectral envelope as the feature [2]. It was noted in both these studies that when using such short frames, the HMM states did not model musically meaningful parts, as was hoped. Abdallah et al increased the frame length considerably and the number of states up to 80 [1]. After acquiring the state sequence, each frame was provided with some knowledge about the surrounding context by calculating a state histogram in a 15 frame window. The histograms were then used in clustering the frames by optimising a cost function with simulated annealing. Rhodes et al added a term to control the duration of stay in a certain cluster [23], while Levy et al refined the clustering method to a context-aware variant of fuzzy C-means [18].

Another popular starting point of the analysis is to calculate frame-by-frame similarities over the whole signal, constructing a self-similarity matrix. Foote proposed to use the similarity matrix for visualising music [9]. It was noted that the parts of music having similar timbral characteristics created visible areas in the similarity matrix. The borders of these areas were sought and used in segmenting the piece in [10]. In [11] Foote and Cooper used a spectral clustering method to group similar segments.

When the used feature describes the tonal (pitch) content of the signal instead of general timbre, e.g., chroma instead of MFCCs, repetitions generate off-diagonal stripes to the similarity matrix instead of rectangular areas of high similarity. Such stripes reveal similar sequential structures, e.g., melody lines or chord progressions, instead of just denoting parts having similar timbral characteristics, or sounding the same. The two main approaches (HMM-based "state" method and "sequence" method relying on stripes in the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
AMCMM'06, October 27, 2006, Santa Barbara, California, USA.
Copyright 2006 ACM 1-59593-501-0/06/0010 ...$5.00.

similarity matrix) were compared by Peeters [22]. He noted that as the sequence approach requires a part to occur at least twice to be found, the HMM approach would be more robust analysis method. Still, the stripes have been used in structure analysis by several authors. Bartsch and Wakefield extracted chroma from beat-synchronised frames and used the most prominent off-diagonal stripe to define a thumbnail for the piece [3]. Lu et al proposed a distance metric considering the harmonic content of sounds, and used 2D morphological operations (erosion and dilation) to enhance the stripes [20]. In popular music pieces, the clearest repeated part is often the chorus section. Goto aimed at detecting it using chroma, and presented a method for handling the musical key modulation sometimes taking place in the last in the last refrain of the piece [12].

Music tends to show repetition and similarities on different levels, starting from consecutive bars to larger parts like chorus and verse. Some authors have tried to take this into account and proposed methods operating on several temporal levels. Jehan constructed several hierarchically related similarity matrices [16]. Shiu et al extracted chroma from beat-synchronised frames and then used dynamic time warping (DTW) to calculate a similarity matrix between all the measures of the piece [24]. The higher level musical structure was then modelled with a manually parametrised HMM. Dannenberg and Hu gathered the shorter repeated parts and gradually combined them to create longer, more meaningful, parts in [8]. Later, Dannenberg used the stripes in similarity matrix to find similar musical sections, and then utilised this information to aid a beat tracker [7].

Chai proposed to take the context into account by matching two windows of frame level features with DTW. Sliding the other window while keeping the other fixed provided a method to calculate the similarity on different lags and to determine the lag of maximum similarity. Gathering this information in a matrix formed stripes of prominent lags, like the stripes in a similarity matrix. The longer stripes were then interpreted information about the repeats of structural parts [6].

Maddage et al proposed a method for analysing a musical piece combining different sources of information. They used beat-synchronised pitch class profile as the feature and detected chords with pre-trained HMMs. Using assumptions of the lengths of the repeated parts, fixed length segments were matched to get a measure of similarity. Finally heuristic rules, claimed to apply on English-language pop songs, were used to deduce the high-level structure of the piece. [21]

Like in some of the earlier publications, we are interested in the structure defined by parts that are repeated. In other words, the proposed analysis method is not interested, nor able, to find musical parts that occur only once during the piece. These remain as unexplained segments in between the repeating parts and in principle they can be labelled, but no substructure is imposed on them. This limits the information that can be extracted from music, but as mentioned, e.g., in [9, 7], musical pieces usually have a structure relying to some extent on repetitions of different parts.

This paper focuses on estimating the musical structure from the result of low-level signal analysis front-end. Formulation of the problem is given in Section 2.1. Then we propose a parametric cost function for evaluating the fitness of an description. Depending on the parameter values it allows emphasising different properties: the complexity of the description, amount of the piece left unexplained, and consistency of the found parts. The parameters provide a way of circumventing the ambiguity of the structural description by allowing controlling the desired properties of the result. In Section 2.5, we propose a computationally efficient algorithm for searching the structural description that minimises the cost function by solving the computationally expensive combinatorial problem.

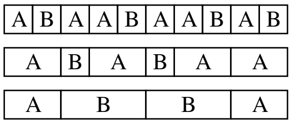


Figure 1: A simple example of three different structural descriptions for the same piece. All three are valid from the point of view of detecting repetitions.

The structural descriptions found by the system are evaluated using a database of 50 musical pieces with manually annotated structural information. The evaluation procedure and results are presented in Section 3.

As stated, e.g., in [5], the level of structural description may be different depending on the algorithm, person in question, etc. An example of the level differences on the descriptions is illustrated in Figure 1. The description on the top is the most detailed and gives the largest amount of structural information. However, that level of detail is not always needed, so the description on middle or on bottom might be adequate. Even though they do not contain so fine details, the coarse structure of the piece is more readily visible. The proposed analysis algorithm discards all hierarchical information. All operations changing the hierarchical level, i.e., combining parts always occurring in conjunction or splitting longer segments, is left for an optional post-processing stage.

2. PROPOSED METHOD

A block diagram of the proposed system can be seen in Figure 2. The front-end signal analysis consists of methods that have been utilised earlier and found to be working relatively well (musical meter analysis [17], chroma extraction [12] with multi-octave modifications, beat synchronised feature extraction [3], measure-level similarity matrix [24], signal segment border generation from timbral novelty [10], and segment matching with DTW [5]).

The front-end analysis provides features and initial set of candidate section boundaries (see Section 2.3) for the latter stages. The proposed method then creates longer segments from the blocks, and finds the optimal description of the piece with non-overlapping repeated parts in respect to the cost function defined in Section 2.1.

2.1 Defining a "Good" Structural Description

The musical piece to be described is subdivided into *blocks* $b_1, b_2, \ldots, b_N$, for example, at the times of bar lines. Each block is described uniquely by its start and end times, $b_j = (\tau_j, t_j)$, where τ_j is the start time and t_j is the end time. A *segment* s_k is a part of the piece consisting of one or more consecutive blocks, $s_k = b_i, b_{i+1}, \ldots, b_j$, i.e., an occurrence of a structural part. A *segment group* g_i is a set of non-overlapping segments, and represents the multiple occurrences of one structural part. These concepts are illustrated in Figure 3.

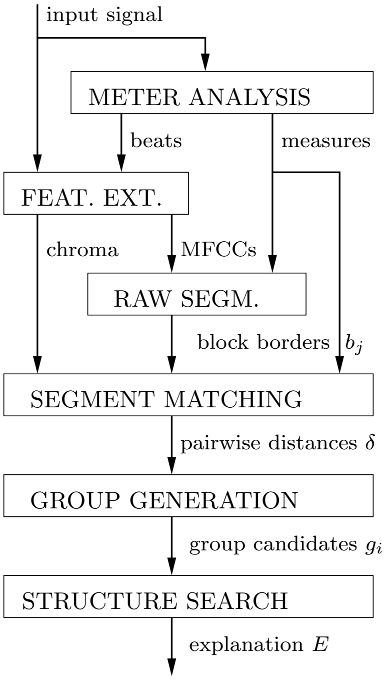


Figure 2: A block diagram of the whole system.

A segment group is characterised by properties termed *within-group dissimilarity* d_i, and *coverage* c_i. The first one describes how much the segments in a group differ from each other, and is detailed in Section 2.4. The coverage of a group c_i describes how much of the whole piece it covers,

$$c_i = \frac{\sum_{s_k \in g_i} \sum_{b_j \in s_k} t_j - \tau_j}{\sum_{j=1}^{N} t_j - \tau_j}. \tag{1}$$

All segments groups form a set of *segment group candidates* $G = \{g_1, g_2, \ldots, g_M\}$. An *explanation* E of a piece is a subset of the group candidates, $E \subset G$, such that the segments of the included groups do not overlap in time. If some part of the piece is not covered by the explanation E, it remains unexplained.

Given an explanation E of the structure of the piece, its fitness or cost can be calculated. For this purpose, we propose a cost function consisting of three terms as follows:

$$C = \text{dissimilarity} + \alpha \text{ unexplained} + \beta \text{ complexity}, \tag{2}$$

where α is a weighting factor for the cost due to unexplained blocks in the piece, and β is a weighting factor for the complexity (number of structural parts g_i) of the explanation. The first term penalises groups that have segments differing from each other, i.e., a group should consist only of the occurrences of a certain part. The second term defines the cost for the unexplained parts of the piece: the larger the explained proportion is, the smaller the cost will be. The last term defines the cost of the complexity of the explanation. The smaller the number of groups in an explanation $|E|$, the better. The effect of the weight parameters is illustrated in Figure 4, which contains the annotated structure of a piece and analysis results with three different parameter values.

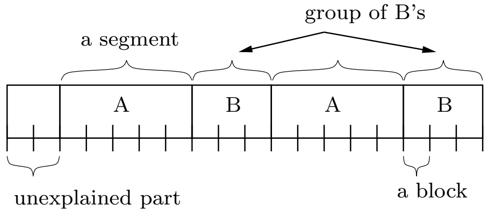


Figure 3: A piece is divided into blocks. Consecutive blocks form segments. Segments with same label create groups. Some part of the piece may be left unexplained.

The cost (2) is evaluated as

$$\begin{aligned} C(E) &= \sum_{g_i \in E} d_i c_i + \alpha(1 - \sum_{g_i \in E} c_i) + \beta \log(1 + |E|) \\ &= \alpha + \sum_{g_i \in E} c_i(d_i - \alpha) + \beta \log(1 + |E|). \end{aligned} \tag{3}$$

Structure analysis can now be viewed as an optimisation task to minimise the cost (3), and an algorithm for this purpose is presented in Section 2.5. It operates on segment groups and does not depend on how the groups are formed, there are several ways how the piece can be divided into blocks b_j, how the blocks are combined into segments s_k, and how the dissimilarities d_i of segment groups are calculated. Before describing the optimisation method, we present the way we used to perform these tasks.

2.2 Front-End Signal Analysis

As noted, e.g., in [24, 5, 4], the result of musical structure analysis may improve if the low-level analysis is done in synchrony with the metrical structure of the piece. The proposed system initially estimates the time-varying period and phase of the metrical levels: beat and musical measure. The estimation is done with the method presented in [17], utilising a bank of comb filter resonators and a probabilistic model to determine the periods and phases. The periods are here halved to minimise the problems due to possible *pi*-phase errors in the estimated pulses, which cause the pulses to be shifted by half a period from the correct locations.

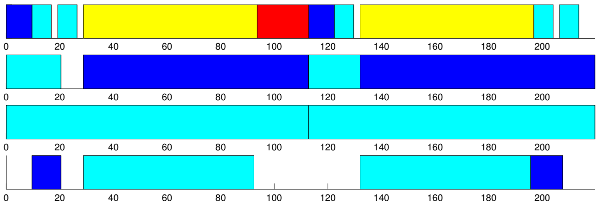


Figure 4: An illustration of the effect of the cost function parameter values. The top panel is the ground truth annotation of the structure of the piece. The lower panels contain example analysis results with varying values of the parameters α and β. The second panel shows the result with "reasonable" parameter values. Increasing the complexity weight β leads to the result in the third panel, and by decreasing the weight of unexplained parts, the result of the bottom panel was obtained.

The acoustic signal is then divided into frames according to the beat pulse, i.e., a frame is taken between two beat occurrences. A 36-dimensional chroma vector is extracted from each frame. Normally, the chroma vector is 12-dimensional, corresponding to the 12 pitch classes on the Western musical scale and describing the amount of energy present in all the frequency bands corresponding to occurrences of the pitch classes, ignoring the absolute height.

Here, we extract the chroma feature from three frequency ranges ($63.5 - 254$ Hz, $254 - 1020$ Hz, and $1.02 - 4.07$ kHz) using the chroma calculation method from [12] in each range separately. The results are concatenated into a feature vector. The use of three separate frequency regions gives more information about the tonal contents of the signal. This is roughly similar to [20], where the chroma was extracted from a range of three octaves and the pitch class equivalences were not summed together, resulting in a 36-dimensional feature vector. Principal component analysis is applied to reduce the feature vector dimensionality and only the 15 largest components are retained.

In addition to the chroma, 13 MFCCs are calculated with the same frames as the chroma values. The zeroth coefficient which is often discarded, is also retained. For each measure, the mean and variance of the MFCCs from the beat frames within it are calculated. The mean and variance values are concatenated to yield a 26-dimensional feature vector, and the statistics vectors are normalised to zero mean and unity variance over the whole piece.

2.3 Raw Segmentation

After extracting the features from the signal, a rough segmentation is employed to create a candidate set of the locations where a structural part may start or end. These segment borders can be defined in several ways. The simplest alternative would be not to define them at all, but to allow a segment to start and stop anywhere. As noted in [8], the main problem with this is that the computational complexity becomes very high at the latter stages. If there are N different locations where the part borders may lie, there exists $O(N^4)$ pairs of non-overlapping segments between them.

A data-adaptive method for determining the lengths and locations of segments utilising their recurrence information was described in [5]. Here, we chose to use a simple method relying on musical texture changes [10]. It has been noted that often different musical parts in a piece have clearly different timbral properties, and the method utilises this observation. A self-similarity matrix $\mathbf{S}$ is calculated from the measure-wise MFCC statistics. The similarity between feature vectors $\mathbf{m}_i$ and $\mathbf{m}_j$ from measures i and j, respectively, is defined to be

$$\sigma(\mathbf{m}_i, \mathbf{m}_j) = 0.5 + 0.5 \frac{< \mathbf{m}_i, \mathbf{m}_j >}{\parallel \mathbf{m}_i \parallel \parallel \mathbf{m}_j \parallel}, \qquad (4)$$

which stems from cosine distance measure. The elements in the matrix are $[\mathbf{S}]_{i,j} = \sigma(\mathbf{m}_i, \mathbf{m}_j)$.

A novelty vector is calculated from the similarity matrix by correlating a Gaussian taper checkerboard kernel matrix along the main diagonal of the similarity matrix (see [10] for details). Effectively it operates as a 2D border detection method, resulting into peaks in the novelty vector wherever the texture changes considerably. The border locations are assigned to the locations of the largest peaks. The exact number is not critical, as the locations do not define the final structural segmentation of the piece. However, it should be large enough to make sure that the real segment borders are contained within them, and still small enough not to make the subsequent matching computationally too expensive.

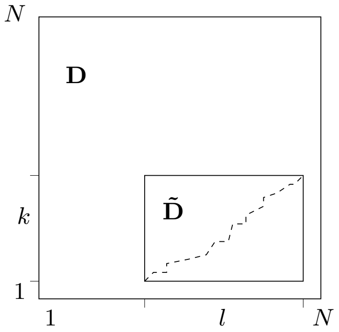


Figure 5: Pairwise distance calculation.

2.4 Segment Matching

To enable evaluating the cost function (3), the within-group dissimilarity d_i must be defined. Initially, all possible non-overlapping segment pairs are generated, so that the segments in a pair may start and end at any of the border locations, as long as they do not overlap each other. After that, the distance between the segments in a pair is calculated by utilising a two-level dynamic time warping, motivated by the multi-scale processing in [15] and the measure-level similarity matrix in [24]. The distance calculations are done using the reduced-dimensionality chroma vectors.

First, a measure-level distance matrix $\mathbf{D}$ is calculated with a method similar to the one described in [24]. The element $[\mathbf{D}]_{i,j}$ denotes the distance between the measures i and j in the piece. For each measure, the chroma frames contained within its range are used. The frames of two measures are matched with DTW. It allows small adjustments to the temporal alignment of the frames, enabling the matching even if the metrical estimation has had problems and the number of frames differs between the matched measures. At the feature vector level, the used distance measure is cosine, i.e., distance between vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ is $1 - \sigma(\mathbf{v}_i, \mathbf{v}_j)$.

The pairwise distance $\delta(k, l)$ between the segments s_k and s_l is calculated by finding the cheapest path through the sub-matrix $\tilde{\mathbf{D}}$ of $\mathbf{D}$ defined by the borders of the matched segments, as illustrated in Figure 5. The path can be found using dynamic programming. The pairwise distance $\delta(k, l)$ is defined to be the total cost of the optimal path, normalised with the length of the diagonal of the sub-matrix.

