universität wien

MASTERARBEIT

Titel der Masterarbeit

"Unpitched Percussion Transcription in Audio Signals"

Verfasser
Alex Maximilian Wöhrer

angestrebter akademischer Grad
Diplom-Ingenieur (Dipl.-Ing.)

Wien, 2009

Studienkennzahl lt. Studienblatt:	A 066 935
Studienrichtung lt. Studienblatt:	Master Medieninformatik
Betreuer:	Univ. Prof. Dr. Wolfgang Klas
	ao. Univ. Prof. Dr. Andreas Rauber
Mitbetreuer:	Mag. Stefan Leitich
	Dipl.Ing. Thomas Lidy

Kurzfassung

Die Forschung im Bereich der inhaltsbasierten Beschreibung von Musik hat an Bedeutung gewonnen, seitdem die Menge von digital erhältlicher Musik unüberschaubar geworden ist. Eines der interessantesten und schwierigsten Probleme, unter der Vielzahl von Disziplinen in diesem Feld, ist das der Musiktranskription. Dieser Begriff bezeichnet im weitesten Sinn die zeitliche Erkennung musikalischer Ereignisse und die Benennung der daran beteiligten Instrumente. Bisherige Forschung in diesem Bereich konzentrierte sich hauptschlich auf die Extraktion von melodischen und tonalen Informationen, bis vor kurzem der Extraktion von rhythmischen Strukturen derselbe Stellenwert beigemessen wurde. Da Schlaginstrumente das rhythmische Rückgrat eines musikalischen Stückes bilden, spielt deren Transkription eine entscheidende Rolle bei der Darstellung und dem Verständnis von Musik.

Diese Masterarbeit beschreibt angewendete Signalverarbeitungstechniken im Bereich der Transkription von Schlaginstrumenten und stellt ein Vorlagen-basiertes Verfahren im Detail vor, das im Verlauf dieser Masterarbeit implementiert und getestet worden ist.

Abstract

Content-based description of music has become a significant research topic, since technological advances let the amount of digitally available music explode. One of the most interesting and challenging problems, among the wide range of disciplines in this field, is that of music transcription. The term transcription refers to the task of estimating the temporal locations of sound events and recognising the instruments which have been used to produce them. Research in this discipline primarily focused on the extraction of melodic and tonal information, until more recently the extraction of rhythmic structures received the same degree of attention. As percussive instruments form the rhythmic backbone of a musical piece, their transcription is a key component in representing and understanding music.

This thesis explores state of the art signal processing techniques that have found application in percussion transcription and describes a template-matching-based transcription system in more detail, which has been implemented and evaluated in the course of this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The human auditory system is a remarkable information processing system. From just two input channels we are able to identify and extract valuable information from a variety of simultaneous sounds. Attempts to mimic this ability with computers have been studied under interdisciplinary research areas like Computational Auditory Scene Analysis (CASA) and Music Information Retrieval (MIR), where CASA focuses on the recognition of various sounds with the help of sound stream segregation, and MIR explores technologies to extract information from musical content. An interesting subset of these two research areas is that of music transcription.

Transcription is the act of listening to a piece of music and writing down musical notation that corresponds to events being present in that piece. By means of MIR, this process can be described as retrieval of structural data for the symbolic representation of melody and rhythm. Although research in this field is predominantly shaped by the transcription of melodic and harmonic instruments, it should be noted that rhythm produced by percussive instruments is an essential concept for musical structure. That is why the transcription of percussive instruments is receiving more and more interest by the research community, leading to numerous new approaches and algorithms to solve this task.

1.1 Motivation

We are all to a greater or lesser extent able to memorize the melody or beat of a song, but for most people it is far from easy to write the heard structures down. The cause lies within a transcription problem whose solution requires certain skills including musical knowledge, patience, and a lot of experience. Therefore, an automated transcription system would be useful to simplify the extensive transcription process. People without necessary skills could attempt transcriptions themselves for learning or other purposes. As learning aid it would allow people to play to a piece of music where only the recording and no notation is available. There are even more potential applications for the automated transcription of percussive instruments, for example in areas of MIR. The automated transcription of percussive events could be utilised in all systems that rely on the identification of certain structures. For instance, it would support query-by-humming systems, whereby the user hums or plays a piece of music and the computer attempts to identify the piece. Another example is genre classification, where the identification of typical drum patterns unique

to musical styles could be used to determine the correct genre of a musical piece. In the same manner, music similarity metrics could be enhanced with percussive information to improve the grouping of similar music. Another possible application is audio segmentation, where aims to retrieve the structure of a song could be supported by the detection of rhythmical pattern changes, since these are a strong indicator for structural boundaries. Further, the extraction of drum events is helpful for obtaining information about tempo and metrical structure, which are important for numerous MIR-related tasks.

1.2 Scope of Work

This master thesis deals with the creation of systems for the transcription of percussive instruments. It was decided to limit the set of percussive instruments to be transcribed to those instruments found in a typical rock/pop drum kit, and furthermore concentrate only on instruments that are primarily utilised to create a basic drum beat. These drums include the bass drum (also known as kick drum; these two names are used synonymously throughout this thesis), the snare drum, and the hi-hat. The limitation to these instruments is also motivated by the decision that it is more desirable to transcribe a small number of drums robustly than a larger number with less accuracy.

To avoid misunderstandings, this paragraph explains several issues of importance: First, the term transcription, in the context of this thesis, does not fully equal with the preliminarily presented explanation. It is rather taken to be simply a list of pairs, each pair being a combination of instrument type and the time at which that instrument occurred. The reason for this proceeding is that it is not a trivial task to map the transcription results to a metric grid and align the musical events in a meaningful manner into bars, because of the ambiguity of time signatures. However, the generated list can be seen as starting ground for future efforts to create some form of symbolic notation. Second, the term 'polyphonic' within this thesis is used to mean the occurrence of two or more sound sources simultaneously. Similarly the term 'monophonic' means that only one sound source occurs at a time. Third, in the following sections the term 'drum' is used to encompass all instruments in a standard drum kit.

1.3 Structure

As this thesis mainly deals with the instruments found in a typical rock/pop drum kit, it is adequate to start with a description of these instruments in chapter 2. This chapter explains the classification of drum kit instruments and their characteristic properties.

Chapter 3 aims to give an overview of existing approaches and state of the art methods for percussive instrument transcription. The chapter is based on the contents of [9].

Chapter 4 introduces some well-known work on transcription systems and briefly describes two systems that were reviewed in the course of research.

Chapter 5 describes a percussion transcription system based on [32], which was implemented in the course of this thesis. The chapter describes the overall design of the transcription system and includes a detailed explanation of all stages with comments on their realisation.

Chapter 6 deals with the evaluation of the implemented percussion transcription system, whereby the attention is focused on implementation-specific parameters. The chapter includes a description of the calculated performance measures, the utilised ground truth and the performed test cases.

Chapter 7 then contains comments and conclusions on the work done and highlights possible improvements for future research in the field of percussion transcription.

Chapter 2

The Drum Kit

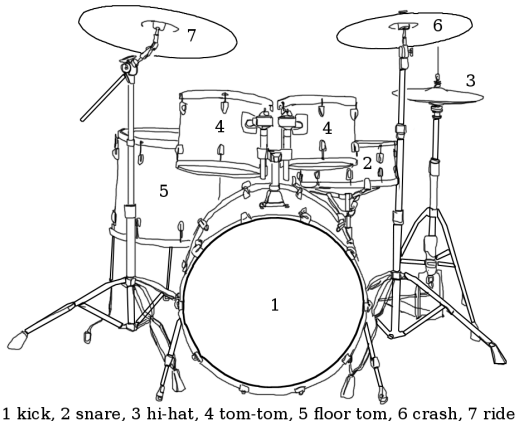


Figure 2.1: A typical rock/pop drum kit. [1]

A typical rock/pop drum kit in Western popular music is shown in figure 2.1. A standard setup includes a bass drum, a snare drum, a hi-hat, one or more tom-toms, and a crash and ride cymbal. These instruments can be divided into two groups: membranophones and idiophones.

2.1 Membranophones

Membranophones are instruments consisting of a membrane respectively a skin stretched across a shell. The sound is produced by exciting the membrane causing it to vibrate. Drum kit shells usually consist of multiple plies of various woods or other shell materials. A batter head and an optional resonant head are held to the bearing edges on top and bottom of the shell by cast or pressed metal rims. The diameter of the shell determines the note of the drum, the depth influences articulation and resonance. The larger the

[1] The image is based on `http://commons.wikimedia.org/wiki/File:Drum_kit_illustration.png` from the freely usable media files database Wikimedia Commons and stands under the GNU license for free documentation. The author of the image is Clngre.

diameter of the shell, the lower the pitch of the drum will be. Kick drum, snare drum and tom-toms are membranophones.

2.1.1 Kick drum

The kick drum has the biggest diameter in the drum kit and thereby produces the lowest pitch. It is the only drum played with a pedal-operated beater, either with a felt, wood or plastic head. Sizes for kick drums range from 16 to 26 inch in diameter and 14 to 22 inch in depth. The standard size of the kick drum is 22x18 (22 inch diameter by 18 inch depth).

2.1.2 Snare drum

The snare drum has a unique feature in the drum kit, its eponymous snares. Strands of curled snares made of metal cords stretch across the resonant drumhead. When the batter head is struck, the snares vibrate with the resonant head and create the typical snare drum sound. Often the snare tension can be adjusted with a mechanism that includes a lever to completely throw off the snares, so that the drum only produces a tom-tom sound. A snare is generally played by hitting the centre of the batter head, but there are other techniques as well: A rimshot is a playing technique where the drum stick hits the head and rim simultaneously, creating an accented and loud note. Also used is a playing technique known as side stick, rim click or cross stick, where the stick hits against the rim of the snare while lying reversed on the drum head. The standard size of the snare drum is 14x6 (14 inch diameter by 6 inch depth).

2.1.3 Tom-toms

Tom-toms are normal drums of different diameter and depth. Smaller tom-toms are usually mounted in front of the drummer over the kick drum and are called hanging toms or rack toms. Bigger tom-toms have legs attached and can be put on the floor, that is why they are called floor toms. The primary use of tom-toms is to make drum fills. Fills are breaks from the normal beat that emphasize transitions between two separate musical parts. Further to that, tom-toms can also be used like any other drum to create a rhythm. Most toms range between 6 and 18 inch in diameter and depth.

2.2 Idiophones

Idiophones are instruments that produce sound with the entire body. Typically the body is made of a solid, non-stretchable and resonant material, such as that of cymbals. Cymbals are circular plates with a raised cup or bell in the centre made out of alloys such as brass or various bronzes. The outer-most area of the cymbal is called rim or edge and the area in between the edge and the centre is called bow. The raised centre is called the cup or dome and has a drilled hole in the middle to attach the cymbal to a cymbal holder. Cymbals vary in diameter and thickness which affects the sound in the following way: The larger a cymbal gets, the more volume it produces. The thinner a cymbal, the more responsive it will be. The larger a cymbal of the same thickness, the lower its pitch. The thicker

a cymbal of the same size, the higher its pitch. Also, the sound of larger and thicker cymbals sustains longer. To play a cymbal it may be struck, scraped, shaken, or stroked. All cymbals in a drum kit, namely crash, hi-hat and ride belong to idiophones.

2.2.1 Crash

The crash produces a loud and sharp tone. It is mainly used for occasional accents in a musical phrase, mostly on the first beat, though it can also be played at any time or even continuously in refrains. Crashes are usually thinner and smaller cymbals. Typical sizes range from 16 to 20 inches in diameter.

2.2.2 Hi-Hat

The hi-hat instrument is a combination of a stand and two cymbals. The cymbals are mounted opposite each other on the stand and by the use of a foot pedal the distance between both cymbals can be adjusted. The pedal allows to play the hi-hat in a variety of ways: It can be played closed, open or various levels in between. It is also possible to create a very short chick sound by rapidly closing the hi-hat with the foot. Usually the top cymbal is thinner to make the hi-hat more responsive for playing. The function of the hi-hat is to maintain a steady rhythmic pattern stream that connects various instruments in a beat, to keep the groove going. Over the past, sizes from 8 to 15 inches in diameter have been built. Today's standard size is 14 inches, with 13 inches as a less-common alternative.

2.2.3 Ride

The ride is usually a larger and thicker cymbal. It is used to play continuous patterns rather than to provide accents as with the crash cymbal. Thereby its function is similar to the hi-hat. An exception are designated crash/ride or more rarely ride/crash cymbals which indicate that they are designed to serve either function. The most common diameter for a ride cymbal is about 20 inches, but anything from 18 to 22 inches is possible.

Chapter 3

Drum Sound Detection Approaches

The work presented in this chapter concentrates on the transcription of unpitched percussion instruments. Most of the research to date within this field has focused on unpitched percussion instruments found in a standard rock/pop drum kit. It is worth to mention that generally labelling these instruments as unpitched is not absolutely accurate, because membranophones can be tuned to a desired pitch by loosening or tightening the drum heads. There are even membranophones that can change the pitch dynamically, for example timpani (also known as kettle drums), which can be tuned during a performance by the use of a pedal. Still, most membranophones and in particular the drums in a drum kit are labelled as unpitched, because it is assumed that the pitch does not change during a performance.

Today's drum kits support drummers to create uniquely and individually sounding setups. Drums and cymbals are available in a wide assortment providing great tonal diversity to each and everyone's favour. This may be good for differentiation through a unique sound, but from the viewpoint of transcription systems the downside of different sizes, materials, fabrication techniques and tuning possibilities is, that they result in considerable timbre[1] variations obtained within a given instrument. Nevertheless, some general characteristics for drum kit instruments can be noted. Membranophones have most of their spectral energy contained in the lower regions of the frequency spectrum, typically below 1000 Hz. On the other hand, idiophones spread out their spectral energy more evenly across the frequency spectrum and thus contain more high-frequency content.

The striking of a given drum or cymbal can be modelled as an impulse function with a broad range of frequencies occurring at the impact, resulting in all possible modes of vibration being excited simultaneously. Examples of time domain waveforms and corresponding spectrograms for three different drum kit instruments in figure 3.1 confirm this statement. Furthermore, spectral saliences can be obtained from the spectrograms. A kick drum, as pure representative of membranophones, is responsible for creating the bass in the drum kit and thereby contains a lot of low-frequency energy in the spectrogram. Compared to

[1] The term timbre refers to the quality of a sound which is given by its overtones and distinguishes it from other sounds of the same pitch and volume.