Since all occurrences of a part should be approximately of the same length, we prune all pairs with length ratio outside the range $r = [\frac{5}{6}, \frac{6}{5}]$, resulting to pruning of a large quantity of the pairs. The within-group dissimilarity d_i of the group g_i is then defined to be the average of all the pairwise distances in the group

$$d_i = \frac{\sum_{k, s_k \in g_i} \sum_{l, s_l \in g_i, l \neq k} \delta(k, l)}{\sum_{k, s_k \in g_i} \sum_{l, s_l \in g_i, l \neq k} 1}. \qquad (5)$$

The group candidate set G can now be constructed iteratively with the following steps:

1. Initialise the group set G with all the possible segment pairs after pruning pairs with length ratio outside r.

2. Create new groups by extending the groups that were added in the previous iteration with all non-overlapping segments. I.e., for each group, as many new groups are generated as there are non-overlapping segments. If the ratio of the largest and smallest pairwise distance between the segments in a group is outside r, delete the group.

3. Calculate new within-group dissimilarities d_i for the added groups using (5).

4. Repeat from Step 2, until no more items can be added or some maximum group size is reached.

Now the optimal explanation can be found by creating all such subsets of G that are valid explanations E, and evaluating the cost function for them. However, this is computationally very expensive.

2.5 Structure Search Algorithm

An efficient search algorithm can be implemented by constructing and evaluating the explanations incrementally after some data reorganisation. The group candidates in G are sorted into ascending order based on the within-group dissimilarities d_i. Initialise global variables
$C_{best} \leftarrow \alpha$, and
$E_{best} \leftarrow \{\emptyset\}$, initialise iteration base with
$E \leftarrow \{\emptyset\}$,
$l \leftarrow 1$, and
$C_{old} \leftarrow \alpha$, and call the function `Cutting-Search` with the initialised parameters.

```
 1: function Cutting-Search($E$, $l$, $C_{old}$)
 2:    for $i \leftarrow l, M$ do
 3:       if $g_i \cap g_j = \{\emptyset\}, \forall g_j \in E$ then
 4:          $E_{new} \leftarrow E \cup g_i$
 5:          $C_{new} \leftarrow C_{old} + c_i(d_i - \alpha) + \beta \log \frac{|E|+2}{|E|+1}$
 6:          if $C_{new} < C_{best}$ then
 7:             $C_{best} \leftarrow C_{new}$
 8:             $E_{best} \leftarrow E_{new}$
 9:          end if
10:          $C_{limit} \leftarrow C_{new} + (1 - \sum_{j, g_j \in E_{new}} c_j)(d_i - \alpha) + \beta \log \frac{|E|+3}{|E|+2}$
11:          if $C_{limit} < C_{best}$ then
12:             CuttingSearch($E_{new}$, $i + 1, C_{new}$)
13:          end if
14:       end if
15:    end for
16:    return $C_{best}$, $E_{best}$
17: end function
```

The loop upper limit M is the number of group candidates in the set G. Here g_i refers to the member i of the list of group candidates G constructed above, and $|E|$ denotes the number of groups in the explanation E.

The algorithm would evaluate all possible combinations of the candidate groups by extending the existing explanation with the recursive call on line 12, but the size of the search space is reduced. The reduction is done by estimating the lower bound, C_{limit}, of the cost that can be obtained by extending this explanation (line 10). If the obtainable cost is worse than the minimum cost found so far, there is no use of extending the search in this direction. Hence, the algorithm finds the globally optimal combination of groups g_i in respect to the cost function (3).[1] This explanation is returned in E_{best}.

Table 1: Statistics about the musical structures of the pieces in the used database.

data set	parts	occurrences	length
MUSIC	4.2	3.4	19.2s
RWC-Pop	4.3	2.4	21.4s
The Beatles	3.7	2.8	13.2s
whole db	**4.1**	**3.1**	**18.9s**

3. EVALUATION

The performance of the system was evaluated in simulations analysing the structure of a set popular music pieces and the structural explanations given by the algorithm were compared with a manually annotated reference structures.

3.1 Material

To enable evaluating the analysis result of the algorithm, a database of structural annotations for 50 musical pieces was collected. The used pieces were from three different sources: 34 popular music pieces from the MUSIC database, described in [14], 10 pieces from the RWC Popular Music database [13], and 6 pieces performed by The Beatles.[2]

The structure annotations were done manually, without any automatic tools. From each piece, the clear structural elements were annotated, i.e., some temporal locations of the piece may be left unannotated, as they did not belong into any clear structural part. An annotation consists of a list of the occurrences of structural parts, each with a start and a stop time and a name. The main focus was on parts that occurred more than once during the piece. Still, if there was some clear parts that could be named (e.g., intro, bridge, solo), they were annotated too. Also, if some part could be interpreted to be an occurrence of some other label, these alternative labels were annotated. E.g., if the solo part was very similar to the main verse part, but still different enough, it was annotated to be solo, but with the alternative label of verse. This was done because the labelling is not always unambiguous even for a human listener, let alone to a computer. By defining the alternative labels, more structural interpretations were allowed. The pieces in the database contain in total 642 annotated part occurrences, and 31 of these occurrences have an alternative label definition. As the system is capable of recognising only the parts that occur at least twice, the parts occurring only once were removed from the reference before evaluation.

Some statistics about the structural annotations of the individual parts and the whole database are presented in Table 1. The column *parts* tells the average number of unique labels in each piece on average, the column *occurrences* how many time each label occurs in a piece on average, and the column *length* the average length of a part occurrence in a piece. The statistics have been calculated from all annotated parts, without discarding the ones occurring only once.

[1] For a 5 min piece, with reasonable values for α and β, the optimisation algorithm finds the solution in a matter of seconds, when run on a 1.7 GHz Pentium 4 PC.

[2] The used pieces are listed at `http://www.cs.tut.fi/sgn/arg/paulus/structure/dataset.html`.

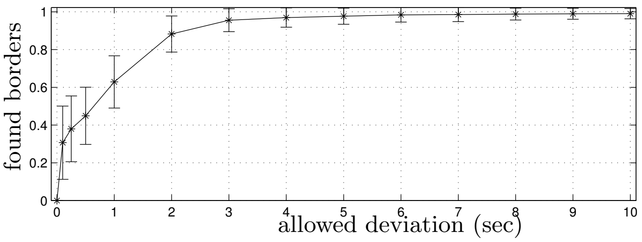


Figure 6: Ratio of found segment borders with different accuracy requirements. The error bars specify the standard deviation over all test pieces.

The actual evaluations were done with a 3-fold cross-validation scheme. In each fold, the values for the parameters α and β were trained with a grid search for the 2/3 of the material, and the determined values were used in evaluations for the remaining 1/3 of the material. The division of the available pieces to the three subsets was done randomly from each data source, i.e., it was taken care that the pieces from each source were divided approximately evenly to each subset. The presented results are calculated over all the folds.

3.2 Segmentation Accuracy

Since the performance of the border candidate generation affects the performance of the rest of the proposed system, it was tested separately. The border candidates were generated by using a convolution kernel of the length 12 s in novelty vector calculation, and choosing at most 50 largest peaks.[3] Each annotated segment border was matched with the closest found border if no other annotated border had been coupled with it yet and their time difference was smaller than the allowed deviation. The matching was performed with several different allowed deviation values and the ratio of found segment borders was calculated. The result is illustrated in Figure 6, where the line is the mean ratio of found segment borders and the error bars illustrate the standard deviation over the whole test sets.

3.3 Comparison of Structures

Given the explanation E_{best} for the structure of the piece by the algorithm and the reference structure annotation E_{ref}, the task is to compare the similarity of the two. As noted in [5] and illustrated by Figure 1, the structure of musical pieces is often hierarchical, at least in Western popular music. The annotated reference structure and the explanation given by the algorithm may be on a different level of the hierarchy, making a direct comparison impossible. Also, the labelling of the structural parts is not determined by the algorithm: it only assigns unique labels to the parts. Still, it would be desired that if the reference explanation and the one given by the algorithm describe the same structure in some level of the hierarchy, the similarity is recognised. An extreme case is illustrated in the top panel of Figure 7, the explanations describe different fine structure, but still they both have the same structure on the highest level of hierarchy, and hence they should be accepted as a match.

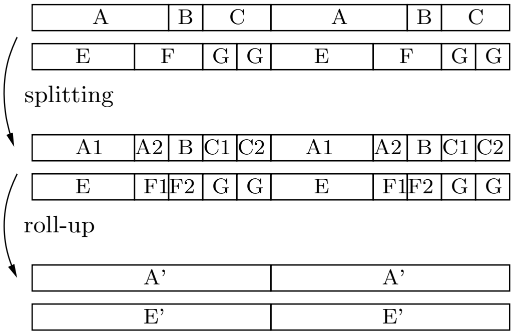


Figure 7: An example of how the fine structure of two descriptions differ, but the proposed evaluation method will judge the two similar.

To our knowledge, the level ambiguity has been addressed only by Chai [5]. She proposed to use a roll-up process in the evaluations to find a common level for comparison. In roll-up, if some part is split into finer structural description in either reference annotation or analysis result, the repetition of the finer structure are replaced with occurrences of the longer part. However, the proposed roll-up process does not yield a desired result in some cases, e.g., the one illustrated in top panel of Figure 7. Here, we extend the method presented therein.

The first problem is the temporal alignment of the reference and proposed structures: because the segmentation method described in Section 2.3 may not always find the same borders for parts as those assigned in reference annotations, or the borders lie at different levels of the hierarchy, it is required to create a common time base for both structural representations. This can be done by concatenating all the start and end times of all segments in both the reference and the analysis structures, sorting the times into ascending order, and removing possible duplicates. The main advantage of using the common time base is that the descriptions may be compared as sequences of characters.

The structural descriptions must now be represented using the generated time base. This is done by splitting the original segments, separately in both reference and analysis result, from the common temporal borders with the following steps:

1. Take the occurrences of a structural part that has not yet been handled.
2. Divide the occurrences according to the common time base. The resulting shorter sections are referred to as subparts.
3. Go through the subparts and assign them with new labels: (a) If a subpart occurs at the same location within more than one of the unsplit part occurrences with the same duration, all these occurrences with a same label. (b) If some subpart occurs without any repetition, assign it with its own label.

This operation is illustrated in the top part of Figure 7. The two structural descriptions with different segment borders in the top of the figure are split with common borders to result the descriptions in the middle of the figure.

The next step is to rise on the structural hierarchy with a procedure called the roll-up. It is illustrated in the lower

[3] The used maximum number of peaks was determined empirically to be a reasonable tradeoff between the segmentation accuracy and them computational complexity of the further steps in the algorithm.

Table 2: Evaluation results for each data subset and for the whole database. The results are calculated over the cross-validation folds.

segm.	data	R	P	F
perfect	MUSIC	72.3%	74.9%	72.9%
	RWC-Pop	80.9%	88.1%	84.2%
	The Beatles	95.8%	99.8%	97.7%
	whole db	**76.8%**	**80.6%**	**78.1%**
novelty	MUSIC	62.2%	68.5%	63.6%
	RWC-Pop	73.3%	85.3%	78.2%
	The Beatles	59.0%	81.5%	67.6%
	whole db	**64.0%**	**73.4%**	**67.0%**

part of Figure 7. In the roll-up, if similar sequence of segments occurs in both descriptions, ignoring the labels, the sequence is replaced in both descriptions with a part on a higher hierarchical level. In the illustrated example, the sequence "A1, A2, B, C1, C2" in the split reference occurs at the same location as the sequence "E, F1, F2, G, G" in the split analysis result, and hence the sequences may be rolled up in hierarchy to result to the lowest two descriptions.

After the roll-up, the structural descriptions should be on the same level of hierarchy and the actual evaluation can be done. The mapping between the label sets of the reference and analysis result is done so that each label from other set can be associated with one or no label in the other set. Each mapping is evaluated according to two measures: recall rate R and precision rate P. Recall rate describes how large temporal portion of the parts annotated in the reference are also explained by the analysis result. Precision rate describes how much of the analysis result is correct. From these two metrics, the harmonic F-measure is calculated to give one value describing the performance by $F = 2RP/(R + P)$.

3.4 Results

The overall evaluation results are presented in Table 2. The column *segm.* describes the method used in the segment border candidate determination, the value *novelty* denotes that the segmentation was generated with the audio novelty method described in Section 2.3, and the value *perfect* that the segment borders candidates were taken from the reference annotation. This was done to allow evaluating the effect of the border candidate generation method. The evaluation measures are presented for each data subset and for the whole database. The results for each song in the database are illustrated in Figure 8.

It can be observed that the performance on the MUSIC data set is considerably lower than with other subsets. When inspecting the reason for this, it was noted that the analysis had failed completely on some pieces. The reason for which similar failures were not encountered in the other two data sets is probably due to the musical diversity of the pieces in MUSIC. This leads to the conclusion that it may not be possible to find such values for the parameters α and β that they would perform equally well with all pieces. It can also be seen that the used segment border candidate generation method needs attention in future work.

Some typical results are illustrated in Figures 9-11. The figures contain the annotated structure in top panel and the analysis result in the lower panel. The analysis results were obtained using the system generated border location candidates and fixed parameter values found in the cross-validation runs.

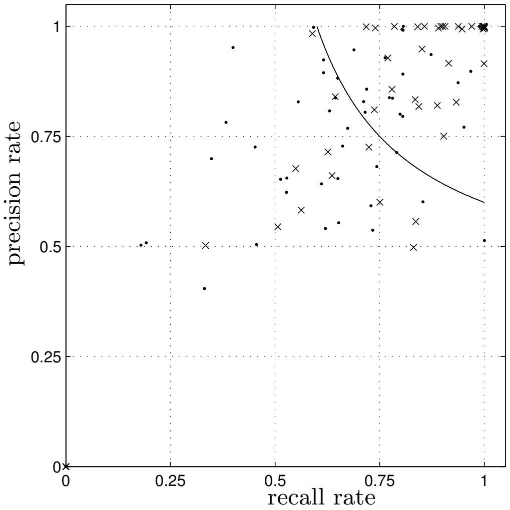


Figure 8: System performance on different pieces of the test database. The dots (·) are the performance using the novelty based border candidate generation, the crosses (×) are the performance with simulated perfect segmentation, and the solid line illustrates F-measure value 0.75 for a reference point.

4. CONCLUSION

The problem of structural analysis of musical pieces can be formulated in terms of a parametric cost function which allows varying the costs assigned to different aspects of the resulting description. The computational load incurred by the search for the description that minimises the cost is negligible compared to that of the signal-processing front-end. This allows to create an analysis application in which the user may control the cost function parameters interactively. An example of a such application is illustrated in Figure 12.

5. ACKNOWLEDGEMENTS

The musical structures for the evaluation material database was annotated by Teemu Karjalainen.

6. REFERENCES

[1] S. Abdallah, K. Nolad, M. Sandler, M. Casey, and C. Rhodes. Theory and evaluation of a Bayesian music structure extractor. In *Proc. of 6th International Conference on Music Information Retrieval*, London, UK, Sept. 2005.

[2] J.-J. Aucouturier and M. Sandler. Segmentation of musical signals using hidden Markov models. In *Proc. of 110th Audio Engineering Society Convention*, Amsterdam, The Netherlands, May 2001.

[3] M. A. Bartsch and G. H. Wakefield. To catch a chorus: Using chroma-based representations for audio thumbnailing. In *Proc. of 2003 IEEE Workshop on Applications of Signal Processing to Audio and*

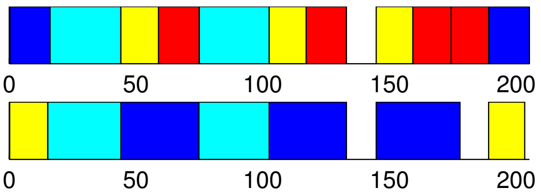


Figure 9: Structure annotation (top panel) and analysis result (lower panel) of Abba's song "S.O.S.". The result is relatively good with only one totally missed part.

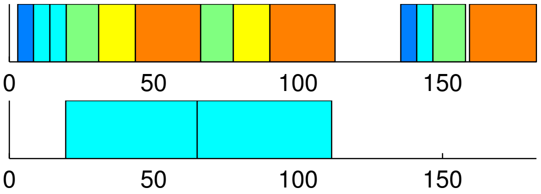


Figure 10: Annotation and analysis result of the piece RM-P035 ("Midarana kami no moushigo" by Hiromi Yoshii) from RWC Popular Music Database. The result is not that good, but still it has found a long part occurring twice.

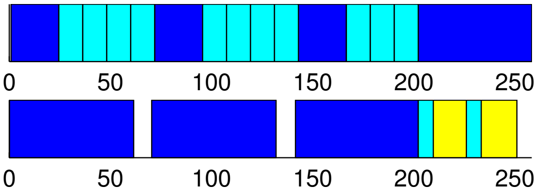


Figure 11: Annotation and analysis result of the piece "You got me" by The Roots. The analysis has grouped the continuation of chorus and three occurrences of the verse as one part and revealed a finer structure within the outro/chorus.