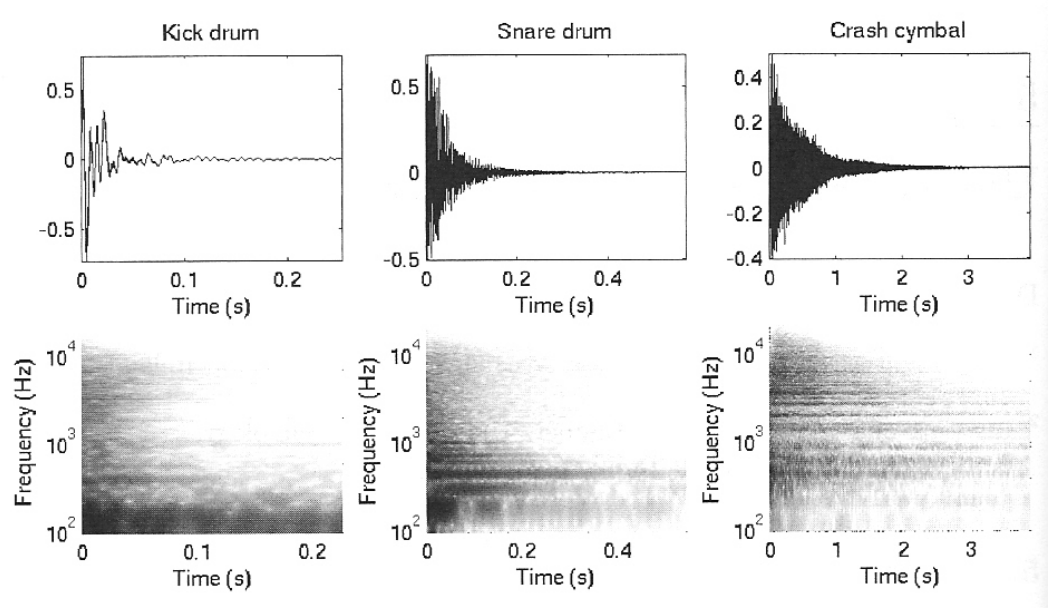


Figure 3.1: Example Waveforms and corresponding spectrograms of a kick drum, a snare drum and a crash cymbal. [9]

other drums in a drum kit the kick drum will have a lower spectral centroid[2]. A snare drum is also a membranophone, but the snares on the resonant head cause additional high-frequency energy to be present. A crash cymbal represents the instrument class of idiophones and produces energy over a wide frequency range. It is worth to mention that the example idiophone sound of the crash cymbal rings about ten times longer than the example membranophone sounds of kick and snare drum.

Percussion transcription systems may process either audio signals that contain percussive instruments only, or complex music signals where the presence of pitched instruments is allowed. In both cases it should be noted that input signals often contain heavily altered or no real live recordings of drums at all. The cause lies in prevalent production methods in the music business today where either samples of real drums, drum loops or synthetic drums are used to mimic real drums. A positive side effect of this development is that the waveforms for a given instrument will exhibit less variations between hits, making a correct detection easier for many transcription systems. However, the use of synthetic sounds can also have negative effects on transcription systems. Synthetic drum sounds try to mimic timbral characteristics of their real counterparts, but the spectral characteristics often differ considerably from them. This may lead to problems for transcription systems that are trained with real drums, because the right training data is crucial for the success of these transcription system.

Up-to-date approaches for percussion transcription can be divided into two categories: pattern recognition and separation-based approaches. The former approach segments the signal into events which are subsequently classified utilising pattern recognition methods.

[2]The spectral centroid is a measure to characterize the centre of a spectrum and is calculated as the weighted mean of the frequencies present in a signal, with their magnitudes as the weights.

The latter approach separates the mixture of drum sounds to single instrument streams which are then sought for onsets. Beside this classification, a distinction can be made between systems using a supervised or unsupervised approach. Systems using a supervised approach make use of templates or a training set to derive characteristic classifiers for a given instrument. Systems using an unsupervised approach use methods such as clustering of similarities followed by a recognition of the clusters.

In general, the problem of transcribing percussion instruments can be characterized by two simple questions:

- When did something happen in the musical piece?
- What was the event that took place?

3.1 Pattern Recognition Approaches

Transcription systems based on a pattern recognition approach typically operate by answering the two questions from above in this same order. Thus, they can be referred to as event-based systems. Usually these systems operate with roughly the following steps:

1. Segment the input signal into events by
   (a) locating potential sound event onsets in the input signal, or
   (b) generating a regular temporal grid over the signal.
2. Extract a set of features from each segment.
3. Classify the contents of each segment based on the extracted features.
4. Combine the segment time stamps with information about their content to yield the transcription.

3.1.1 Temporal Segmentation

There are two main approaches to segment an audio signal containing percussive instruments. The first approach segments an input signal into events marking the start time of each musical note. This approach, known as onset detection, is a common procedure and is used as a basis for many music information retrieval tasks. An overview of several proposed onset detection methods can be found in [17]. The second approach is to generate a regular temporal grid over the whole signal and subsequently use it for segmentation. This idea is driven by the assumption that almost all events lie on a temporal grid with a spacing that is determined by the fastest rhythmic pulse in the signal. However, in practice this approach can lead to a number of problems. For example, while playing an instrument, humans cannot keep in time like a machine. This causes small variances in the rhythmic pulse, which in turn causes problems with the assumption of fixed equidistant grid points.

Before the subsequent feature extraction, meaningful sections within the signal need to be segmented. The simplest approach is to extract a part of the signal between two consecutive located onset or grid points. But this procedure may lead to problems when

consecutive onsets are far apart or have no constant time difference. Therefore, it is a good idea to limit the minimum and maximum length of extracted segments and coevally assure that enough relevant information is present for feature extraction. In addition, a window function could be utilised on extracted segments, for example the common Hamming or Hanning window, though it is not usual, since it smooths out the attack part at the beginning of the segment, which usually contains valuable information. A half-Hanning window starting from a fixed value decaying to zero at the end of the segment would be more suitable, but is also omitted since sound events tend to decay to a small amplitude naturally.

3.1.2 Feature Extraction

Feature extraction is the process of picking features or regions of interest, used as indicators to identify or group the contents of selected segments. Within this context the prime objective is to calculate numerical values which specify the contents of a particular segment. Descriptors ideally provide significant evidence for a given instrument and exclude irrelevant information in the input signal. Moreover, descriptors should be robust to other simultaneously occurring sounds being present in the signal, but in practice this is difficult to achieve. A lot of research has been carried out to identify suitable descriptors for percussive instrument classification. Many of these descriptors can also be found in pitched musical instrument classification. Descriptors for percussive sound classification are either using properties of the frequency or the time-domain. Commonly used frequency descriptors are Mel Frequency Cepstral Coefficients (MFCCs), bandwise energy descriptors or simple spectral shape features.

MFCCs are well known in speech recognition and represent a signal spectrum in a compact way through the use of a number of coefficients. Usually 5 to 15 coefficients are calculated for short and partially overlapping frames in an analysed segment. These retained coefficients are not directly used as features, typically the mean and variance of each coefficient over the segment is calculated. Further to this, the coefficients first- and second-order temporal differences, respectively their means, are commonly used as features.

Bandwise energy descriptors are another commonly used set of features. The energy content of a signal is calculated for several frequency bands, where the number of bands and their spacing depends on a desired frequency resolution. The result of each bands energy in relation to the total signal energy is then used as descriptor. Typically values for 6 to 24 bands are calculated.

Beside the rough description of a spectrum through MFCCs and bandwise energy descriptors, a more simple definition of features is possible. Simple spectral shape features like spectral centroid, spectral spread, spectral skewness and spectral kurtosis can be used as descriptors. The normalized magnitude spectrum is defined by

$$\widetilde{X}(k) = \frac{|X(k)|}{\sum_{k \in K_+} |X(k)|}, \tag{3.1}$$

where $\widetilde{X}(k)$ denotes the discrete Fourier spectrum, k is a frequency index, and the set K_+

contains only non-negative frequency indices. The spectral centroid is then defined as

$$C_f = \sum_{k \in K_+} k\widetilde{X}(k). \tag{3.2}$$

The bandwidth of the spectrum is described by the spectral spread,

$$S_f^2 = \sum_{k \in K_+} (k - C_t)^2 \widetilde{X}(k). \tag{3.3}$$

Further, the spectral skewness describes the asymmetry of the frequency distribution around the spectral centroid with

$$\gamma_1 = \frac{\sum_{k \in K_+} (k - C_t)^3 \widetilde{X}(k)}{S_f^3}. \tag{3.4}$$

Finally, the spectral kurtosis describes the peakiness of the frequency distribution and is defined by

$$\gamma_2 = \frac{\sum_{k \in K_+} (k - C_t)^4 \widetilde{X}(k)}{S_f^4}. \tag{3.5}$$

Compared to spectral features, only a relative small number of time-domain features have been used in percussive sound classification. The reason for this is that the temporal progression of a sound is often modelled through differentials of spectral features extracted in short frames over the segment. Nevertheless, the two most common features in the time-domain are temporal centroid and zero crossing rate. The temporal centroid is a direct analogue to the spectral centroid and describes the temporal balancing point of the sound energy by

$$C_t = \frac{\sum_t tE(t)}{\sum_t E(t)}, \tag{3.6}$$

where $E(t)$ stands for the Root Mean Square (RMS) of the signal in a frame at time t, with the summation done over a fixed-length, starting at the beginning of the sound event. In correlation to the temporal centroid stands the zero crossing rate, which describes how often the signal changes its sign. Both features can be used to differ between idiophone or membranophone like sounds. Under the expectation that idiophones typically have a long ringing noise-like sound, whereas membranophones have a short and clearer pitched sound, the distinction between both based on temporal centroid and zero crossing rate becomes simple. The temporal centroid allows the discrimination of the length of a sound, the zero crossing rate the discrimination of the brightness or noise.

Most feature sets are typically selected empirically through trial and error, though efforts have been made to use automatic feature selection algorithms. It has been observed that in most cases a selection of features, gained through a feature selection method, yielded better results than the use of all available features. It should also be mentioned that a dimension reduction method, for example a principal component analysis, can lead to better results if applied to a set of extracted features prior to classification.

3.1.3 Segment Recognition

In the segment recognition phase the previously extracted features are processed to identify percussive sounds. There are at least two ways to do this. One way is to detect given percussive instruments separately, even if other instruments are present simultaneously, and the other is to detect combinations of percussive instruments directly. For instance, if an segment contains a kick drum and a hi-hat sound, the first approach will try to detect the kick drum and the hi-hat independently from each other, whereas the second approach goes beyond the detection of singular sounds with the detection of 'kick drum + hi-hat', thereby treating instrument combinations as separate entities.

A problem may arise when not only detecting individual percussive instruments, but also combinations of them. Given n different percussive instruments, the number of all possible combinations is 2^n. This relation indicates that the number of possible combinations exponentially increases as function of n. This can lead to a very large number of instrument combinations, making it difficult to take them all into account. However, the analysis of real signals shows that only a small subset of combinations is common. Figure 3.2 shows the relative occurrence frequencies of the ten most common instrument combinations for the RWC music database[3] containing 315 musical pieces. These combinations consider a class of five drum kit instruments and cover 95% of the drum sound events in the analysed data.

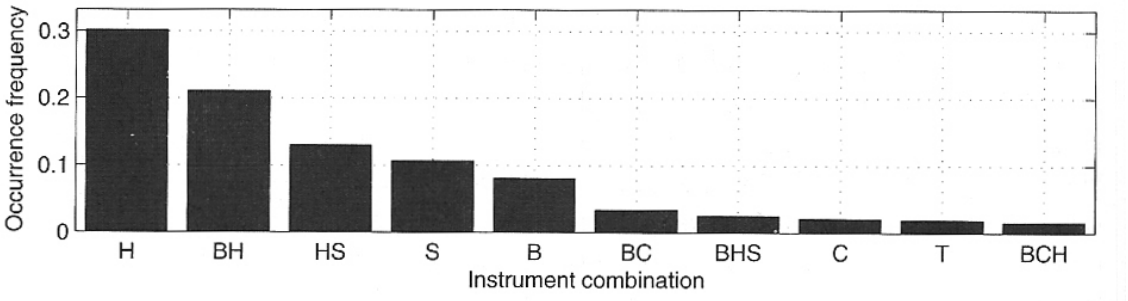


Figure 3.2: The ten most frequently occurring drum sound combinations in the RWC music database[3], in order of their relative occurrence frequency. These combinations cover 95% of all drum sound events present in the database. The abbreviations stand for the following instruments: H hi-hat, B bass drum, S snare drum, C cymbal, T tom-tom. [9]

Proposed transcriptions systems for Western popular music typically limit the number of possible percussive instruments n from two to eight. Some systems focus on transcribing only kick and snare drum, whereas others extend the instrument set with hi-hats, tom-toms, or various other percussion instruments.

Classification algorithms can be divided into three different categories:

- tree-based methods
- instance-based methods

[3] `http://staff.aist.go.jp/m.goto/RWC-MDB/`

- statistical-based methods

There are no extensive comparisons available which would strongly indicate the best classification method. A work by Herrera et al. [19] focusing on the classification of isolated drum sounds suggests that instance-based algorithms perform best and decision tree methods perform worst among the tested methods.

Tree-based methods

Tree based-methods use decision trees for classification. A decision tree is a decision support tool that uses a graph or model of decisions and their possible consequences to identify a strategy that points to a specific goal. Decision trees can be used as a predictive model to map observations of an item to a conclusion about the item's identity. In a decision tree leaves represent classifications and branches represent linkings of features that lead to those classifications. Applied to percussion transcription, decision tree classifiers operate by processing a sequence of preliminary defined questions on each segment. Such a series of question could include the question: 'Is the zero crossing rate in the signal below 500Hz?'. The results of those questions define a path through the decision tree that could be described by a large conditional expression in the same manner. Each question's answer defines the next question and may rule out some of the possible classification results.

Instance-based methods

Instance-based methods make use of stored and labelled training data to determine the content of a given segment. This approach compares each segment to the given training data to make assumptions about its content. In practical use, the weakness of traditional instance based methods like the k-nearest neighbours algorithm, is the memory required for storing training data and the computational load needed to compute distances to all the training samples. In order to overcome these drawbacks, Support Vector Machines (SVMs), well known in the area of machine learning, are used. SVMs help to minimize the training data by pre-processing it so only significant samples are considered for classification. Suppose a set of input data that can be allocated to either two classes and is represented as a set of vectors in an n-dimensional space. The basic function of a trained SVM is to construct a decision surface or separating hyperplane in that space, one which maximizes the margin between these two classes. In other words, given some data points each belonging to one of two classes, the goal is to decide which class a new data point will be in. The decision surface is parametrized by the sample patterns that have the smallest margin, called the support vectors. SVMs have successfully been used in a number of classification tasks, also in percussion transcription. Initially, SVMs were used as binary classifiers to indicate whether a segment contains or does not contain a specific percussive sound event. By gradually extending the SVMs from binary to multiple class decision systems Gillet and Richard were able to compare n binary SVMs and one 2^n class SVM [11]. Their results suggest that a multiclass SVM performs slightly better than several binary classifiers.