Acoustics, pages 15–18, New Platz, New York, USA, Oct. 2003.

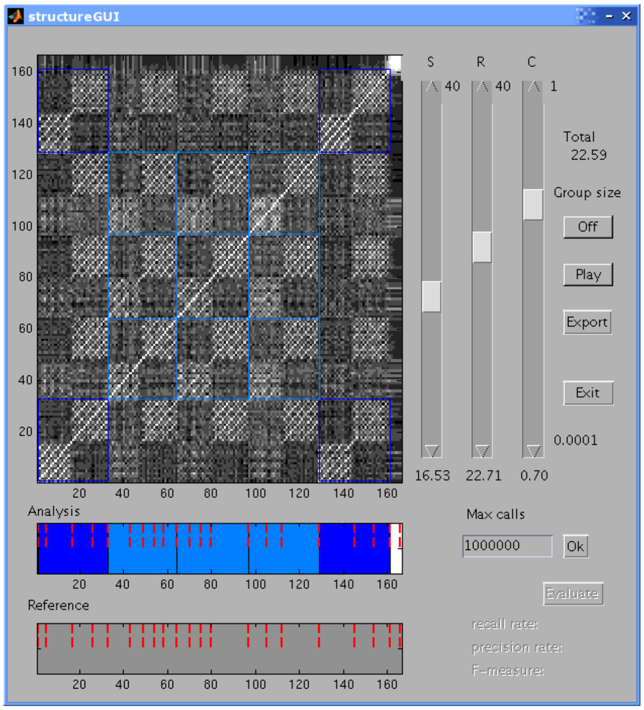
[4] M. A. Bartsch and G. H. Wakefield. Audio thumbnailing of popular music using chroma-based representations. *IEEE Transactions on Multimedia*, 7(1):96–104, Feb. 2005.

[5] W. Chai. *Automated Analysis of Musical Structure.* PhD thesis, Massachusetts Institute of Technology, Sept. 2005.

[6] W. Chai. Semantic segmentation and summarization of music: methods based on tonality and recurrent structure. *IEEE Signal Processing Magazine*, 23(2):124–132, Mar. 2006.

[7] R. B. Dannenberg. Toward automated holistic beat tracking, music analysis, and understanding. In *Proc. of 6th International Conference on Music Information Retrieval*, pages 366–373, London, UK, Sept. 2005.

[8] R. B. Dannenberg and N. Hu. Pattern discovery techniques for music audio. In *Proc. of 3rd International Conference on Music Information Retrieval*, pages 63–70, Paris, France, Oct. 2002.



Total
22.59

Max calls
1000000

Figure 12: An example of an interactive graphical user interface allowing the user to control the cost function parameter values while getting an immediate feedback. The UI displays, among others, the measure-wise similarity matrix, border candidate locations, and the analysis result.

[9] J. Foote. Visualizing music and audio using self-similarity. In *Proc. of ACM Multimedia*, pages 77–80, Orlando, Florida, USA, 1999.

[10] J. Foote. Automatic audio segmentation using a measure of audio novelty. In *Proc. of IEEE International Conference on Multimedia and Expo*, pages 452–455, New York, USA, Aug. 2000.

[11] J. T. Foote and M. L. Cooper. Media segmentation using self-similarity decomposition. In *Proc. of The SPIE Storage and Retrieval for Multimedia Databases*, volume 5021, pages 167–175, San Jose, California, USA, Jan. 2003.

[12] M. Goto. A chorus-section detecting method for musical audio signals. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 437–440, Hong Kong, 2003.

[13] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: Popular, classical, and jazz music databases. In *Proc. of 3rd International Conference on Music Information Retrieval*, pages 287–288, Paris, France, Oct. 2002.

[14] T. Heittola. Automatic classification of music signals. Master's thesis, Tampere University of Technology, Dec. 2003.

[15] T. Jehan. *Creating Music by Listening.* PhD thesis, Massachusetts Institute of Technology, Sept. 2005.

[16] T. Jehan. Hierarchical multi-class self similarities. In *Proc. of 2005 IEEE Workshop on Applications of*

Signal Processing to Audio and Acoustics, pages 311–314, New Platz, New York, USA, Oct. 2005.

[17] A. Klapuri, A. Eronen, and J. Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Speech and Audio Processing*, 14(1):342–355, Jan. 2006.

[18] M. Levy, M. Sandler, and M. Casey. Extraction of high-level musical structure from audio data and its application to thumbnail generation. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 13–16, Toulouse, France, May 2006.

[19] B. Logan and S. Chu. Music summarization using key phrases. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 749–752, Istanbul, Turkey, June 2000.

[20] L. Lu, M. Wang, and H.-J. Zhang. Repeating pattern discovery and structure analysis from acoustic music data. In *Proc. of Workshop on Multimedia Information Retrieval*, pages 275–282, New York, USA, Oct. 2004.

[21] N. C. Maddage, C. Xu, M. S. Kankanhalli, and X. Shao. Content-based music structure analysis with applications to music semantics understanding. In *Proc. of ACM Multimedia*, pages 112–119, New York, New York, USA, Oct. 2004.

[22] G. Peeters. Deriving musical structure from signal analysis for music audio summary generation: "sequence" and "state" approach. In *Lecture Notes in Computer Science*, volume 2771, pages 143–166. Springer-Verlag, 2004.

[23] C. Rhodes, M. Casey, S. Abdallah, and M. Sandler. A Markov-chain Monte-Carlo approach to musical audio segmentation. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 797–800, Toulouse, France, May 2006.

[24] Y. Shiu, H. Jeong, and C.-C. J. Kuo. Musical structure analysis using similarity matrix and dynamic programming. In *Proc. of SPIE Vol. 6015 - Multimedia Systems and Applications VIII*, 2005.

Publication P7

J. Paulus and A. Klapuri, "Labelling the structural parts of a music piece with Markov models," in *Computer Music Modeling and Retrieval: Genesis of Meaning in Sound and Music - 5th International Symposium, CMMR 2008 Copenhagen, Denmark, May 19-23, 2008, Revised Papers*, ser. Lecture Notes in Computer Science, vol. 5493, S. Ystad, R. Kronland-Martinet, and K. Jensen, Eds. Springer Berlin / Heidelberg, 2009, pp. 166–176. doi: 10.1007/978-3-642-02518-1_11

©2009 Springer Berlin / Heidelberg. Reprinted with permission. This is an author-created version of the article. For the definitice version, refer to the original publication.

Labelling the Structural Parts of a Music Piece with Markov Models

Jouni Paulus and Anssi Klapuri

Department of Signal Processing, Tampere University of Technology
Korkeakoulunkatu 1, Tampere, Finland
{jouni.paulus, anssi.klapuri}@tut.fi

Abstract. This paper describes a method for labelling structural parts of a musical piece. Existing methods for the analysis of piece structure often name the parts with musically meaningless tags, e.g., “p1”, “p2”, “p3”. Given a sequence of these tags as an input, the proposed system assigns musically more meaningful labels to these; e.g., given the input “p1, p2, p3, p2, p3” the system might produce “intro, verse, chorus, verse, chorus”. The label assignment is chosen by scoring the resulting label sequences with Markov models. Both traditional and variable-order Markov models are evaluated for the sequence modelling. Search over the label permutations is done with N-best variant of token passing algorithm. The proposed method is evaluated with leave-one-out cross-validations on two large manually annotated data sets of popular music. The results show that Markov models perform well in the desired task.

1 Introduction

Western popular music pieces often follow a sectional form in which the piece is constructed from shorter units. These units, or musical parts, may have distinct roles on the structure of the piece, and they can be named based on this role, for example, as “chorus” or “verse”. Some of the parts may have several occurrences during the piece (e.g., “chorus”) while some may occur only once (e.g., “intro”).

To date, several methods have been proposed to perform automatic analysis of the structure of a musical piece from audio input, see [1] or [2] for a review. Majority of the methods do not assign musically meaningful labels to the structural parts they locate. Instead, they just provide information about the order, possible repetitions, and temporal boundaries of the found parts. There also exist a few methods that utilise musical models in the analysis, and the resulting structure descriptions have musically meaningful labels attached to the found parts [3,4].

The musical piece structure can be used, for example, in a music player user interface allowing the user to navigate within the piece based on musical parts [5]. The results of a user study with a music player having such a navigation ability suggest that the parts should be labelled meaningfully. The additional information of knowing which of the parts is for instance “chorus” and which is “solo” was judged to be valuable [6].

The proposed method does not perform the musical structure analysis from audio, but only labels structural descriptions and should be considered as an add-on or an extension to existing structure analysis systems. So, the problem to be solved here is

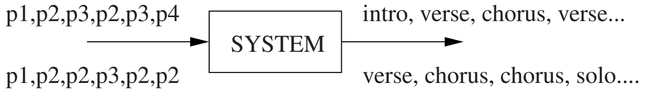

p1,p2,p3,p2,p3,p4
p1,p2,p2,p3,p2,p2
SYSTEM
intro, verse, chorus, verse...
verse, chorus, chorus, solo....

Fig. 1. Basic idea of the system. The system assigns meaningful labels to arbitrary tags based on a musical model. The mapping from tags to labels is determined separately for each input.

how to assign musically more meaningful part *labels* when given a sequence of *tags* describing the structure of a musical piece. The operation is illustrated in Figure 1. As an example, the structure of the piece “Help!” by The Beatles is “intro, verse, refrain, verse, refrain, verse, refrain, outro”, as given in [7]. A typical structure analysis system might produce “p1,p2,p3,p2,p3,p2,p3,p4” as the result, which then would be the input to the proposed system. If the system operation was successful, the output would be the assignment: “p1 → intro, p2 → verse, p3 → refrain, p4 → outro”.

It is often said more or less seriously that popular music pieces tend to be of the same form, such as “intro, verse, chorus, verse, chorus, solo, chorus”.[1] The proposed method aims to utilise this stereotypical property by modelling the sequential dependencies between part labels (occurrences of musical parts) with Markov chains, and searching the label assignment that maximises the probability of the resulting label sequence. Evaluation show that the sequential dependencies of musical parts are so informative that they can be used in the labelling.

The rest of the paper is structured as following: Sect. 2 describes the proposed method. The labelling performance of the method is evaluated in Sect. 3. Sect. 4 gives the conclusions of the paper.

2 Proposed Method

The input to the system is a sequence of tags $R_{1:K} \equiv R_1, R_2, \ldots, R_K$, and the problem is to assign a musical label to each of the unique tags so that no two tags are assigned the same label. This assignment is defined as an injective mapping function $f : T \rightarrow L$ from input set T of tags to the output set L of musically meaningful labels, as illustrated in Figure 2. The mapping function transforms the input tag sequence $R_{1:K}$ into a sequence of labels $f(R_{1:K}) = S_{1:K}$.

The proposed methods assumes that the musical parts depend sequentially on each other in the form of a Markov chain and that it is possible to predict the next musical part given a finite history of the preceding parts. This predictability is used to score different mapping alternatives and the best mapping is then given as the output of the system.

[1] Though statistics from two data sets of popular music pieces show that the structures of the pieces are more heterogeneous than was initially expected, see Sect. 3.1.

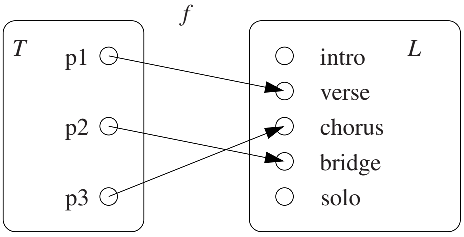


Fig. 2. An example of the mapping function $f : T \rightarrow L$. All tags in T are mapped to one label in L, but some labels in L may remain unused.

2.1 Markov Models

Markov models assume that the probability of a continuation S_i for sequence $S_{1:(i-1)}$ depends only on a finite history of the sequence $S_{(i-(N-1)):(i-1)}$ instead of the full history, i.e., $p(S_i|S_{1:(i-1)}) = p(S_i|S_{(i-(N-1)):(i-1)})$, where $(N-1)$ is the length of the used history. This is also referred as the order of the resulting Markov model and gives rise to the alternative name of N-grams. Based on the Markov assumption, the overall probability of a sequence $S_{1:K}$ is obtained by

$$p(S_{1:K}) = \prod_{k=1}^{K} p(S_k|S_{(k-(N-1)):(k-1)}) \quad . \tag{1}$$

In the beginning of the sequence where there is not enough history available, it is possible to use a lower order model or pad the sequence from the beginning with a special symbol. [8]

The total N-gram probability of (1) is used to score different mapping functions by evaluating it for the output sequences after the mapping $f(R_{1:K}) = S_{1:K}$. The target is to find the mapping function f_{OPT} that maximises the total probability

$$f_{OPT} = \underset{\text{f}}{\text{argmax}}\{p(f(R_{1:K}))\},\ f : T \rightarrow L \text{ injective} \quad . \tag{2}$$

2.2 Optimisation Algorithm

The maximisation problem is solved by using M-best[2] variant of token passing (TP) algorithm, more frequently used in speech recognition [9]. The main principle of TP is that tokens t are propagated time synchronously between the states of the model. Each token knows the path it has travelled and accumulates the total probability over it. Based on the path probabilities, the M tokens with the highest probabilities are selected for propagation in each state, they are replicated and passed to all connected states. The token path probabilities are updated based on the transition probabilities between the states.

[2] Better known as the N-best token passing. The name is adjusted to avoid possible confusion with N-grams.

The state space of TP is formed from the possible labels in L, and the paths of the tokens encode different mapping functions. The optimisation of (2) can be done by searching the most probable path through the states (labels) defining the state transition probabilities with

$$p\left(f_k(R_k) = l_i | R_{1:(k-1)}, f_{k-1}\right) = \begin{cases} 0, & \text{if } \mathtt{DME} \\ p(l_i | f_{k-1}(R_{1:(k-1)})), & \text{otherwise} \end{cases} , \quad (3)$$

where DME denotes the predicate "different mapping exists", which is used to guarantee that the mapping function is injective, and it is defined by

$$\mathtt{DME} = \exists j : \left(R_j = R_k \wedge f_{k-1}(R_j) \neq l_i\right) \vee \left(R_j \neq R_k \wedge f_{k-1}(R_j) = l_i\right), j \in [1, k-1] \ . \quad (4)$$

In the equations above, $p\left(f_k(R_k) = l_i | R_{1:k}, f_{k-1}\right)$ denotes the probability of a token to transition to the state corresponding to label l_i after it has travelled the path $f_{k-1}\left(R_{1:(k-1)}\right)$. The N-gram probability for label l_i when the preceding context is $f_{k-1}\left(R_{1:(k-1)}\right)$, is denoted as $p(l_i | f_{k-1}(R_{1:(k-1)}))$. As the mapping is generated gradually, f_k is used to denote the mapping after handling the sequence $R_{1:k}$.

Pseudocode of the algorithm is given in Algorithm 1. It searches a mapping function $f : T \rightarrow L$ from tags in input sequence to the possible label set. For each label $l \in L$, the probability $\pi_0(l)$ of that label to be the first label in the sequence and the probability the label the be the last $\pi_E(l)$ are defined. In the middle of the sequence, the probability of the continuation given the preceding context is obtained from (3).

As the mapping depends on decisions done within the whole preceding history, the Markov assumption is violated and the search cannot be done with more efficient methods guaranteeing a globally optimal solution. This sub-optimality hinders also the traditional TP, since it might be that the optimal labelling may not be the best one earlier in the sequence, and is therefore pruned during the search. The M-best variant of TP alleviates this problem by propagating M best tokens instead of only the best one. If all tokens were propagated, the method would find the globally optimal solution, but at a high computational cost. With a suitable number of tokens, a good result can be found with considerably less computation than with an exhaustive search. An exhaustive search was tested, but due to the large search space, it proved to be very inefficient. However, it was used to verify the operations of TP with a subset of the data. In that subset, the TP showed to find the same result as the exhaustive search in almost all the cases when storing 100 tokens at each state.

2.3 Modelling Issues

The main problem with N-gram models is the amount of training material needed for estimating the transition probabilities: the amount increases rapidly as a function of the number of possible states and the length of used history (given A possible states and history length of N, there exist A^N probabilities to be estimated). It may happen that not all of the sequences of the required length occur in the training data. This situation can be handled by back-off (using shorter context at those cases), or by smoothing (assigning a small amount of the total probability mass to the events not encountered in the training material).

Algorithm 1: Search label mapping $f : T \rightarrow L$

```
Input sequence $R_{1:K}$.
Label space $L$. Associated with each label $l \in L$, there are input buffer $I_l$ and output buffer $O_l$.
Tokens $t$ with probability value $t.p$ and label mapping function $t.f$.
for $l \in L$ do                                                    // initialisation
    Insert $t$ to $I_l$ and assign $t.p \leftarrow \pi_0(l)$
for $k \leftarrow 1$ to $K$ do
    for $l \in L$ do
        $O_l \leftarrow I_l$                                        // propagate to output
        Clear $I_l$
    for $l \in L$ do                                                // transition source
        for $t \in O_l$ do
            for $\tilde{l} \in L$ do                                // transition target
                $\tilde{t} \leftarrow t$                            // copy token
                if $\exists j : R_j = R_k, j \in [1, k-1]$ then
                    if $\tilde{t}.f(R_k) = \tilde{l}$ then
                        $\tilde{t}.p \leftarrow \tilde{t}.p \times p(\tilde{t}.f(R_k)|\tilde{t}.f(R_{1:(k-1)}))$    // N-gram probability
                    else
                        $\tilde{t}.p \leftarrow 0$
                else
                    if $\forall j : \tilde{t}.f(R_j) \neq \tilde{l}, j \in [1, k-1]$ then
                        Set $\tilde{t}.f(R_k) \leftarrow \tilde{l}$
                        $\tilde{t}.p \leftarrow \tilde{t}.p \times p(\tilde{t}.f(R_k)|\tilde{t}.f(R_{1:(k-1)}))$
                    else
                        $\tilde{t}.p \leftarrow 0$
                Insert $\tilde{t}$ to $I_{\tilde{l}}$
    for $l \in L$ do
        Retain $M$ best tokens in $I_l$
for $l \in L$ do
    $O_l \leftarrow I_l$
    for $t \in O_l$ do
        $t.p \leftarrow t.p \times \pi_E(l)$
Select token $\hat{t}$ with the largest $t.p$
return $\hat{t}.f$
```

In some cases, it is possible that increasing the length of the context does not provide any information compared to the shorter history. Variable-order Markov models (VMMs) have been proposed to replace traditional N-grams. Instead of using a fixed history, VMMs try to deduce the length of the usable context from the data. If increasing the length of the context does not improve the prediction, then only the shorter context is used. VMMs can be used to calculate the total probability of the sequence in the same manner as in (1), but using a variable context length instead of fixed N. [10]

3 Evaluations

Performance of the labelling method was evaluated in simulations using structural descriptions from real music pieces.

3.1 Data

The method was evaluated on two separate data sets. The first, *TUTstructure07*, was collected at Tampere University of Technology. The database contains a total of 557 pieces sampling the popular music genre from 1980's to present day.[3] The musical structure of each piece was manually annotated. The annotation consists of temporal segmentation of the piece into musical parts and naming each of the parts with musically meaningful labels. The annotations were done by two research assistants with some musical background. The second data set, *UPF Beatles*, consists of 174 songs by The Beatles. The original piece structures were annotated by musicologist Alan W. Pollack [7], and the segmentation time stamps were added at Universitat Pompeu Fabra (UPF)[4].

Though many of the forms are thought to be often occurring or stereotypical for music from pop/rock genre, the statistics from the data sets do not support this fully. In TUTstructure07, the label sequences vary a lot. The three most frequently occurring structures are

- "intro", "verse", "chorus", "verse", "chorus", "C", "chorus", "outro"
- "intro", "A", "A", "B", "A", "solo", "B", "A", "outro"
- "intro", "verse", "chorus", "verse", "chorus", "chorus", "outro",

each occurring four times in the data set. 524 (94%) of the label sequences are unique.

With UPF Beatles, there is a clearer top, but still there is a large body of sequences occurring only once in the data set. The most frequent label sequence is

- "intro", "verse", "verse", "bridge", "verse", "bridge", "verse", "outro",

occurring seventeen times in the data set. 135 (78%) of the label sequences are unique.

3.2 Training the Models

Transition probabilities for the models were trained using the data sets. Each label sequence representing the structure of a piece was augmented with special labels "BEG" in the beginning, and "END" in the end. After the augmentation, the total number of occurrences of each label in the data set was counted. Because there exists a large number of unique labels, some of which occur only once in the whole data set, the size of the label alphabet was reduced by using only the labels that cover 90% of all occurrences. The remaining labels were replaced with an artificial label "MISC". The zero-probability problem was addressed by using Witten-Bell discounting (Method C in [11]), except for the VMMs.

[3] List of the pieces is available at
`<http://www.cs.tut.fi/sgn/arg/paulus/TUTstructure07_files.html>`.

[4] `<http://www.iua.upf.edu/%7Eperfe/annotations/sections/license.html>`

In the original data sets, there were 82 and 52 unique labels (without the augmentation labels “BEG”, “END”, and “MISC”) in the data set of TUTstructure07 and UPF Beatles, respectively. After augmentation and set reduction the label set sizes were 15 and 10. On the average, there were 6.0 unique labels and 12.1 label occurrences (musical parts) in a piece in TUTstructure07. The same statistics for UPF Beatles were 4.6 and 8.6. This suggests that the pieces in TUTstructure07 were more complex or they have been annotated on a finer level.

3.3 Simulation Setup

In simulations, the structural annotations from the data base were taken. The original label sequences (with the “MISC” substitution) was taken as the ground truth, while the input to the labelling algorithm was generated by replacing the labels with letters.

To avoid overlap in train and test sets whilst utilising as much of the data as possible, simulations were run using leave-one-out cross-validation scheme. In each cross-validation iteration one of the pieces in the data set was left as the test case while the Markov models were trained using all the other pieces. This way the model never saw the piece it was trying to label.

With conventional N-grams, the length of the Markov chain was varied from 1 to 5, i.e., from using just prior probabilities for the labels to utilising context of length 4. With VMMs, several different algorithms were tested, including: decomposed context tree weighting (DCTW), prediction by partial matching - method C, and a variant of Lempel-Ziv prediction algorithm. The implementations for these were provided by [12]. It was noted that DCTW worked the best of these three, and the result are presented only for it. The maximum context length for VMMs was set to 5. Also the maximum context lengths of 3 and 10 were tested, but the former deteriorated the results and the latter produced practically identical results with the chosen parameter value.

3.4 Evaluation Metrics

When evaluating the labelling result, confusion matrix C for the labels is calculated. The result of the best mapping function applied to the input sequence $f(R_{1:K})$ and the ground truth sequence $S_{1:K}$ are compared. At each label occurrence $S_i, i \in [1,K]$, the value in the element $[S_i, f(R_i)]$ of the confusion matrix is increased by one. This applies weighting for the more frequently occurring labels. The confusion matrix is calculated over all cross-validation iterations. The average hit rate for a target label was calculated as a ratio of correct assignments (main diagonal of confusion matrix) to total occurrences of the label (sum along rows of the confusion matrix).

3.5 Results

The effect of varying the context length in N-grams is shown in Tables 1 and 2 for TUTstructure07 and UPF Beatles, respectively. In addition to the different N-gram lengths, the tables contain also the result for the best VMM (DCTW with maximum memory length of 5). The tables contain the percentage of correct assignments for each label

Table 1. Performance comparison on TUTstructure07 with traditional Markov models of different order. The best VMM result is given for comparison. The given values are the average hit rates in percents. The row *average* is the total average of correct part labels. The best result on each row is typeset with bold.

label	N=1	N=2	N=3	N=4	N=5	VMM
chorus	68.1	76.3	**80.8**	76.6	74.9	78.5
verse	42.3	62.4	64.4	64.9	**66.0**	66.0
bridge	17.7	38.6	45.6	**47.4**	44.4	43.7
intro	27.6	97.6	**98.2**	97.8	97.8	96.4
pre-verse	4.2	40.7	**46.3**	43.3	41.7	43.3
outro	13.9	98.3	**98.6**	97.8	92.1	98.3
c	0.0	38.0	42.1	47.4	**54.8**	49.3
theme	0.0	0.0	2.7	**4.4**	3.3	3.3
solo	0.0	4.4	7.2	16.0	**18.2**	14.9
chorus_a	0.0	0.0	7.5	**15.7**	11.2	3.0
a	0.0	0.0	**32.5**	31.7	27.0	29.4
chorus_b	0.0	0.9	5.3	**12.4**	7.1	2.7
MISC	12.6	29.5	38.3	37.1	**40.3**	38.3
average	30.9	55.6	**60.3**	59.9	59.5	59.8

used. The total average of correct hits ("average") is calculated without the augmentation labels "BEG" and "END".[5]

Based on the results in Tables 1 and 2, it can be seen that increasing the order of traditional Markov model from unigrams to bigrams produce a large increase in the performance. The performance continues to increase when the context length is increased, but more slowly. With TUTstructure07, the performance peak is at $N = 3$, whereas with UPF Beatles, the maximum with traditional N-grams can be obtained with $N = 4$. It was also noted that with TUTstructure07 the use of VMM did not improve the result. However, there is a small performance increase with VMMs in UPF Beatles.

Even though the use of VMM did not improve the result with TUTstructure07, there was one clear advantage with them: it was possible to use longer context in the models. With traditional N-grams, the transition probabilities will become very sparse even with bigrams. The large blocks of zero provide no information whatsoever and only consume memory. With VMMs, the context length is adjusted according to the available information.

From the results, it is notable that "chorus" can be labelled from the input over 80% accuracy, and "verse" almost at 65% accuracy in TUTstructure07. In UPF Beatles "verse" could be labelled with 87% accuracy and "refrain" with 71% accuracy.

[5] For an interested reader, the confusion matrices are given in a document available at `<http://www.cs.tut.fi/sgn/arg/paulus/CMMR08_confMats.pdf>`.

Table 2. Performance comparison on UPF Beatles with traditional Markov models of different order. The best VMM result is given for comparison. For description of the data, see the Table 1.

label	N=1	N=2	N=3	N=4	N=5	VMM
verse	72.4	79.9	86.7	85.7	83.7	**87.5**
refrain	30.1	32.1	62.2	66.3	68.7	**70.7**
bridge	36.7	40.7	**78.0**	74.0	74.0	70.6
intro	0.0	93.2	88.9	92.0	**93.8**	93.2
outro	0.0	99.3	**99.3**	97.2	93.0	97.9
verses	0.0	16.1	48.2	**50.0**	44.6	44.6
versea	0.0	5.9	7.8	17.6	**21.6**	5.9
MISC	0.0	15.9	22.3	**25.5**	23.6	22.3
average	33.5	58.9	72.1	72.8	72.1	**73.0**

3.6 Discussion

It should be noted that the proposed system performs the labelling purely based on a model of sequential dependencies of musical parts. Incorporating some acoustic information might improve the result somewhat (e.g., energetic repeated part might be "chorus"). Also, the knowledge of the high-level musical content, such as the lyrics, instrumentation or chord progressions, could provide valuable information for the labelling. However, the extraction of these from the acoustic input is still a challenging task, as well as creating a usable model for them. In addition, when discussing the principles used when assigning the ground truth labels with the annotators, the main cue was the location of the part in the "musical language model". Incorporating these other information sources in addition to the sequence model should be considered in the future work.

The difference in performance between the two data sets remains partly an open question. The main reason may be that the label sequences in TUTstructure07 are more diverse, as could be seen from the statistics presented in Sec. 3.1 (94% of sequences in TUTstructure are unique, compared to 78% in UPF Beatles). We tested the hypothesis that is was due to the smaller label set (10 vs. 15) by using only as many of the most frequent labels as were used with UPF Beatles. As a slight surprise, the performance on the remaining set was even worse compared label-wise to the larger set. The average result, however, increased slightly because the most rarely occurring (and most often mis-labelled) labels were omitted.

4 Conclusion

This paper has proposed a method for assigning musically meaningful labels to the parts found by music structure analysis systems. The method models the sequential dependencies between musical parts with Markov models and uses the models to score different label assignments. The paper has proposed applying M-best token passing algorithm to the label assignment search to be able to perform the assignment without having to test all possible permutations. The proposed method has been evaluated with leave-one-out cross-validations on two data sets of popular music pieces. The evaluation results suggest that the models for the sequential dependencies of musical parts are so informative even at low context lengths that they can be used alone for labelling. The obtained labelling performance was reasonable, even though the used model was relatively simple.

Acknowledgements

This work was supported by the Academy of Finland, project No. 5213462 (Finnish Centre of Excellence program 2006 - 2011). The TUTstructure07 was annotated by Toni Mäkinen and Mikko Roininen.

References

1. Peeters, G.: Deriving musical structure from signal analysis for music audio summary generation: "sequence" and "state" approach. In: Lecture Notes in Computer Science. Volume 2771. Springer-Verlag (2004) 143–166
2. Ong, B.S.: Structural analysis and segmentation of musical signals. PhD thesis, Universitat Pompeu Fabra, Barcelona (2006)
3. Shiu, Y., Jeong, H., Kuo, C.C.J.: Musical structure analysis using similarity matrix and dynamic programming. In: Proc. of SPIE Vol. 6015 - Multimedia Systems and Applications VIII. (2005)
4. Maddage, N.C.: Automatic structure detection for popular music. IEEE Multimedia **13**(1) (January 2006) 65–77
5. Goto, M.: A chorus-section detecting method for musical audio signals. In: Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, Hong Kong (2003) 437–440
6. Boutard, G., Goldszmidt, S., Peeters, G.: Browsing inside a music track, the experimentation case study. In: Proc. of 1st Workshop on Learning the Semantics of Audio Signals, Athens (December 2006) 87–94
7. Pollack, A.W.: 'Notes on...' series. The Official rec.music.beatles Home Page (http://www.recmusicbeatles.com) (1989–2001)
8. Jurafsky, D., Martin, J.H.: Speech and language processing. Prentice-Hall, New Jersey, USA (2000)
9. Young, S.J., Russell, N.H., Thornton, J.H.S.: Token passing: a simple conceptual model for connected speech recognition systems. Technical Report CUED/F-INFENG/TR38, Cambridge University Engineering Department, Cambridge, UK (July 1989)
10. Ron, D., Singer, Y., Tishby, N.: The power of amnesia: Learning probabilistic automata with variable memory length. Machine Learning **25**(2–3) (1996) 117–149
11. Witten, I.H., Bell, T.C.: The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. IEEE Transcations on Information Theory **37**(4) (1991) 1085–1094
12. Begleiter, R., El-Yaniv, R., Yona, G.: On prediction using variable order Markov models. Journal of Artificial Intelligence Research **22** (2004) 385–421

Publication P8

J. Paulus and A. Klapuri, "Music structure analysis using a probabilistic fitness measure and a greedy search algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1159–1170, Aug. 2009.
doi: 10.1109/TASL.2009.2020533

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the Tampere University of Technology's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org.

By choosing to view this material, you agree to all provisions of the copyright laws protecting it.

©2009 IEEE. Reprinted, with permission, from IEEE Transactions on Audio, Speech, and Language Processing.

Music Structure Analysis Using a Probabilistic Fitness Measure and a Greedy Search Algorithm

Jouni Paulus, *Student Member, IEEE*, and Anssi Klapuri, *Member, IEEE*

***Abstract*—This paper proposes a method for recovering the sectional form of a musical piece from an acoustic signal. The description of form consists of a segmentation of the piece into musical parts, grouping of the segments representing the same part, and assigning musically meaningful labels, such as "chorus" or "verse," to the groups. The method uses a fitness function for the descriptions to select the one with the highest match with the acoustic properties of the input piece. Different aspects of the input signal are described with three acoustic features: mel-frequency cepstral coefficients, chroma, and rhythmogram. The features are used to estimate the probability that two segments in the description are repeats of each other, and the probabilities are used to determine the total fitness of the description. Creating the candidate descriptions is a combinatorial problem and a novel greedy algorithm constructing descriptions gradually is proposed to solve it. The group labeling utilizes a musicological model consisting of N-grams. The proposed method is evaluated on three data sets of musical pieces with manually annotated ground truth. The evaluations show that the proposed method is able to recover the structural description more accurately than the state-of-the-art reference method.**

***Index Terms*—Acoustic signal analysis, algorithms, modeling, music, search methods.**

I. Introduction

HUMAN perception of music relies on the organization of individual sounds into more complex entities. These constructs occur at several time scales from individual notes forming melodic phrases to relatively long sections, often repeated with slight variations to strengthen the perception of musical organization. This paper describes a method for the automatic analysis of the musical structure from audio input, restricting the time scale to musical sections (or, parts), such as intro, verse, and chorus.

Information of the structure of a musical piece enables several novel applications, e.g., easier navigation within a piece in music players [1], piece restructuring (or mash-up of several pieces) [2], academic research of forms used in different musical styles, audio coding [3], searching for different versions of the same song [4], [5], or selecting a representative clip of the piece (i.e., music thumbnailing) [6]. A music structure analysis system provides relatively high-level information about the analyzed signal, on a level that is easily understood by an average music listener.