Statistical-based methods

A member of statistical-based methods are Gaussian Mixture Models (GMMs). In general, a mixture model is a model in which independent variables are fractions of a total. Each cluster is mathematically represented by a gaussian distribution. The entire data set is modelled by a mixture of these distributions. GMMs model the distribution of feature values in each class as a sum of gaussian distributions. GMMs have successfully been applied in pattern recognition because they can perform a 'soft' classification by approximating arbitrary distributions. An example for the use of GMMs in percussion transcription is the work of Paulus and Klapuri [22], who used a separate GMM for each of the 127 non-silent combinations of different drum instruments. GMMs are often used in combination with Hidden Markov Models (HMMs) to represent the feature distributions in their states. A HMM is a statistical model in which the system being modelled is assumed to fulfil the Markov property. The Markov property implies that a future state in a stochastic process depends only upon the present state and not on any past states. In other words, the present state contains all the information that could influence the future states. HMMs are derived from regular Markov models that consist of a finite set of states. The states are associated with a probability distribution and are directly visible to the observer. State changes are called transitions and the probabilities associated with various state-changes are called transition probabilities. In contrast to a regular Markov model the state of HMM is not directly visible, but the variables respectively the outcome influenced by the state are visible. That is why a HMM consists of two parallel processes: a hidden state process which cannot be directly observed, and an observation process holding the variables or features. The observed features are conditioned on the hidden states by using a GMM in each state and, based on the observations, the hidden state sequences can be derived.

Gillet and Richard used the combination of HMMs and GMMs for transcribing drum loops [11] and labelling signals containing Indian tablas [10]. In their work on Indian tablas they achieved a recognition rate of 94% on the test database with the use of GMMs to model different strokes, and HMMs to model event sequences in tabla recordings. A similar approach of utilising GMMs in conjunction with HMMs has been used in their work on transcribing drum loops containing Western drum sounds, where the results suggest that SVM classifier without any sequence modelling perform better than HMM-based approaches.

3.1.4 Instrument Model Adaptation

While isolated percussive sounds can be quite reliably identified, real-world recordings are much harder to analyse. The reason for this is that percussive instruments are usually accompanied by other melodic instruments. Different simultaneous sounds interfere and overlap with each other and create a mixture that is difficult to analyse. In addition, real percussive instruments have a broad range of sounds depending on how and where they are struck. Therefore, even in continuous musical pieces the percussive sounds can vary between occurrences. This traits make it difficult to construct general acoustic models that are applicable to any data and still discriminate reliably between different instruments.

A solution to this problem is to train models with data that is as similar as possible to the target mixture signals. This is difficult to achieve, because exact properties cannot

be known in advance or may vary within the material. To overcome this problem, model adaptation has been proposed. The idea behind this approach is to use general models and adapt them to the analysed material, instead of using inflexible predefined models for each and every target signal. Three percussion transcription systems that make use of this approach are presented below.

The first transcription system utilizing model adaptation was created by Zils et al. [33], using an analysis by synthesis approach. Initially, simple lowpass and bandpass-filtered impulses were used to create synthetic percussion sounds $z_i(n)$. These were used as approximations for kick and snare drum and served afterwards as a starting point for the adaptation process with the intention to generate more accurate models. The algorithm consisted of the following steps:

1. Calculate the correlation function between a synthetic sound event $z_i(n)$ and the polyphonic input signal $y(n)$

$$r_i = \sum_{n=0}^{N_i-1} z_i(n)y(n+\tau), \tag{3.7}$$

   where N_i is the number of samples in the sound i, and $r_i(\tau)$ is defined for $\tau \in [0, N_y - N_i]$, where N_y is the number of samples in $y(n)$.

2. Locate occurrences of the sound $z_i(n)$ in $y(n)$ by picking peaks in $r_i(\tau)$ and by retaining only most reliable peaks.

3. Update the sound $z_i(n)$ with

$$z_i(n) \leftarrow \frac{1}{2}\left[z_i(n) + \frac{1}{U}\sum_{j=1}^{U} y(\tau_j + n)\right], \qquad n = 0, \ldots, N_i - 1, \tag{3.8}$$

   where U is the number of reliable peaks detected and τ_j contains their locations.

4. Repeat steps 1 to 3 until convergence.

This procedure was applied separately for kick and snare drum. If both drums occurred simultaneously, priority was given to the kick drum, thereby limiting the system to a monophonic transcription of either those two drums. The system was evaluated with 100 examples of various music genres. If percussion instruments were louder or as loud as the other instruments in the mixture, the success rate was over 75%. In the opposite case, where percussion instruments were quieter, the success rate dropped to 40%.

Another system using a model adaptation approach was created by Sandvold et al. [23]. It differs from the previously described system, because the adaptation is done on a higher abstraction level. Instead of using time domain or time-frequency domain templates, the models were based on extracted features. After segmentation and feature extraction the algorithm operated through the use of the following steps:

1. Classify the extracted n segments with a tree-based classifier and general instrument models.

2. According to the reliability of the classification results, select $m < n$ of the most reliable results.

3. Create localized models of the selected m segments.

4. Classify the contents of all the n segments again, using the localized models.

The general instrument models were derived from training data by the use of 115 spectral and temporal features and by selecting the most suitable of them with a correlation-based feature selection algorithm. The obtained reduced feature set had on average less than 25 features for all classes. In step two, the classification results of step one were ranked manually according to their reliability, making the overall system operate semi-automatically. During the construction of the localized models in step three all 115 features were reconsidered again and the best reduced feature-subset, determined like before, was taken. After this step, only nine features on average were sufficient for each class. As classification algorithm for the localized models a simple instance-based algorithm was used, more precisely the k-nearest neighbours algorithm with $k = 1$. The system was evaluated with manually annotated ground truth of seventeen polyphonic audio recordings with a length of 20 seconds. The results suggest that the use of localized models reduce the number of required features while coincidentally improving the recognition results from 72% to 92%.

The third system utilizing instrument model adaptation is described later in section 5.4.3 as part of the implementation.

3.2 Separation-Based Approaches

Having explored event-based approaches, it is clear that the mixture of sounds is complicating the task of correctly transcribing percussive events. A completely different way to approach the problem of percussion transcription and resolve this issue, is to use separation-based techniques. Separation-based techniques ease the transcription of signals containing mixed percussion instruments by separating the instruments into distinct streams. Despite their advantages, these techniques have also certain drawbacks to overcome. In many separation-based methods the streams have to be identified after separation and the occurrence of low intensity sounds is difficult to detect. However, transcription systems applying separation-based methods have proven well.

Different separation-based methods have been suggested to solve the problem of blind source separation. Blind source separation describes the problem of separating a set of signals from a set of mixed signals, with very little, or no information about the source signals or the mixing process. It relies on the assumption that the individual source signals in a mix are mutually and statistically independent. The most frequently used source-separation technique is Independent Component Analysis (ICA). The basic ICA requires as many sensors (microphones) as there are sources for separation to occur. The problem is that usually music is distributed in stereo, thus only two channels are available at best. Moreover, percussion instruments tend to be mixed equally in volume for both channels, which minimizes the meaningful information to only one channel. As a result, most work using separation-based approaches focuses on single-channel separation.

Some of the various methods used to separate sources in single-channel recordings are Independent Subspace Analysis (ISA), None-Negative Sparse Coding (NNSC), and Non-Negative Matrix Factorisation (NMF). All these methods have found their use in percussion transcription and are shortly described in the following passage. It has to be noted that separation, in the context of transcription, means the separation of frequency and amplitude characteristics associated with each source in order to identify and describe them.

All above mentioned techniques rely on the following assumptions: A mixture signal and its spectrogram matrix X of size $(K \times T)$, where K is the number of frequency bins and T is the number of time frames, can be described by the superposition of J source spectrograms Y_j of the same size. Further, it is assumed that each of the spectrograms Y_j can be defined by the outer product of an invariant frequency basis function b_j of length K and a corresponding invariant amplitude basis function (or time-varying gain) g_j of length T, describing the gain of the frequency basis function over time. This yields:

$$X = \sum_{j=1}^{J} Y_j = \sum_{j=1}^{J} b_j g_j^T. \tag{3.9}$$

The decomposition from above is usually applied to the magnitude spectrogram, rather than to the power spectrogram. The magnitude spectrogram consists of the absolute values, whereas the power spectrogram consists of the square values of the discrete Fourier transform for consecutive frames. The mentioned techniques differ in how the decomposition of spectrogram X into its amplitude and basis functions is performed. The basic ISA method proposed by Casey performs a Principal Component Analysis (PCA) on the spectrogram, keeping only a small number of decorrelated frequency lines, and then performing ICA on the inferred components. In effect, ISA performs ICA on a low-dimensional representation of the original spectrogram. NNSC attempts to balance modelling accuracy with the sparseness of the recovered sources while imposing non-negativity. NMF attempts to reconstruct the data using non-negative basis functions and a Poisson or Gaussian noise model.

In practice, the use of invariant basis functions is coupled with the condition that pitch changes are not allowed in the individual spectrograms Y_j. This condition is usually fulfilled for most drum sounds, since most percussive instruments are unpitched and do not change their pitch between occurrences. That makes the decomposition well suited for transcribing percussive tracks in polyphonic music.

As a decomposition example, figure 3.3 displays the magnitude spectrogram of a drum loop containing kick and snare drum hits. The corresponding amplitude and frequency basis functions, retained with NMF, are displayed in figure 3.4. It can be seen that the amplitude basis functions reliably indicate each point in time where the instruments were hit. Further, by comparing amplitude basis functions of both instruments, it is evident that the amplitude envelopes have been well separated, though some influence of the kick drum remains visible in the snare basis function. The depicted frequency basis functions indicate the overall spectral characteristics of the sources, with the kick drum having a high amount of low-frequency energy, opposed to the snare having its spectral energy spread out over a wide frequency range. This example vividly demonstrates the application of separation techniques and shows the promising results that can be achieved in the field of

percussive transcription.

However, the above mentioned techniques show certain deficits from the standpoint of percussion transcription, regardless of how decomposition is achieved. One problem is the undetermined source order after segmentation, which requires the identification of each separated source. Another problem is the estimation of the optimal number of basis functions. A smaller number of basis functions leads to more recognizable features, while low-energy sources require an increased number of basis functions to be recognized. Further, ISA recovers basis functions which may have negative elements. This stands in contradiction to the assumption that the overall magnitude or power spectrogram is the additive result of independent spectrograms which are non-negative by definition. This may result in errors during transcription, which NNSC and NMF avoid by restraining non-negativity of the sources.

Different techniques have been proposed to solve the problems described above, all using prior knowledge about the percussion instruments to be transcribed. One way to apply prior knowledge is to add sub-band processing before utilizing source separation. By knowing the characteristics of percussion instruments, it is easy to define filters which emphasize certain sub-bands and their features while coevally reducing unwanted and misleading noise.

This approach was demonstrated in a transcription system described in [6] that was designed to detect kick drum, snare drum, and hi-hats. Two basis functions were used to retrieve the kick and snare drum from the lowpass-filtered signal using ISA, and similarly two basis functions retained the hi-hat and snare from the highpass-filtered signal. The problem of source identification was solved through the assumption that the spectrum of the kick drum had a lower spectral centroid than the snare drum, and that hi-hats occurred more often than the snare drum.

The system was evaluated on a set of 15 drum loops which were taken from different drum sample CDs with the intention to cover a broad range of sounds within each type of instrument. Diverse conditions have been generated through the selection of different tempos, metres, and varying relative amplitudes ranging from 0 dB to -24dB between the drums. The total test set contained 133 drum events and was evaluated using $c = (t - u - i)/t$ where c is the percentage correct, t is the total number of drums, u is the number of undetected drums, and i is the number of incorrectly detected drums. Using this measure a success rate of 90% was achieved.

Beyond this example, more effective approaches to incorporate prior knowledge have been proposed. The interested reader is referred to [9].

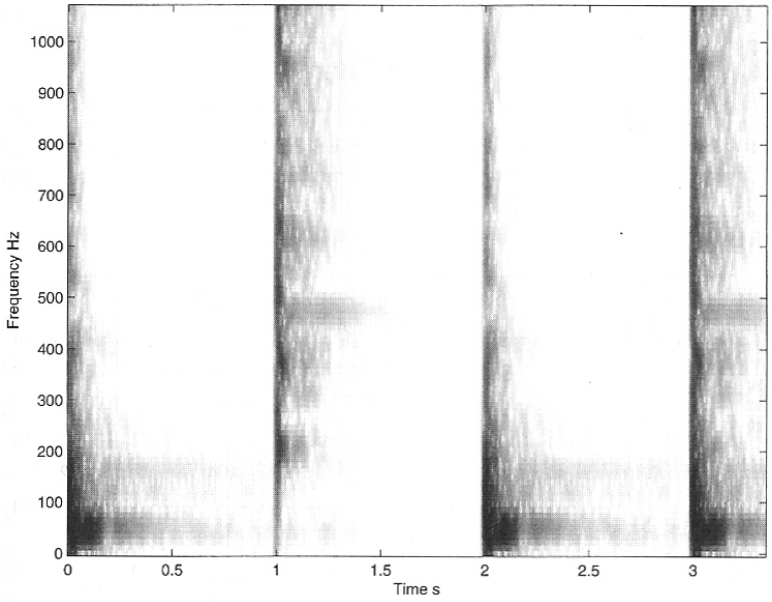


Figure 3.3: Magnitude spectrogram of a drum loop containing snare and kick drum. [9]

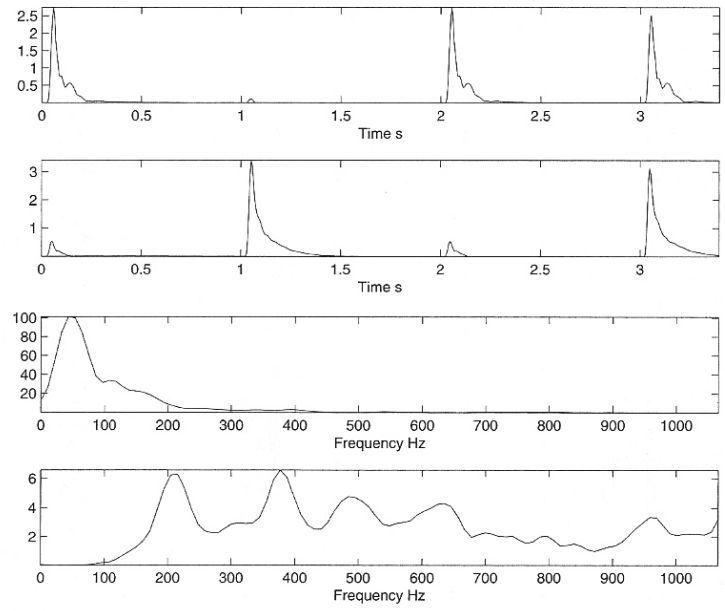


Figure 3.4: Basis functions recovered from the spectrogram in figure 3.3. The first two plots represent basis amplitude functions of kick and snare drum, followed by the corresponding frequency basis functions. [9]