A. Background

Several systems have been proposed for music structure analysis, ranging from attempts to find some repeating part to be used as a thumbnail, to systems producing a structural description covering the entire piece. The employed methods vary also. In the following, a brief overview of some of the earlier methods is provided.

To reduce the amount of data and to focus on the desired properties of the signal, features are extracted from it. The feature extraction is done in fixed-length frames or in frames synchronized to the musical beat. The main motivation for using beat-synchronized frames is that they provide a tempo-invariant time base for the rest of the analysis.

The employed features are often designed to mimic some aspects that have been found to be important for a human listener analyzing the musical structure, including changes in timbre or rhythm, indicating change of musical parts, and repetitions, especially melodic ones, as suggested in [7]. In the following, the feature vector in frame i, $i = 1, 2, \ldots, V$, is denoted by $\boldsymbol{v}_i$, and V is the number of frames in the signal.

A useful mid-level representation employed in many structure analysis methods is a $V \times V$ self-distance (or self-similarity) matrix $\boldsymbol{D}$. The element $D(i, j)$ of the matrix denotes the distance (or similarity) of the frames i and j. The self-distance matrix (SDM) is a generalization of the recurrence plot [8] in which the element values are binary (similar or different). In music structure analysis, the use of SDM was first proposed in [9] where it was used for music visualization. The patterns in the SDM are not only useful for visualization but also important in many analysis methods.

In [10], structure analysis methods are categorized into *state* and *sequence*-based systems. State-based methods consider the piece as a succession of states, while sequence-based methods assume that the piece contains repeated sequences of musical events. Fig. 1 presents an idealized view of the patterns formed in the SDM. The state representation methods basically aim to locate blocks of low distance on the main diagonal, while the sequence-based methods aim to locate off-diagonal stripes (a stripe representing low distance of two sequences). The blocks are formed when the used feature remains somewhat similar during an occurrence of a musical part, and the stripes are formed when there are sequences that are repeated later in the piece.

The locations of the block borders on the main diagonal can be searched from the SDM for segmentation [11]–[13], or

Manuscript received December 30, 2008; revised March 20, 2009. Current version published June 26, 2009. This work was supported by the Academy of Finland under Project 5213462 (Finnish Centre of Excellence Program 2006–2011). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yariv Ephraim.

The authors are with the Department of Signal Processing, Tampere University of Technology, Korkeakoulunkatu 1, FI-33720 Tampere, Finland (e-mail: jouni.paulus@tut.fi; anssi.klapuri@tut.fi).

Digital Object Identifier 10.1109/TASL.2009.2020533

1558-7916/$25.00 © 2009 IEEE

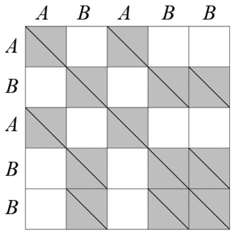


Fig. 1. Example of the structures formed in the self-distance matrix. Darker pixel value denotes lower distance. Time proceeds from left to right and from top to bottom. The example piece consists of five sections, where two parts, A and B, occur as indicated.

blocks themselves can be searched by dynamic programming [14], [15] for segmentation and recurrence analysis.

Some methods utilize the block-like information less explicitly by directly handling the feature vectors with agglomerative clustering [16], or by clustering them with hidden Markov models [17], [18]. The temporal fragmentation resulting from the use of the vector quantization models has been attempted to be reduced by pre-training the model [19], or by imposing duration modeling explicitly [20]–[22].

Because of the assumption of repetition, the sequence methods are not able to describe the entire song, but the parts that are not repeated remain undiscovered. This is not always a weakness, as some methods aim to find the chorus or a representative thumbnail of the piece utilizing the formed stripes. The stripes can be located from the SDM after enhancing them by filtering the matrix [23], [24], or by heuristic rules [25].

In addition to locating only one repeating part, some sequence methods attempt to provide a description of all repeated parts of the piece. By locating all of the repetitions, it is possible to provide a more extensive description of the structure of the piece [1], [26]. Finding a description of the whole piece can be obtained by combining shorter segments with agglomerative clustering [27], refining the segment iteratively [28], selecting repeated segments in a greedy manner [29], or by transitive deduction of segments found utilizing iterative search [30].

The authors of [31] propose to combine vector quantization of framewise features and string matching on the formed sequences to locate repeating parts. Aiming to find a path through the SDM so that the main diagonal is used as little as possible, thus utilizing the off-main diagonal stripes with *ad hoc* rules for piece structures has been attempted in [32]. Heuristic rules to force the piece structure to be one of the few stereotypical ones were presented in [33]. Formulating the properties of a typical or "good" musical piece structure mathematically, and utilizing this formulation to locate a description of the repeated parts has been attempted in [13], [34]. The method proposed in this paper can be seen as an extension of this kind of approach to provide a description of the structure of the whole piece.

B. Proposed Approach

The main novelty of the proposed method is that it relies on a probabilistic fitness measure in analyzing the structure of music pieces. A structure description consists of a segmentation of the piece to occurrences of musical parts, and of grouping of segments that are repeats of each other. The acoustic information of each pair of segments in the description is used to determine the probability that the two segments are repeats of each other. The probabilities are then used to calculate the total fitness of the description. A greedy algorithm is proposed for solving the resulting search problem of finding the structure that maximizes the fitness measured. Furthermore, the resulting description is labeled with musically meaningful part labels. To the authors' knowledge, this is the first time that the labeling can be for arbitrary music pieces.

The proposed method utilizes three acoustic features describing different aspects of the piece. Self-distance matrices are calculated from all the three features, and using the information embedded in the SDM, the system performs a search to create a segmentation and a segment clustering that maximize the fitness over the whole piece. The "blocks" and the "stripes" in multiple SDMs are used.

The rest of the paper is organized as follows. Section II details the proposed method. Then experimental results are described in Section III. Finally, Section IV concludes the paper. Parts of this work have been published earlier in [35]–[37].

II. Proposed Method

The proposed analysis method relies on a fitness function for descriptions of musical structures. This function can be used to compare different descriptions of the same piece and determine how plausible they are from the perspective of the acoustic signal. In addition to the fitness function, a search method for generating a maximally fit description is presented.

A. Fitness Measure

From the point of view of acoustic properties, a good description of musical structure has much in common with defining a good clustering of data points: the intra-cluster similarity should be maximized while minimizing the inter-cluster similarity. In terms of musical structure: the segments assigned to a group (forming the set of all occurrences of a musical part) should be similar to each other while the segments from different groups should be maximally dissimilar. Compared to basic clustering, individual frames of the musical piece cannot be handled as individual data points in clustering, because it would fragment the result temporally, as noted in [21]. Instead, the frames are forced to form sequences.

All the possible segments of a piece are denoted by set $\mathbb{S}$. A subset $\mathbb{S}' \subset \mathbb{S}$ of this consisting of S segments $s_i \in \mathbb{S}'$ that do not overlap and cover the whole piece defines one possible segmentation of the piece. The *group* of segment s_i is returned by a group assignment function $g(\cdot)$; if $g(s_i) = g(s_j)$, the segments belong to the same group and are occurrences of the same musical part. A description $\mathbb{E}$ of the structure of the piece is a combination of a segmentation and grouping of the segments $\mathbb{E} = (\mathbb{S}', g)$.

When a segmentation $\mathbb{S}'$ and the acoustic data is given, it is possible to compare all pairs of segments s_i and s_j, and to determine a probability $\hat{p}(g(s_i) = g(s_j))$ that the segments belong to the same group. Because the segments can be of different lengths, a weighting factor $W(s_i, s_j)$ is determined for each segment pair in addition to the probability. The overall fitness of the description $\mathbb{E}$ is defined as

$$P(\mathbb{E}) = \sum_{s_i \in \mathbb{S}'} \sum_{s_j \in \mathbb{S}'} W(s_i, s_j) l(s_i, s_j, g) \tag{1}$$

where

$$l(s_i, s_j, g) = \begin{cases} \log(\hat{p}(g(s_i) = g(s_j))), & \text{if } g(s_i) = g(s_j) \\ \log(1 - \hat{p}(g(s_i) = g(s_i))), & \text{if } g(s_i) \neq g(s_j). \end{cases} \tag{2}$$

Here, the value of the weighting factor $W(s_i, s_j)$ is defined as

$$W(s_i, s_j) = L(s_i)L(s_j) \tag{3}$$

where $L(s_i)$ denotes the length of segment s_i in frames. This causes the sum of all weighting factors to equal the number of elements in the SDM.

Having defined the fitness measure, the structure analysis problem now becomes a task of finding the description $\mathbb{E}_{OPT}$ that maximizes the fitness function given the acoustic data

$$\mathbb{E}_{OPT} = \arg\max_{\mathbb{E}} \{P(\mathbb{E})\}. \tag{4}$$

Equation (1) defines the fitness of structural descriptions using relatively abstract terms. To apply the fitness measure, candidate descriptions should be constructed for evaluation and the probabilities in (1) and (2) should be calculated from the acoustic input. The rest of this paper describes how these tasks can be accomplished using a system whose block diagram is illustrated in Fig. 2. The system extracts acoustic features using beat-synchronized frame blocking. Separate SDMs are calculated for each feature, to be used as a mid-level representation. Using the information in the SDMs, a large amount of candidate segments is created and all non-overlapping segment pairs are compared. The comparison produces the pairwise probabilities $\hat{p}(g(s_i) = g(s_j))$ and the weights $W(s_i, s_j)$ that are used to evaluate the fitness measure (1). A greedy search algorithm is employed to create description candidates gradually and to evaluate their fitness. The resulting descriptions are labeled using musically meaningful labels, such as verse and chorus. The best description found is then returned. These steps are described in the rest of this section.

B. Feature Extraction

The use of three features is proposed, all of them with two different time scales to provide the necessary information for further analysis. The use of multiple features is motivated by the results of [7], which suggest that change in timbre and in rhythm are important cues for detecting structural boundaries. The use of multiple time scales has been proposed, e.g., in [4] and [38].

The feature extraction starts by estimating the locations of rhythmic beats in the audio using the method from [39]. It was noted that the system may do π-phase errors in the estimation. The effect of these errors is alleviated by inserting extraneous beats between each two beats, effectively halving the pulse period.

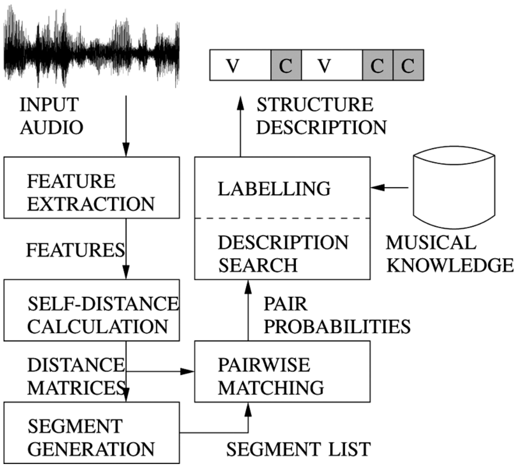


Fig. 2. Overview of the proposed method. See the text for description.

Like in several earlier publications, mel-frequency cepstral coefficients (MFCCs) are used to describe the timbral content of the signal. The rhythmic content is described with *rhythmogram* proposed in [14]. The third feature, chroma, describes the tonal content. The MFCCs and chroma are calculated in 92.9-ms frames with 50% frame overlap, while rhythmogram uses frames up to several seconds in length with the hop of 46.4 ms. After the calculation, each feature is averaged over the beat frames to produce a set of beat-synchronized features.

The MFCCs are calculated using 42-band filter bank, omitting the high-pass pre-emphasis filter sometimes used as a preprocessing. The log-energies of the bands are discrete cosine transformed (DCT) to reduce the correlation between bands and to perform energy compaction. After the DCT step, the lowest coefficient is discarded and 12 following coefficients are used as the feature vector.

The chroma is calculated using the method proposed in [40]. First, the saliences for different fundamental frequencies in the range 80–640 Hz are calculated. The linear frequency scale is transformed into a musical one by selecting the maximum salience value in each frequency range corresponding to a semitone. The semitone number for frequency f is given in MIDI note numbers by

$$F_{\text{MIDI}}(f) = F_{A4} + \left\lfloor 12 \log_2 \left(\frac{f}{f_{A4}} \right) \right\rceil \tag{5}$$

where $F_{A4} = 69$ is the MIDI note number for the reference frequency $f_{A4} = 440$, and $\lfloor \cdot \rceil$ denotes rounding to the nearest integer. Finally, the octave equivalence classes are summed over the whole pitch range to produce a 12-dimensional chroma vector. This method is used instead of directly mapping frequency bins after discrete Fourier transform (as done, e.g., in [1], [23]), because in the experiments the salience estimation front-end proved to focus more on the energy of tonal sounds and reduce some of the undesired noise caused by atonal sounds, such as drums.

For both MFCC and chroma, the feature sequences are temporally filtered with a Hanning window weighted median filter.

The purpose of the filtering is to focus the feature on the desired time-scale. The shorter filter length is used to smooth short-time deviations for enhancing the stripes on the SDM. The longer window length is intended to focus on longer time-scale similarities, enhancing the block formation on the SDMs.

The rhythmogram calculation utilizes the onset accentuation signal produced in the beat detection phase. The original method [14] used a perceptual spectral flux front-end to produce a signal sensitive to sound onsets. In the proposed method, this is replaced by summing the four accentuation signals to produce one onset accentuation signal. The rhythmogram is the autocorrelation function values of the accentuation signal calculated in successive windows after the global mean has been removed from it. The window length is determined by the target time-scale, and the autocorrelation values between the lags 0 and a maximum of 2 s are stored.

The time-scale focus parameters (the median filter window lengths for MFCCs and chroma, and the autocorrelation window length for rhythmogram) were selected with a method described in Section II-E. After the temporal filtering the features are normalized to zero mean and unity variance over the piece.

C. Self-Distance Matrix Calculation

From each feature and time-scale alternative, a self-distance matrix is calculated. Each element $D(i,j)$ of the matrix defines the distance between the corresponding frames i and j calculated with cosine distance measure

$$d(\boldsymbol{v}_i,\boldsymbol{v}_j) = 0.5\left(1-\frac{\langle \boldsymbol{v}_i,\boldsymbol{v}_j\rangle}{||\boldsymbol{v}_i||||\boldsymbol{v}_j||}\right) \tag{6}$$

where $\boldsymbol{v}_i$ is the feature vector in frame i, $\langle\cdot,\cdot\rangle$ denotes vector dot product, and $||\cdot||$ is vector norm.

In many popular music pieces, musical modulation of the key in the last chorus section is used as an effect. This causes problems with the chroma feature as the energies shift to different pitch classes, effectively causing a circular rotation of the chroma vector.[1] To alleviate this problem, it has been proposed to apply chroma vector rotations and calculate several SDMs instead of only one testing all modulations and using the minimum distances [1], [41]. Modulation inversion both on frame and segment pairs were tested, but they did not have a significant effect on the overall performance and the presented results are calculated without them.

D. Segment Border Candidate Generation

Having the SDMs, the system generates a set of segment border candidates that are points in the piece on which a segment may start or end. If a segment is allowed to begin or end at any location, the number of possible segmentations and structural descriptions increases exponentially as a function of the border candidate locations. The combinatorial explosion is reduced by generating a smaller set of border candidates. Not all of the candidates have to be used in the final segmentation, but the points used in the segmentation have to be from this set.

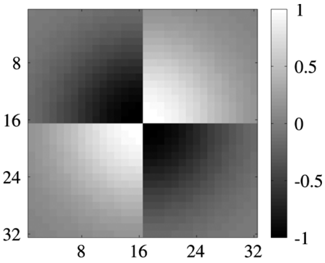


Fig. 3. Example of a Gaussian weighted detection kernel with $m = 32$ and $\sigma = 0.5$.

In the proposed method, the border candidates are generated using the novelty calculation proposed in [11]. A $m\times m$ detection kernel matrix $\boldsymbol{K}$ is correlated along the main diagonal of the SDM. The correlation values are collected to a novelty vector $\boldsymbol{n}$. Peaks in this vector, corresponding to corners in the SDM, are detected using median-based dynamic thresholding and used as the border candidates. The novelty vector is calculated from all six SDMs, three acoustic features and two time-scale parameters, and then summed. For one SDM the novelty is calculated as

$$n(k) = \sum_{i=-m/2}^{m/2-1}\sum_{j=-m/2}^{m/2-1} K\left(\frac{m}{2}+i,\frac{m}{2}+j\right)D(k+i,k+j). \tag{7}$$

The matrix $\boldsymbol{D}$ is padded with zeros in non-positive indices and indices larger than the size of the matrix.