Chapter 4

Related Work

A problem with research on percussion transcription systems is the lack of comparative results. This is due to varying problem definitions, and to the use of disparate test sets and evaluation measures. These circumstances make an objective evaluation of the effectiveness and performance of many systems difficult. However, considerable effort has been spent in the past to solve the problem of percussion transcription. A summary of important transcription systems to date is presented in table 4.1. As can be seen, a variety of approaches have been utilised to tackle the problem of percussion transcription. A term used in the given table, which has not been mentioned yet, is musicological modelling. The term summarises approaches that apply musical knowledge on low-level recognition results in order to gain more accurate transcriptions. Attempts in this context usually incorporate compositional rules by modelling statistical dependencies of event sequences. It is reasonable to believe that such approaches improve the performance considerably. Still, many transcription methods focus only on low-level recognition and do not consider the possible advantages of higher abstraction levels.

Only a few comparative evaluations exist that compare different transcription systems on the same test data. The most widely known evaluation was performed in the framework of MIREX[1] 2005 at the Audio Drum Detection contest[2]. The results of this contest were taken as first starting point for research on possible implementations, whereby most of the attention was given to the three systems that performed best. The systems that ranked second and third are covered below, whereas the winning system is described at length in chapter 5.

The algorithm which placed second was submitted by Tanghe et al. [25]. The algorithm can be categorized as an event-based approach and consists of three main parts: onset detection, feature extraction, and feature classification. The onset calculation splits the input signal in different frequency bands and calculates amplitude changes in order to decide with a heuristic peak grouping detector whether an onset occurred or not. The feature extraction stage then calculates simple spectral shape and time-domain features for each onset. Finally, the classification stage applies a classification model computed by a SVM that decides to which drum types the extracted feature vectors correspond. The

[1] Music Information Retrieval Evaluation eXchange is an annual benchmarking contest which runs in conjunction with the International Conference on Music Information Retrieval (ISMIR).

[2] The list of participants and the official results can be found at `http://www.music-ir.org/evaluation/mirex-results/audio-drum/index.html`

Author	Classes	Drums only	Mono/ Poly	Method	Main algorithm
Dittmar & Uhle [3]	5		p	S+A	Non-negative ICA
FitzGerald et al. [6]	3	X	p	S	Sub-band ISA
FitzGerald et al. [5]	3	X	p	S+A	PSA
FitzGerald et al. [7]	3		p	S	PSA
FitzGerald et al. [8]	7	X	p	S+A	Adaption & PSA
Gillet & Richard [11]	8	X	p	E+M	HMM, SVM
Goto & Muraoka [14]	9	X	p	E	Template matching
Goto [13]	2		p	E	Frequency histograms
Gouyon et al. [16]	2	X	m	E	Feature Extraction
Herrera et al. [19]	9	X	m	E	Various
Herrera et al. [18]	33	X	m	E	Various
Paulus & Klapuri [22]	7	X	p	E+M	GMMs & N-grams
Paulus & Virtanen [21]	3	X	p	S	NMF
Sandvold et al. [23]	3		p	E+A	Localized models
Sillanp et al. [24]	7		p	E	SVMs
Van Steelant et al. [27]	2		p	S	Sparse coding
Virtanen [28]	2		p	S	Sparse coding
Yoshii et al. [30]	2		p	E+A	Template matching
Zils et al. [33]	2		m	E+A	Template matching

Table 4.1: List of percussion transcription systems. The *Classes* column specifies the number of covered percussion instruments by the system. *Method* described the overall approach, whereby E stands for event-based systems, S for separation based systems, M for systems including musicological modelling, and A for systems using an adaptive approach. *Drums only* indicates that a system operates on signals containing drums only. *Mono/Poly* states whether the system can detect one or more simultaneous sounds. [9]

implemented algorithm is available for download on the Musical Audio-Mining (MAMI) project website[3] and includes a set of console applications, an overall audio-to-MIDI application, and a C/C++ library.

The system ranked third was submitted by Dittmar [1]. The signal processing chain of the algorithm can also be divided into three main stages. The first stage calculates onset times and collects the corresponding onset spectra. Then higher order statistical computations follow in order to estimate frequency and amplitude bases of the involved drum instruments, thus providing a decomposition into source components. Finally, a refinement stage that validates and enhances the intermediate results found so far delivers the classification and detection results. A implementation of the algorithm is encapsulated in the Transcription Toolbox [2] from the Fraunhofer IDMT institute, which could be tested in the course of research. A screenshot from the Graphical User Interface (GUI) is shown in figure 4.1.

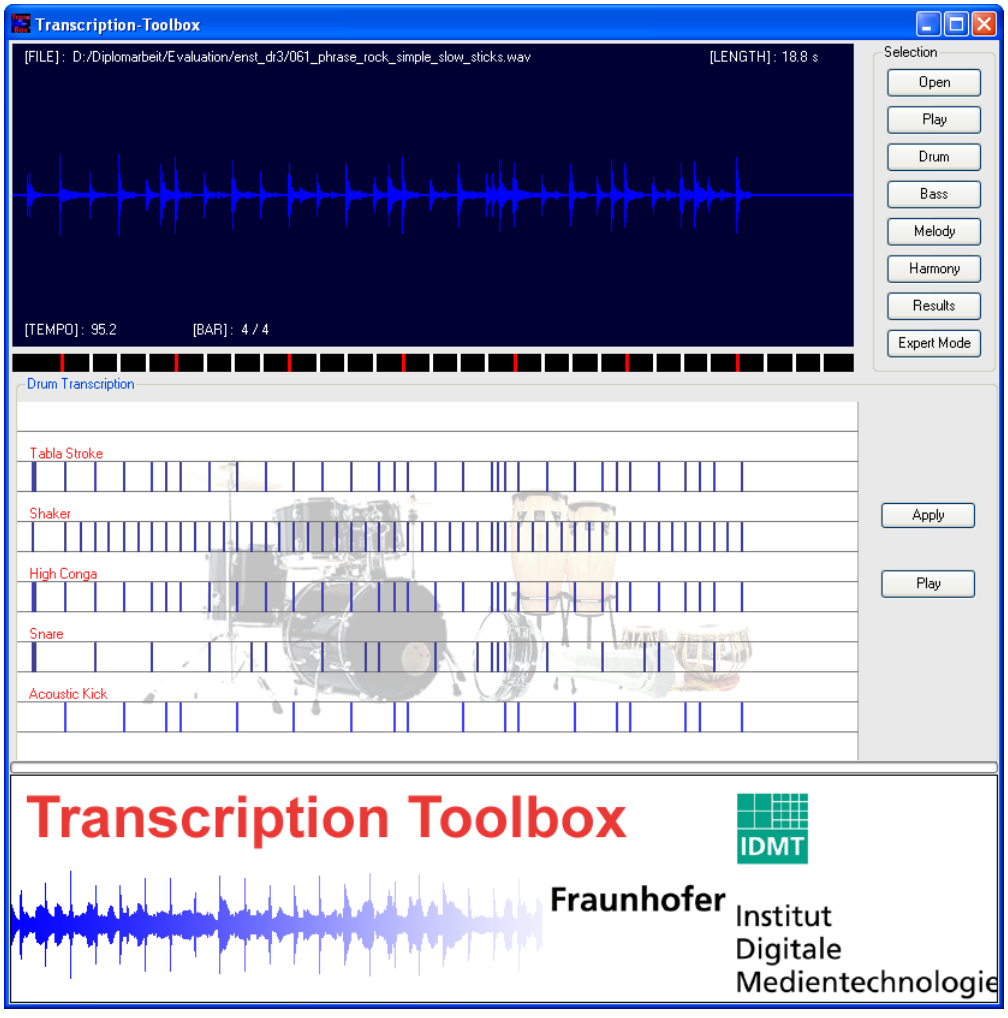


Figure 4.1: Fraunhofer toolbox for automatic transcription of polyphonic music.