The kernel matrix $\boldsymbol{K}$ has a 2×2 checkerboard-like structure

$$\boldsymbol{K} = \begin{pmatrix} \boldsymbol{Q}_{\mathrm{TL}} & \boldsymbol{Q}_{\mathrm{TR}} \\ \boldsymbol{Q}_{\mathrm{BL}} & \boldsymbol{Q}_{\mathrm{BR}} \end{pmatrix}$$

where the following symmetries hold:

$$\boldsymbol{Q}_{\mathrm{TL}} = -\boldsymbol{J}\boldsymbol{Q}_{\mathrm{BL}} = -\boldsymbol{Q}_{\mathrm{TR}}\boldsymbol{J} = \boldsymbol{J}\boldsymbol{Q}_{\mathrm{BR}}\boldsymbol{J}. \tag{8}$$

Matrix $\boldsymbol{J}$ is an $m/2\times m/2$ matrix with ones on the main anti-diagonal and zeros elsewhere. It reverses the order of matrix columns when applied from right and the order of matrix rows when applied from left.

In the simplest approach, the values in $\boldsymbol{Q}_{\mathrm{TL}}$ are all -1, but as suggested in [11], the kernel matrix values are weighted by radial Gaussian function giving less weight to the values far from the center of the kernel

$$Q_{\mathrm{BR}}(x,y) = -\exp\left(-\frac{r^2}{2\sigma^2}\right) \tag{9}$$

where the radius r is defined by

$$r^2 = \frac{4}{m^2}\left((x-1)^2+(y-1)^2\right) \tag{10}$$

and the width parameter value $\sigma = 0.5$ and kernel width $m = 32$ were noted to perform well in the evaluations. The resulting kernel is illustrated in Fig. 3. In the experiments, the 30 largest peaks in the novelty vector and the signal end points were used as the set of segment border candidates.

[1] Naturally the modulation affects also MFCCs, but the effect is considerably smaller.

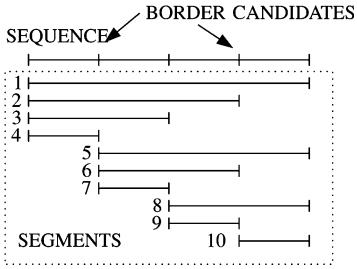


Fig. 4. Illustration of generating the segments.

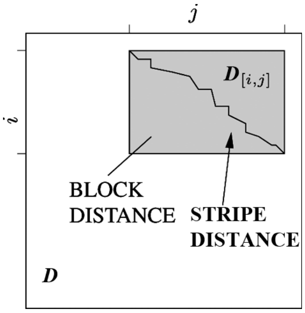


Fig. 5. Submatrix $D_{[i,j]}$ of SDM D used in the calculation of the distances between the segments s_i and s_j.

E. Segment Pair Distance Measures

After the set of border candidates has been generated, all segments between all pairs of border candidates are created. These segments form the set $\mathbb{S}$, from which the segmentation in the final description is a subset of. This is illustrated in Fig. 4, where ten possible segments are generated from five border candidates.

For each segment pair and feature, two distances are calculated: a *stripe distance* and a *block distance*. The stripe distance measures the dissimilarity of the feature sequences of the two segments, whereas the block distance measures the average dissimilarity of all frame pairs of the two segments. Two distance measures are used because it is assumed that they provide complementary information.

The main difference and motivation of using these two distance measures are illustrated in Fig. 1 which contains a stereotypical SDM of a simple piece with the structure "A, B, A, B, B." If only stripe distance was used, it would be difficult to locate the border between "A" and "B" without any additional logic, because "A" is always followed by "B." Similarly, if only block distance was used, the border between the second and third "B" would be missed without any addition logic.

The compared segments s_i and s_j define a submatrix $D_{[i,j]}$ of distance matrix D. The contents of this submatrix are used to determine the acoustic match of the segments. The submatrix and the distance measures are illustrated in Fig. 5.

The block distance $d_B(s_i, s_j)$ is calculated as the average of the distances in the submatrix

$$d_B(s_i, s_j) = \frac{1}{L(s_i)L(s_j)} \sum_{x=1}^{L(s_i)} \sum_{y=1}^{L(s_j)} D_{[i,j]}(x, y). \tag{11}$$

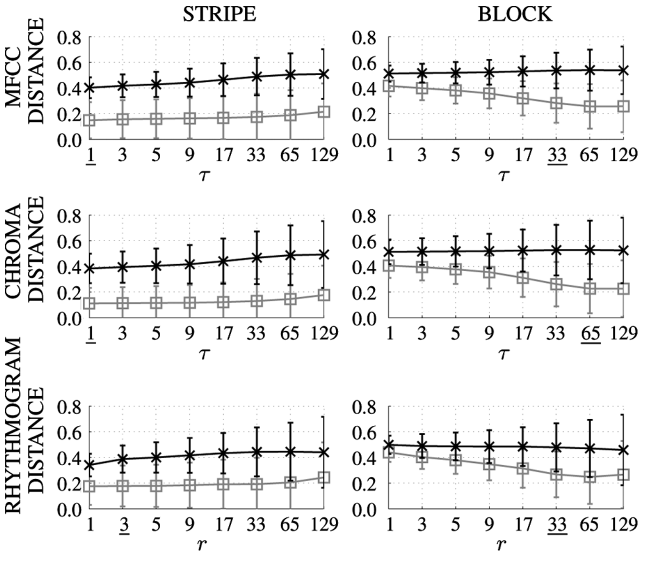


Fig. 6. Effect of the time-scale parameter on segment pair distances calculated over all pieces in the *TUTstructure07* data set. For MFCC and chroma feature the parameter τ is the median filtering window length. For rhythmogram the varied parameter r is the autocorrelation length. The lines denote the average distance values for segments from the same group (□) and from a different group (×). The error bars around the marker denote the standard deviation of the distances. The chosen parameter values are marked with underlining.

The stripe distance $d_S(s_i s_j)$ is calculated by finding the path with the minimum cost through the submatrix $D_{[i,j]}$ and normalizing the value by the minimum possible path length

$$d_S(s_i s_j) = \frac{\tilde{D}_{[i,j]}(L(s_i), L(s_j))}{\max(L(s_i), L(s_j))} \tag{12}$$

where elements of the partial path cost matrix $\tilde{D}_{[i,j]}$ are defined recursively by

$$\tilde{D}_{[i,j]}(x, y) = D_{[i,j]}(x, y) + \min \begin{cases} \tilde{D}_{[i,j]}(x-1, y-1) \\ \tilde{D}_{[i,j]}(x-1, y) \\ \tilde{D}_{[i,j]}(x, y-1) \end{cases} \tag{13}$$

with the initialization $\tilde{D}_{[i,j]}(0, 0) = 0$. Note that the path transitions do not have any associated cost.

The effect of the time-scale parameter on the resulting distance values was evaluated using a manually annotated data set of popular music pieces that will be described in Section III-A. The median filtering window length τ was varied with MFCC and chroma features, and the autocorrelation window length r was varied for rhythmogram. The values of distances for segments from the same groups and from different groups were calculated with both of the proposed distance measures. The effect of the time-scale parameter is illustrated in Fig. 6. The final parameter values used in the evaluations were determined from this data by assuming the distance values to be distributed as Gaussians and selecting the parameter value minimizing the overlapping mass of the distributions. The used parameter values are indicated in the figure.

F. Probability Mapping

Once the distance of two segments has been calculated based on the used features and distance measures, the obtained distance values are transformed to probabilities to enable evaluating the overall fitness measure (1). In the following, both block and stripe distance of a segment pair are denoted with $d(s_i, s_j)$ to simplify the notation and because the processing is similar to both. The probability that two segments s_i and s_j belong to the same group is determined from the distance $d(s_i, s_j)$ between the segments using a sigmoidal mapping function from distance to probability. The mapping function is given by

$$p(g(s_i) = g(s_i)) = (1 + \exp(z_1 d(s_i, s_j) + z_0))^{-1} \qquad (14)$$

where $d(s_i, s_j)$ is the distance measured from the acoustic data. The sigmoid parameters z_1 and z_0 are determined using the Levenberg–Marquardt algorithm for two-class logistic regression [42]. The data for the fit is obtained from the manual ground truth annotations.

The probabilities obtained for all of the six distance values (three acoustic features and two distance measures) are combined with weighted geometric mean

$$\begin{aligned}\hat{p}(g(s_i) =& g(s_j)) \\ &= \left(\prod_b p_b(g(s_i) = g(s_j))^{w_b}\right)^{1/\sum_b w_b} \qquad (15)\end{aligned}$$

where b is a variable distinguishing the six probability values, and w_b is the weight of the corresponding feature and distance combination. In the experiments, binary weights were tested and the presented results are obtained using all but rhythmogram stripe probability with equal weights. For more details on the feature combinations, see [36].

It is possible to impose heuristic restrictions on the segments by adjusting the pairwise probabilities manually after the different information sources have been combined. Here, a length restriction was applied prohibiting larger than 50% differences in segment lengths within a group.

G. *Solution for the Optimization Problem*

The optimization problem (4) is a combinatorial problem. It can be formulated as a path search in a directed acyclic graph (DAG) where each node represents a possible segment in the piece with a specific group assignment, and there is an arc between two nodes only if the segment of the target node is directly following the segment of the source node. This process is illustrated by the graph in Fig. 7 which is constructed from the segments in Fig. 4 after allowing the use of two groups.

The way the total fitness (1) is defined to evaluate all segment pairs in the description causes the arc costs to depend on the whole earlier path, i.e., the transition from a node to a following one has as many different costs as there are possible routes from the start to the source node. This prohibits the use of many efficient search algorithms as problem cannot be partitioned into smaller subproblems.

Considering the applications of the structure analysis system, it would be desirable that the search would be able to produce some solution relatively quickly, to improve it when given more time, and to return the globally optimal result at some point. If the search for the global optimum takes too long, it should be possible to stop the search and use the result found at that point. A novel algorithm named *Bubble token passing* (BTP) is proposed to fulfil these requirements. BTP is inspired by the token passing algorithm [43] often used in continuous speech recognition. In the algorithm, the search state is stored using *tokens* tracking the traveled path and recording the associated fitness. In the following, the term node is changed to *state* to better conform the token passing terminology.

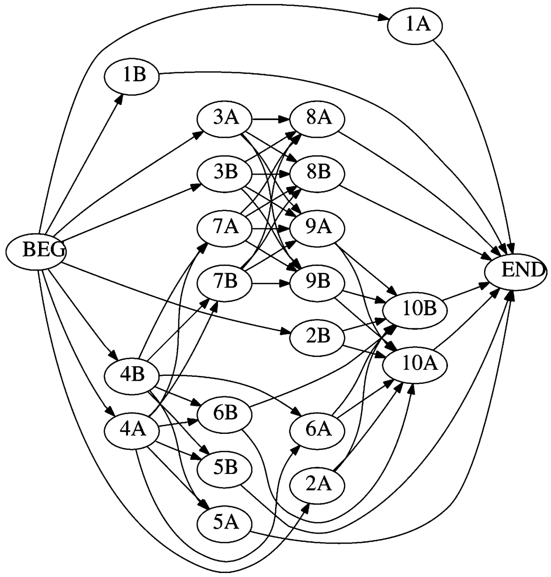


Fig. 7. Example DAG generated by the segments in Fig. 4 after allowing only two groups: A and B.

```
 1: while iter<maxItes & not converged do
 2:   for sourceState in allStates do
 3:     // propagate the $\beta$ best tokens
 4:     for $n = 1$ to $\beta$ do
 5:       // token is propagated to the following states
 6:       for targetState in sourceState.followingStates do
 7:         copyToken = sourceState.tokenList[n].copy()
 8:         copyToken.updatePath(target)
 9:         copyToken.updateCost(target)
10:         targetState.arrivingList.insert(copyToken)
11:       end for
12:     end for
13:     // after propagation, remove the token
14:     sourceState.tokenList.remove(1 ... $\beta$)
15:   end for
16:   for state in allStates do
17:     // merge arrived to storage and store only the best
18:     state.tokenList.mergeWith(arrivingList)
19:     state.tokenList.sort()
20:     state.tokenList[$(\alpha + 1)$ ... end].delete()
21:   end for
22: end while
```

Fig. 8. Pseudo-code description of the proposed bubble token passing search algorithm.

A pseudocode description of the algorithm is given in Fig. 8. The search is initiated by augmenting the formed DAG with start and end states and inserting one token to the start state. After this, the algorithm main loop is executed until the solution converges, some maximum iteration limit is reached, or there are no more tokens in the system. At each iteration, each state selects the β best tokens and propagates them to the following states (loop on line 4 of Fig. 8). When a token is inserted to a state, the

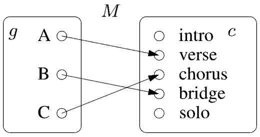


Fig. 9. Labeling process searches for an injective mapping M from a set of segment groups g to musically meaningful labels c.

state is added to the traveled path and the fitness value is updated with (16). After all states have finished the propagation, the arrived tokens are merged to a list of tokens, the list is sorted, and only α fittest are retained, the rest are removed (loop starting on line 16). After this the main iteration loop starts again.

The tokens arriving to the end state describe the found descriptions. The first solutions will be found relatively quickly, and as the iterations proceed, more tokens will "bubble" through the system to the final state. Since the tokens are propagated in best-first order and only some of the best tokens are stored to following iterations, the search is greedy, but the parameters β and α control the greediness and the scope of the search. The number of stored tokens α controls the overall greediness: the smaller the value, the fewer of the less fit partial paths are considered for continuation and more probable it will be to miss the global optimum. An exhaustive search can be accomplished by storing all tokens. The number of propagated tokens β controls the computational complexity of each main loop iteration: the more tokens are propagated from each state, the more rapidly the total number of tokens in the system increases and the more fitness updates have to be calculated at each iteration. The values used in the experiments $(\beta = 10,\ \alpha = 200)$ proved to be a reasonable tradeoff between the exhaustivity and computational cost of the search, and the search converged often after 30–40 iterations.

When a token is inserted to a state corresponding to segment s_i with the group set to $g(s_i)$, the associated path fitness is updated with

$$\Delta P(s_i) = W(s_i, s_i) l(s_i, s_i, g) + 2 \sum_{s_j \in \mathbb{S}'_{i-1}} W(s_j, s_i) l(s_j, s_i, g) \tag{16}$$

where $\mathbb{S}'_i$ is a subset of $\mathbb{S}'$ after adding the ith segment to it, and starting from $\mathbb{S}_0 = \emptyset$.

The fitness of the whole description $\mathbb{E}$ can be obtained by summing these terms over the whole piece

$$P(\mathbb{E}) = \sum_{s_i \in \mathbb{S}'} \Delta P(s_i). \tag{17}$$

It is trivial to verify that this is equal to (1).

H. Musical Part Labelling

The description found by solving the optimization problem (4) consists of a segmentation of the piece and a grouping of the segments. Especially if the analysis result is presented for a human, the knowledge of musically meaningful labels on the segments would be appreciated, as suggested by a user study [44]. To date, none of the structure analysis systems, with the exception of the system proposed in [33], provides musically meaningful labels to the groups in the analysis result. The method in [33] utilized rigid forms where the analyzed piece was fitted to, and the forms contained also the part label information.

The method proposed here models sequences of musical parts with N-grams utilizing the $(N-1)$th order Markov assumption stating that the probability of label c_i given the preceding labels $c_{1:(i-1)}$ depends only on the history of length $N-1$

$$p(c_i | c_{1:(i-1)}) = p(c_i | c_{(i-N+1):(i-1)}). \tag{18}$$

The N-gram probabilities are trained using a set of musical part label sequences that are formed by inspecting the manually annotated structures of a large set of musical pieces. The parts are ordered based on their starting time, and the part labels are set in the corresponding order to produce a training sequence. The N-gram models are then used to find an injective mapping M from the groups g in the analysis result to the musical labels c

$$M : g \rightarrow c. \tag{19}$$

This process is illustrated also in Fig. 9.

When labeling the analysis result, the label assignment maximizing the resulting cumulative N-gram probability over the description

$$p(c_{1:S}) = \prod_{i=1}^{S} p(c_i | c_{(i-N-1):(i-1)}) \tag{20}$$

is searched. An algorithm for the *post-process labeling* a found structural description was presented and evaluated in [35].

Another way to perform the labeling is to integrate the labeling model to the overall fitness function. In this case, the fitness does not only assess the segmentation of the piece and the grouping of the segments, but also the labeling of the groups. The difference to (1) is that now the order in which the segments are evaluated matters, and the segment set $\mathbb{S}'$ needs to be ordered by the starting times of the segments $\mathbb{S}' = (s'_1, s'_2, \ldots, s'_S)$. The description $\mathbb{E}$ can be transformed into a label sequence by applying the mapping function by

$$M(g(s'_{1:i})) \equiv c_{1:i}. \tag{21}$$

The N-gram probabilities have to be evaluated already during the search which is accomplished by modifying the fitness measure (1) to

$$P(\mathbb{E}_{\mathrm{LM}}) = \sum_{i=1}^{S} \sum_{j=1}^{S} W(s'_i, s'_j) l(s'_i, s'_j, g) + \frac{w_l A}{S-1} \sum_{i=1}^{S} \log\left(p(M(g(s'_i)) M(g(s'_{1:(i-1)})))\right) \tag{22}$$

where w_l is the relative weight given for the labeling model, and

$$A = \sum_{i=1}^{S} \sum_{j=1}^{S} W\left(s'_i, s'_j\right). \tag{23}$$

The subscript in $\mathbb{E}_{\mathrm{LM}}$ is added to denote the integrated "labeling model." In effect, the additional term is the average part label transition log-likelihood multiplied by the weighting factors of the segment pairs. The labeling model likelihoods are normalized with the number of transitions. This is done to ensure that explanations with different number of parts would give an equal weight for the labeling model.

Now the fitness function can be considered to be constructed of two terms: the acoustic information term on the top row of (22) and the musicological term on the bottom row. The optimization of this fitness function can be done using the same bubble token passing algorithm after modifying the token fitness update formula (16) to include the N-gram term. In fact, the same search algorithm can be used to perform the postprocess labeling, too. In that case, the acoustic matching terms have to be modified to enforce the grouping sequence.

III. Results

The proposed analysis system was evaluated with simulations using three manually annotated data sets of popular music pieces. Several different evaluation metrics were used to provide different points of view for the system performance.

A. Data

Three data sets were used in the evaluations *TUTstructure07*, *UPF Beatles*, and *RWC Pop*. The first consists of 557 pieces aimed to provide a representative sample of radio-play pieces. Approximately half of the pieces are from pop/rock genres and the rest sample other popular genres, such as hip hop, country, electronic, blues, jazz, and schlager.[2] The data set was compiled and annotated at Tampere University of Technology, and the annotation was done by two research assistants with some musical background. A notable characteristics of the data set is that it contains pieces from broad range of musical styles with differing timbral, melodic, and structural properties.

The second used data set consists of 174 songs by The Beatles. The original piece forms were analyzed and annotated by musicologist Alan W. Pollack [45], and the segmentation time stamps were added at Universitat Pompeu Fabra (UPF).[3] Some minor corrections to the data were made at Tampere University of Technology, and the corrected annotations along with a documentation of the modifications are available.[4] Major characteristic of this data set is that all the pieces are from the same band, with less variation in musical style and timbral characteristics than in the other data sets.

The audio data in the third data set consists of the 100 pieces of the Real World Computing Popular Music Database [46], [47]. All of the pieces were originally produced for the database; a majority of the pieces (80%) represent 1990's Japanese chart music, while the rest resemble the typical 1980s American chart hits.

All data sets contain the structure annotated for the whole piece. Each structural segment is described by its start and end times, and a label provided to it. Segments with the same label are considered to belong to the same group.

B. Reference System

The performance of the proposed system is compared with a reference system [22] aimed for the same task. As the low-level feature it uses the MPEG-7 AudioSpectrumProjection [48] from 600 ms frames with 200-ms hop. The frames are clustered by training a 40-state hidden Markov model on them and then decoding with the same data. The resulting state sequence is transformed to another representation by calculating sliding state histograms from seven consecutive frames. The histograms are then clustered using temporal constraints. The used implementation was from the "QM Vamp Plugin" package version 1.5.[5] The implementation allows the user to select the feature used, the maximum number of different segment types, and minimum length of the segment. A grid search over the parameter space was done to optimize the parameters, and the presented results were obtained using the "hybrid" features, maximum of six segment types, and minimum segment length of 8 s. These parameter values provided the best general performance, and when tested with the same 30-song Beatles data set[6] as in the original publication they produced F-measure of 60.7% compared to the 60.4% reported in [22].

C. Experimental Setup

Because the proposed method needs training of some parameters, the evaluations were run using a tenfold cross-validation scheme with random fold assignment. At each cross-validation fold, 90% of the pieces are used to calculate the N-gram models for part label sequences and to train the distance-to-probability mapping functions, while the remaining 10% are used for testing. The presented results are averaged over all folds. As the reference method [22] does not need training, the evaluations were run for the whole data at once, and different parameter values were tested in a grid search manner.

To allow determining the possible bottlenecks of the proposed system, several evaluation schemes were employed:

- Full analysis. The system is given only the audio; it has to generate the candidate border locations, determine segmentation, grouping, and group labeling. Referred with *full* in the result tables.
- Segmentation and labeling, extraneous borders. The system generates border candidates by itself, but the border locations from the annotations are included in the candidate set by replacing the closest generated candidate with the one taken from annotations. Referred with *salted* in the results.
- Grouping and labeling. The system is given the correct segmentation, but it has to determine the grouping of the segments and labeling of the groups. Referred with *segs* in the tables.
- Labeling only. The correct segmentation and grouping is given to the system. It only has to assign each group with an appropriate musical label. This is referred with *labeling* in the result tables.

[2] A full list of pieces is available at http://www.cs.tut.fi/sgn/arg/paulus/TUT structure07_files.html

[3] http://www.iua.upf.edu/%7Eperfe/annotations/sections/license.html

[4] http://www.cs.tut.fi/sgn/arg/paulus/structure.html#beatles_data

[5] http://www.elec.qmul.ac.uk/digitalmusic/downloads/index.html#qm-vamp-plugins

[6] http://www.elec.qmul.ac.uk/digitalmusic/downloads/#segment

TABLE I
EVALUATION RESULTS ON *TUTSTRUCTURE*07 (%)

method	F	R_P	R_R	labeling	S_O	S_U
annotators	89.4	90.1	89.8	68.0	91.0	91.6
reference [22]	59.9	62.2	60.2	N/A	64.1	67.3
full w/LM	62.4	68.5	62.6	37.9	68.9	71.6
full post-LM	62.6	69.8	62.1	35.3	68.8	72.5
salted w/LM	67.5	73.8	67.7	38.2	74.5	79.0
salted post/LM	67.7	75.5	67.1	37.3	74.5	80.5
segs w/LM	84.4	91.4	81.4	48.1	87.8	94.5
segs post-LM	84.4	91.1	81.5	47.7	87.7	94.2
labeling	N/A	N/A	N/A	62.5	N/A	N/A

TABLE II
EVALUATION RESULTS ON *UPF BEATLES* (%)

method	F	R_P	R_R	labeling	S_O	S_U
reference [22]	58.4	68.3	53.3	N/A	55.2	68.3
full w/LM	59.9	72.9	54.6	35.2	60.4	71.7
full post-LM	58.6	73.7	52.6	26.6	59.3	72.5
labeling	N/A	N/A	N/A	71.8	N/A	N/A

TABLE III
EVALUATION RESULTS ON *RWC POP* (%)

method	F	R_P	R_R	labeling	S_O	S_U
reference [22]	58.1	49.3	73.8	N/A	77.0	60.5
full w/LM	63.7	60.3	72.1	34.4	79.3	72.0
full post-LM	63.3	59.9	72.1	34.3	78.9	71.3
labeling	N/A	N/A	N/A	72.8	N/A	N/A

Two different labeling schemes were tested. First, the labeling was done as a postprocessing step. This is denoted by *post-LM* in the result tables. As an alternative the labeling was integrated in the fitness function using (22). The results obtained with this are referred with *w/LM* in the result tables.

The label set used in all of the tasks is determined from the whole data set prior the cross-validation folds. All part occurrences of all the pieces were inspected and the labels covering 90% of all occurrences were used as the label set. The remaining labels were assigned an artificial "MISC" label.

The proposed system was implemented in Matlab with C++ routines for the feature extraction, the segment matching, and the search algorithm. When run on a 1.86-GHz Intel Core2-based PC, the average analysis time of a piece with the postprocessing labeling corresponds approximately to the duration of the piece.

D. Evaluation Metrics

Three different metrics are used in the evaluations: frame pairwise grouping F-measure (also precision and recall rates from which the F-measure is calculated are reported), conditional entropy based measure for over- and under-segmentation, and total portion of frames labeled correctly.

The first measure is also used in [22]. It considers all frame pairs both in the ground truth annotations and in the analysis result. If both frames in a pair have the same group assignment, the pair belongs to the set $\mathbb{F}_A$ in the case on ground truth and to $\mathbb{F}_\mathbb{E}$ in the case of analysis result. The pairwise precision rate is defined as

$$R_P = \frac{|\mathbb{F}_A \cap \mathbb{F}_\mathbb{E}|}{|\mathbb{F}_\mathbb{E}|} \tag{24}$$

the pairwise recall rate as

$$R_R = \frac{|\mathbb{F}_A \cap \mathbb{F}_\mathbb{E}|}{|\mathbb{F}_\mathbb{E}|} \tag{25}$$

and the pairwise F-measure as their harmonic mean

$$F = \frac{2R_P R_R}{R_P + R_R}. \tag{26}$$

In the equations above $|\cdot|$ denotes the cardinality of the set. The pairwise clustering measure is simple, yet effective and seems to provide values that agree quite well with the subjective performance.

The second evaluation measure considers the conditional entropy of the frame sequences labeled with the group information given the other sequence (ground truth versus result). The original entropy-based evaluation measure was proposed in [49], but it was further modified by adding normalization terms to allow more intuitive interpretation of the obtained numerical values in [50]. The resulting evaluation measures are over-segmentation score S_O and under-segmentation score S_U. Due to their complexity the formal definitions of S_O and S_U are omitted here, see [50] instead.

The third evaluation metric is the strictest: it evaluates the absolute analysis performance with musical labels. This is done by comparing the label assigned to each frame in the result and in the ground truth annotations. The evaluation measure is the proportion of correctly recovered frame labels.

E. Annotation Reliability Check

It has been noted in earlier studies, e.g., in [7], that the perception of structure in music varies from person to person; therefore, a small experiment was conducted to obtain an estimate of the theoretically achievable accuracy level. A subset of 30 pieces in the *TUTstructure07* data set was analyzed by both annotators independently. Then one set of annotations was considered as the ground truth while the other was evaluated against it. Despite the small size of the data set, this provides an approximation of the level of "human-like performance."

F. Evaluation Results

Tables I–III show the main evaluation results on the different data sets. When comparing the results of tasks with different segmentation levels, the results suggest that the segment border candidate generation is a crucial step for the overall performance. If there are too many extraneous candidate locations, as the case is in "salted" case, the performance drops. The difference between "salted" and "full" is surprisingly small, suggesting that the border candidate generation is able to recover the candidate locations relatively accurately.

The performance increase from the reference system is statistically significant ($p < 0.05$) in the data sets of *TUTstructure07* and *RWC Pop*, but not in *UPF Beatles*. The performance difference between postprocessing labeling and integrated labeling is not significant when evaluated with pairwise F-measure or with over- and under-segmentation measures. Based on the labeling measure, the improvement with integrated labeling in *TUTstructure07* and *UPF* Beatles data sets is statistically significant, whereas in *RWC Pop* it is not.

TABLE IV
SEGMENT BOUNDARY RETRIEVAL PERFORMANCE (%)

data	system	F	R_P	R_R
TUTstructure07	reference [22]	65.3	70.5	63.3
	full w/LM	55.9	50.7	65.9
UPF Beatles	reference [22]	61.2	60.0	64.6
	full w/LM	55.0	52.1	61.2
RWC Pop	reference [22]	64.5	77.3	56.6
	full w/LM	63.0	71.7	57.8

TABLE V
SEGMENTATION STATISTICS ON THE USED DATA SETS

data		segments	groups	seg. dur. (s)
	annotation	12.1	6.01	19.6
TUTstructure07	reference [22]	10.7	5.11	21.5
	full w/LM	12.1	6.98	20.0
	annotation	8.57	4.59	19.2
UPF Beatles	reference [22]	9.48	4.72	17.4
	full w/LM	10.3	6.21	16.8
	annotation	17.1	9.10	14.7
RWC Pop	reference [22]	12.2	5.06	20.6
	full w/LM	13.4	7.96	19.1

Table IV presents the segment boundary retrieval results for both systems on all data sets. A boundary in the result is judged as a hit if it is within 3 s from the annotated border as suggested in [22] and [28].

More direct analysis of the annotated structures and the obtained results is provided in Table V. The table provides the average number of segments in the pieces in the data sets, the average number of groups, and the average duration of a segment. The reference system groups the generated segments using fewer groups than was annotated, while the proposed system uses extraneous groups. Similar under-grouping behavior of the proposed system can be seen in the statistics for *UPF Beatles*. Both systems under-segment the result in *RWC Pop*. This may be partly because the structures in the data have more and shorter segments.

A detailed analysis on the labeling performance is given in Tables VI–VIII. The values describe for each ground truth label the average amount of its duration that was correctly recovered in the result, e.g., value 50% denotes that, on the average, half of the frames with that label were assigned the same label in the result. The tables present the result on all data sets in percents for the labeling only task and for the full analysis with integrated labeling model. The labels are ordered in descending order by their occurrences, the most frequently occurring on top.

TABLE VI
PER LABEL RECOVERY ON *TUTSTRUCTURE*07 (%)

label	labeling	full w/LM
chorus	77.9	57.6
verse	64.3	44.2
MISC	26.5	10.1
bridge	48.6	17.9
intro	99.2	77.7
pre-verse	55.0	15.8
outro	99.0	60.2
c	47.0	25.9
solo	8.2	5.0
theme	2.2	0.5
chorus_a	7.1	1.8
a	24.4	11.1
chorus_b	6.8	5.1

TABLE VII
PER LABEL RECOVERY ON *UPF BEATLES* (%)

label	labeling	full w/LM
verse	83.9	38.4
refrain	54.8	26.6
bridge	81.4	34.6
intro	94.6	83.2
MISC	16.9	9.6
outro	99.3	62.5
verses	50.0	27.8
versea	9.1	20.5

TABLE VIII
PER LABEL RECOVERY ON *RWC POP* (%)

label	labeling	full w/LM
chorus a	75.0	41.0
verse a	76.0	45.7
MISC	52.3	22.9
verse b	73.9	25.1
chorus b	73.2	16.6
bridge a	61.5	24.2
intro	100.0	87.7
ending	100.0	61.5
pre-chorus	40.0	8.7
verse c	43.6	6.5

G. Discussion

When comparing the results of different data sets, the differences in the material become visible. The performance of the proposed method measured with the F-measure quite similar in all data sets, but the recall and precision rates differ greatly: in *TUTstructure07* the two are close to each other, in *UPF Beatles* the method over-segments the result, and in *RWC Pop* the result is under-segmented. As the operational parameters were selected based on the *TUTstructure07* data, this suggests that some parameter selection should be done for differing material.

Some of the earlier methods tend to over-segment the result and the segment duration had to be assigned in the method "manually," e.g., the reference method [22]. From this point of view it is encouraging to note how the proposed method is able to locate approximately correct length segments even though there is no explicit information given of the appropriate segment length. However, the segment length accuracy differences between the data sets suggest that some additional information should be utilized to assist determining the correct segment length.

It can be noted from Table I that the human baseline for the performance given by the annotator cross-evaluation is surprisingly low. Closer data analysis revealed that a majority of the differences between the annotators was due to hierarchical level differences. Some differences were also noted when a part occurrences contained variations: one annotator had used the same label for all of the occurrences, while the other had created a new group for the variations. It can be assumed that similar differences would be encountered also with larger population analyzing same pieces.

IV. CONCLUSION

A system for automatic analysis of the sectional form of popular music pieces has been presented. The method creates several candidate descriptions of the structure and selects the best by evaluating a fitness function on each of them. The resulting optimization problem is solved with a novel controllably greedy search algorithm. Finally, the segments are assigned with musically meaningful labels.

An important advantage of the proposed fitness measure approach is that it distinguishes the definition of a good structure description from the actual search algorithm. In addition, the fitness function can be defined on a high abstraction level, without committing to specific acoustic features, for example. The system was evaluated on three large data sets with manual annotations and it outperformed a state-of-the-art reference method. Furthermore, assigning musically meaningful labels to the description is possible to some extent with a simple sequence model.

Acknowledgment

The authors would like to thank M. Levy for the assistance on his reference system.

References

[1] M. Goto, "A chorus-section detecting method for musical audio signals," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Hong Kong, 2003, pp. 437–440.

[2] T. Jehan, "Creating music by listening," Ph.D. dissertation, Mass. Inst. of Technol., Cambridge, MA, 2005.

[3] V. M. Rao, "Audio compression using repetitive structures in music," M.S. thesis, Univ. of Miami, Miami, FL, 2004.

[4] M. Marolt, "A mid-level melody-based representation for calculating audio similarity," in *Proc. 7th Int. Conf. Music Inf. Retrieval*, Victoria, B.C, Canada, Oct. 2006, pp. 280–285.

[5] E. Gómez, B. Ong, and P. Herrera, "Automatic tonal analysis from music summaries for version identification," in *Proc. 12st Audio Eng. Soc. Conv.*, San Francisco, CA, Oct. 2006.

[6] T. Zhang and R. Samadani, "Automatic generation of music thumbnails," in *Proc. IEEE Int. Conf. Multimedia Expo*, Beijing, China, Jul. 2007, pp. 228–231.

[7] M. J. Bruderer, M. McKinney, and A. Kohlrausch, "Structural boundary perception in popular music," in *Proc. 7th Int. Conf. Music Inf. Retrieval*, Victoria, BC, Canada, Oct. 2006, pp. 198–201.

[8] J.-P. Eckmann, S. O. Kamphorst, and D. Ruelle, "Recurrence plots of dynamical systems," *Europhys. Lett.*, vol. 4, no. 9, pp. 973–977, Nov. 1987.

[9] J. Foote, "Visualizing music and audio using self-similarity," in *Proc. ACM Multimedia*, Orlando, Fl, 1999, pp. 77–80.

[10] G. Peeters, "Deriving musical structure from signal analysis for music audio summary generation: Sequence and state approach," in *Lecture Notes in Computer Science*. New York: Springer-Verlag, 2004, vol. 2771, pp. 143–166.

[11] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proc. IEEE Int. Conf. Multimedia Expo*, New York, Aug. 2000, pp. 452–455.

[12] M. Cooper and J. Foote, "Summarizing popular music via structural similarity analysis," in *Proc. 2003 IEEE Workshop Applicat. Signal Process. Audio Acoust.*, New Platz, NY, Oct. 2003, pp. 127–130.

[13] J. Paulus and A. Klapuri, "Music structure analysis by finding repeated parts," in *Proc. 1st ACM Audio Music Comput. Multimedia Workshop*, Santa Barbara, CA, Oct. 2006, pp. 59–68.

[14] K. Jensen, "Multiple scale music segmentation using rhythm, timbre, and harmony," *EURASIP J. Adv. Signal Process.*, 2007, article ID 73205.

[15] M. M. Goodwin and J. Laroche, "A dynamic programming approach to audio segmentation and music/speech discrimination," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Montreal, QC, Canada, May 2004, pp. 309–312.

[16] C. Xu, X. Shao, N. C. Maddage, M. S. Kankanhalli, and T. Qi, "Automatically summarize musical audio using adaptive clustering," in *Proc. IEEE Int. Conf. Multimedia Expo*, Taipei, Taiwan, Jun. 2004.

[17] B. Logan and S. Chu, "Music summarization using key phrases," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Istanbul, Turkey, Jun. 2000, pp. 749–752.

[18] J.-J. Aucouturier and M. Sandler, "Segmentation of musical signals using hidden Markov models," in *Proc. 110th Audio Eng. Soc. Conv.*, Amsterdam, The Netherlands, May 2001.

[19] S. Gao, N. C. Maddage, and C.-H. Lee, "A hidden Markov model based approach to music segmentation and identification," in *Proc. 4th Pacific Rim Conf. Multimedia*, Singapore, Dec. 2003, pp. 1576–1580.

[20] S. Abdallah, M. Sandler, C. Rhodes, and M. Casey, "Using duration models to reduce fragmentation in audio segmentation," *Mach. Lear.*, vol. 65, no. 2–3, pp. 485–515, Dec. 2006.

[21] M. Levy, M. Sandler, and M. Casey, "Extraction of high-level musical structure from audio data and its application to thumbnail generation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Toulouse, France, May 2006, pp. 13–16.

[22] M. Levy and M. Sandler, "Structural segmentation of musical audio by constrained clustering," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 2, pp. 318–326, Feb. 2008.

[23] M. A. Bartsch and G. H. Wakefield, "Audio thumbnailing of popular music using chroma-based representations," *IEEE Trans. Multimedia*, vol. 7, pp. 96–104, Feb. 2005.

[24] A. Eronen, "Chorus detection with combined use of MFCC and chroma features and image processing filters," in *Proc. 10th Int. Conf. Digital Audio Effects*, Bordeaux, France, Sep. 2007, pp. 229–236.

[25] L. Lu and H.-J. Zhang, "Automated extraction of music snippets," in *Proc. ACM Multimedia*, Berkeley, CA, Nov. 2003, pp. 140–147.

[26] R. B. Dannenberg and N. Hu, "Pattern discovery techniques for music audio," in *Proc. 3rd Int. Conf. Music Inf. Retrieval*, Paris, France, Oct. 2002, pp. 63–70.

[27] W. Chai, "Automated analysis of musical structure," Ph.D. dissertation, Mass. Inst. of Technol., Cambridge, MA, 2005.

[28] B. S. Ong, "Structural analysis and segmentation of musical signals," Ph.D. dissertation, UPF, Barcelona, Spain, 2006.

[29] G. Peeters, "Sequence representation of music structure using higher-order similarity matrix and maximum-likelihood approach," in *Proc. 8th Int. Conf. Music Inf. Retrieval*, Vienna, Austria, Sep. 2007, pp. 35–40.

[30] M. Müller and F. Kurth, "Towards structural analysis of audio recordings in the presence of musical variations," *EURASIP J. Adv. Signal Process.*, 2007, article ID 89686.

[31] C. Rhodes and M. Casey, "Algorithms for determining and labeling approximate hierarchical self-similarity," in *Proc. 8th Int. Conf. Music Inf. Retrieval*, Vienna, Austria, Sep. 2007, pp. 41–46.

[32] Y. Shiu, H. Jeong, and C.-C. J. Kuo, "Similarity matrix processing for music structure analysis," in *Proc. 1st ACM Audio Music Comput. Multimedia Workshop*, Santa Barbara, CA, Oct. 2006, pp. 69–76.

[33] N. C. Maddage, "Automatic structure detection for popular music," *IEEE Multimedia*, vol. 13, no. 1, pp. 65–77, Jan. 2006.

[34] E. Peiszer, "Automatic Audio Segmentation: Segment Boundary and Structure Detection in Popular Music," M.S. thesis, Vienna Univ. of Technol., Vienna, Austria, 2007.

[35] J. Paulus and A. Klapuri, "Labelling the structural parts of a music piece with Markov models," in *Proc. Comput. in Music Modeling and Retrieval Conf.*, Copenhagen, Denmark, May 2008, pp. 137–147.

[36] J. Paulus and A. Klapuri, "Acoustic features for music piece structure analysis," in *Proc. 11th Int. Conf. Digital Audio Effects*, Espoo, Finland, Sep. 2008, pp. 309–312.

[37] J. Paulus and A. Klapuri, "Music structure analysis with probabilistically motivated cost function with integrated musicological model," in *Proc. 9th Int. Conf. Music Information Retrieval*, Philadelphia, PA, Sep. 2008, pp. 369–374.

[38] D. Turnbull, G. Lanckriet, E. Pampalk, and M. Goto, "A supervised approach for detecting boundaries in music using difference features and boosting," in *Proc. 8th Int. Conf. Music Inf. Retrieval*, Vienna, Austria, Sep. 2007, pp. 51–54.

[39] A. Klapuri, A. Eronen, and J. Astola, "Analysis of the meter of acoustic musical signals," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 1, pp. 342–355, Jan. 2006.

[40] M. P. Ryynänen and A. P. Klapuri, "Automatic transcription of melody, bass line, and chords in polyphonic music," *Comput. Music J.*, vol. 32, no. 3, pp. 72–86, 2008.

[41] M. Müller and M. Clausen, "Transposition-invariant self-similarity matrices," in *Proc. 8th Int. Conf. Music Inf. Retrieval*, Vienna, Austria, Sep. 2007, pp. 47–50.

[42] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*, A. J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA: MIT Press, 1999.

[43] S. J. Young, N. H. Russell, and J. H. S. Thornton, "Token passing: A simple conceptual model for connected speech recognition systems," Cambridge Univ. Eng. Dept., Cambridge, U.K., 1989, Tech. Rep. CUED/F-INFENG/TR38.

[44] G. Boutard, S. Goldszmidt, and G. Peeters, "Browsing inside a music track, the experimentation case study," in *Proc. 1st Workshop Learn. Semantics of Audio Signals*, Athens, Greece, Dec. 2006, pp. 87–94.

[45] A. W. Pollack, "Notes on… series," The Official rec.music.beatles Home Page, 1989–2001. [Online]. Available: http://www.recmusicbeatles.com

[46] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical, and jazz music databases," in *Proc. 3rd Int. Conf. Music Inf. Retrieval*, Paris, France, Oct. 2002, pp. 287–288.

[47] M. Goto, "AIST annotation for the RWC music database," in *Proc. 7th Int. Conf. Music Inf. Retrieval*, Victoria, BC, Canada, Oct. 2006, pp. 359–360.

[48] M. Casey, "General sound classification and similarity," *MPEG-7, Organized Sound*, vol. 6, no. 2, pp. 153–164, 2001.

[49] S. Abdallah, K. Nolad, M. Sandler, M. Casey, and C. Rhodes, "Theory and evaluation of a Bayesian music structure extractor," in *Proc. 6th Int. Conf. Music Inf. Retrieval*, London, U.K., Sep. 2005.

[50] H. Lukashevich, "Towards quantitative measures of evaluating song segmentation," in *Proc. 9th Int. Conf. Music Inf. Retrieval*, Philadelphia, PA, Sep. 2008, pp. 375–380.

Jouni Paulus (S'06) received the M.Sc. degree from the Tampere University of Technology (TUT), Tampere, Finland, in 2002. He is currently pursuing a postgraduate degree at the Department of Signal Processing, TUT.

He has been as a Researcher at TUT since 2002. His research interests include signal processing methods and machine learning for music content analysis, especially automatic transcription of drums and music structure analysis.

Anssi Klapuri (M'06) received the M.Sc. and Ph.D. degrees from the Tampere University of Technology (TUT), Tampere, Finland, in 1998 and 2004, respectively.

In 2005, he spent six months at the Ecole Centrale de Lille, Lille, France, working on music signal processing. In 2006, he spent three months visiting the Signal Processing Laboratory, Cambridge University, Cambridge, U.K. He is currently a Professor at the Department of Signal Processing, TUT. His research interests include audio signal processing, auditory modeling, and machine learning.

Appendix A

Dynamic Time Warping

Dynamic time warping (DTW) is a dynamic programming algorithm for time-aligning two sequences, and it was used in speech recognition before HMMs gained success [Rab93]. The general idea is that there are two sequences $X = \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_\mathrm{X}}$ and $Y = \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{N_\mathrm{Y}}$, and a distance function $d(\boldsymbol{x}_i, \boldsymbol{y}_j)$ to match two elements of the sequences. The problem is to determine the optimal alignment between the two sequences so that the total mismatch between the aligned sequences is minimised. This optimisation is solved by dividing the problem into smaller sub-problems that are solved optimally and then the solution is extended at the larger level. The matching cost $d_\mathrm{DTW}(i,j)$ of subsequences up to indices i and j is defined recursively as

$$d_\mathrm{DTW}(i,j) = d(\boldsymbol{x}_i, \boldsymbol{y}_j) + \min \begin{cases} C(1,1) + d_\mathrm{DTW}(i-1, j-1) \\ C(1,0) + d_\mathrm{DTW}(i-1, j) \\ C(0,1) + d_\mathrm{DTW}(i, j-1) \end{cases}, \tag{A.1}$$

where $C(\cdot,\cdot)$ is the transition cost for the local path step. The total cost of the optimal alignment is returned by $d_\mathrm{DTW}(N_\mathrm{X}, N_\mathrm{Y})$, and the actual alignment can be obtained by storing the transition decisions and then backtracking the path to the beginning. The given definition (A.1) with three possible path continuations is only one of the many possible local path constraints, but perhaps the simplest to understand. The same principle is behind the Viterbi algorithm used to decode HMMs: the matched sequences are only changed to represent possible states and the observation sequence. The local distance $d(\boldsymbol{x}_i, \boldsymbol{y}_j)$ is replaced by the log-likelihood of the ith state to have produced the jth observation, the transition cost is replace the the state transition log-probabilities, and instead of minimising the function, it is maximised.

Appendix B

Complexity Analysis of Structural Descriptions

FOR B border candidates and keeping end points fixed, there are $\begin{pmatrix} B-2 \\ b \end{pmatrix}$ possible ways to select b borders to produce the segmentation. In total, there are

$$H(B) = \sum_{b=0}^{B-2} \begin{pmatrix} B-2 \\ b \end{pmatrix} = 2^{B-2} \tag{B.1}$$

ways to segment the piece. The Stirling number of second kind states that given $S = b+1$ segments, they can be distributed into G non-empty groups in

$$\left\{ \begin{matrix} S \\ G \end{matrix} \right\} = \frac{1}{G!} \sum_{i=0}^{G} (-1)^{G-i} \begin{pmatrix} G \\ i \end{pmatrix} i^S \tag{B.2}$$

ways [Gol72, p. 824]. Therefore, having B border candidates, there are

$$C(B) = \sum_{b=0}^{B-2} \begin{pmatrix} B-2 \\ b \end{pmatrix} \sum_{G=1}^{b+1} \left\{ \begin{matrix} b+1 \\ G \end{matrix} \right\} \tag{B.3}$$

possible segmentations and groupings of the segments.

Setting a value for the maximum number G_{MAX} of different groups that can be created, (B.3) is modified slightly to

$$C(B, G_{\text{MAX}}) = \sum_{b=0}^{B-2} \begin{pmatrix} B-2 \\ b \end{pmatrix} \sum_{G=1}^{\min(b+1, G_{\text{MAX}})} \left\{ \begin{matrix} b+1 \\ G \end{matrix} \right\}. \tag{B.4}$$

If the labelling is included in the fitness function (3.24) and the labels can be assigned from a set of size J, there are

$$C(B, G_{\text{MAX}}, J) = \\ \sum_{b=0}^{B-2} \begin{pmatrix} B-2 \\ b \end{pmatrix} \sum_{G=1}^{\min(b+1, G_{\text{MAX}})} \left\{ \begin{matrix} b+1 \\ G \end{matrix} \right\} \frac{J!}{\max(0, J-G)!} \tag{B.5}$$

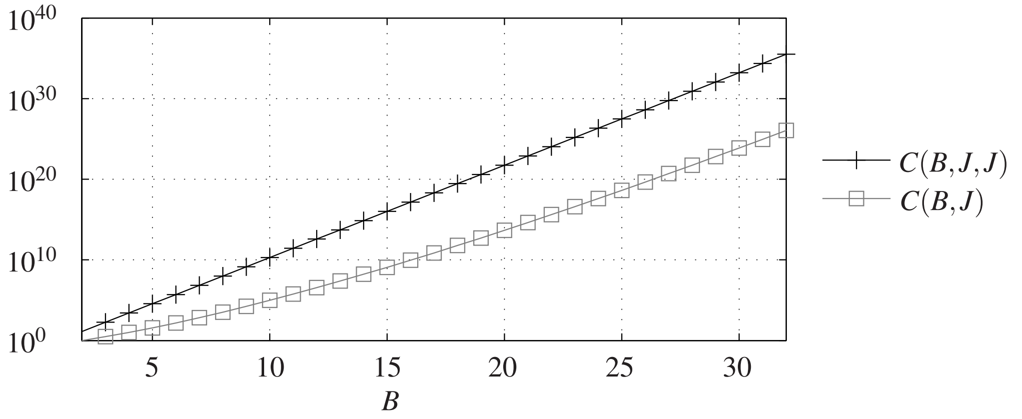


Figure B.1: Number of descriptions as a function of number of border candidates in the case the labelling is integrated in the search (+) and in post-process labelling after only restricting the maximum number of segment groups (□) when the number of possible labels is set to $J = 13$.

different structural descriptions. If the maximum number of usable groups is determined from the number of possible labels, i.e., $G_{\mathrm{MAX}} = J$ the above can be simplified into

$$C(B,J,J) \\ = \sum_{b=0}^{B-2} \left(\begin{array}{c} B-2 \\ b \end{array} \right) \sum_{G=1}^{\min(b+1,J)} \left\{ \begin{array}{c} b+1 \\ G \end{array} \right\} \frac{J!}{(J-G)!}. \tag{B.6}$$

The search space size increase caused by the integrated labelling is illustrated in Fig. B.1. The graphs represent the number of different descriptions as a function of the border candidate count. The maximum number of groups G_{MAX} was set to 13 to correspond to the value used in the evaluation with *TUTstructure07* data set. The line marked with + is the number of descriptions if the labelling is integrated to the search, and the line marked with □ is the number of descriptions if the labelling is done as post-processing. The computational load of the post-process labelling is negligible compared to the description search and is omitted from the graphs.

The time is gone.
The song is over.
Thought I'd something more to say.

Time, R. WATERS 1973