[3] http://www.ipem.ugent.be/MAMI/

Chapter 5

Implementation

The following described implementation of a percussion transcription system is based on a system proposed by Yoshii et al. [32]. The system was first published in 2004 [30], [31] and was later submitted under the name 'AdaMast' [29] to MIREX 2005, where it won the Audio Drum Detection task. The overall description of the system is consistent throughout the mentioned papers, though minor differences exist. These differences most likely represent improvements over time, that is why the implementation focuses on the last published paper mentioned first.

Before describing the transcription system in detail, its goals plus a brief description of fundamental work is presented. Afterwards, the original algorithm is fully described and where necessary comments concerning the implementation are added.

5.1 Goals

The goal of the implemented percussion transcription system is to detect onset times of the three most commonly occurring instruments in a drum kit: kick drum, snare drum, and hi-hat. These three instruments usually form the rhythmic foundation of a musical piece, thus their transcription allows the extraction of meaningful and comprehensive percussive phrases for a given song. The system should be capable to analyse polyphonic audio signals as given input, for example recordings of commercial CDs. Further, it should not matter if drum sounds originate from real drums or generated samples. In order to simplify the complexity of the transcription system only basic playing techniques are considered. That means the system will be limited to detect sounds produced by a stick or beater hitting the batter head or cymbal directly, and neglects different playing tools (e.g. brushes) or special performance-techniques (e.g. head-muting).

5.2 Fundamental Work

The proposed system of Yoshii et al. utilizes the ideas of two preceding works in the field of MIR. The first work of influence was published by Goto and Muraoka [15], who created a system to recognize drum sounds in a signal containing synthetic drum performances, produced by a MIDI tone generator. Fixed-time-length power spectrograms of nine different drum kit instruments were prepared to function as templates. These templates were then

compared to the power spectrogram of the input signal under the assumption that the input signal is a polyphonic sound mixture of the prepared templates. To decide whether an instrument occurred or not, they proposed a distance measure that is robust to the spectral overlapping of simultaneous occurring drum sounds. The method achieved a high recognition accuracy, but the drawback is that the power spectrogram of each occurring instrument in the analysed signal must be known beforehand. Further, the method does not consider spectral overlapping of harmonic sounds produced by melodic instruments. For the sake of convenience, this approach is from no on referred to as Goto's distance measure.

The second work having an impact on the proposed system of Yoshii et al. discusses the resynthesis of drum sounds from CD recordings and was published by Zils et al. [33]. The system uses a template adaption method to recognize kick and snare drum sounds from polyphonic audio signals and has already been described in detail in 3.1.4, but for the sake of brevity can be summarized as follows: The method operates in the time domain and calculates the correlation between the waveform of predefined instrument templates and the input signal. If the calculated correlation for a given template in the input signal is high, it is assumed that the analysed instrument occurred and the corresponding template is updated with data from the input signal. This procedure is repeated until the templates converge. Although the evaluation results of this method were promising, the system is not fully capable of analysing overlapping drum sounds in the presence of other musical instruments.

5.3 System Overview

The general idea behind the designed transcription system of Yoshii et al. is to utilise a pattern recognition or otherwise called event-based approach. Beyond that, the intention is to apply a supervised approach in the detection of drum sounds, through the usage of predefined instrument templates. Considering this course of action and the desired goals two main problems arise:

- The acoustic features of the analysed drum sounds vary among musical pieces and are usually not known beforehand, which results in problems defining generally applicable templates for transcription.
- Percussive sounds are distorted by other simultaneous occurring musical instruments, thus it is difficult to correctly decide whether a given drum sound occurred or not.

To address these problems special methods are integrated in the transcription system. First, to deal with the problem of inhomogeneous drum sounds among analysed signals, a template adaptation method is introduced. Second, to overcome the problem of distorted percussive instrument sounds, a robust matching method is utilised.

Figure 5.1 shows an overview of the general system approach and design. Abstractly, the system can be seen as black box with a separate input and output layer. The data of the input layer is processed by the recognition system which produces the transcription results. More precisely, the output of onset times for a given instrument is the result of its predefined template, which is provided as input together with the audio signal to

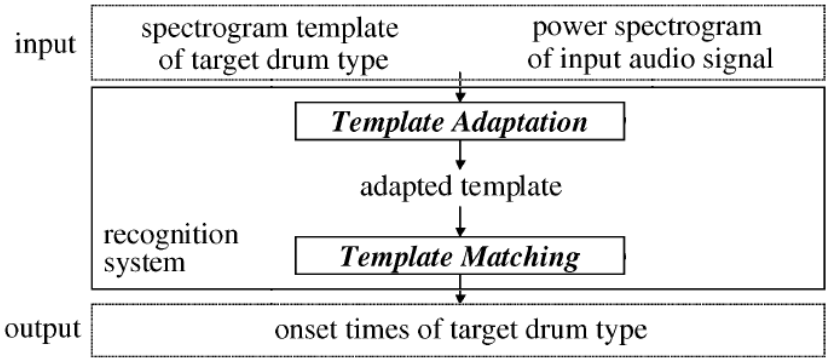


Figure 5.1: Overview of the percussion transcription system. [32]

be analysed. It can be seen that the two main problems mentioned before are solved in separate functional entities in the recognition system.

Template Adaptation The goal of this phase is to obtain a spectrogram template that is adapted to its corresponding drum sound in the analysed audio signal. Before the adaptation, spectrogram templates for each drum are defined; one for bass drum, snare drum, and hi-hat. These so-called seed-templates are then adapted by extending Zil's method to the frequency domain.

Template Matching In this phase the actual onset times of drum sounds in the musical piece are detected. This is done by comparing the adapted templates and the input signal with the help of Goto's distance measure, which takes the mixture of simultaneous occurring sounds into account.

Before describing the two stages from above in more detail, two general comments can be given. First, the overall system operates in the frequency domain. Second, it is worth to mention that two different distance measures are used in the *Template Adaptation* and *Template Matching* stage. In the adaptation stage it is desired to detect only pure drum sounds with little or no overlapping of other sounds, because these sounds reflect the instrument characteristics best and tend to result in good adapted templates without disruptive fragments. Further, it is not necessary to detect all onset times of an instrument in this stage, thus an Euclidean distance measure with focus on close spectral proximity is utilised. In the matching stage, on the other hand, it is necessary to exhaustively detect all onset times even if drum sounds are overlapped with other sounds, hence Goto's distance measure is utilised.

5.4 Template Adaptation

The timbre variations of drum sounds between musical pieces do not allow to create universally applicable templates. Therefore, a template adaptation is performed. As already explained in section 3.1.4, the idea behind template adaptation is to derive an appropriate instrument template for each audio signal from a predefined instrument template that describes only general features.

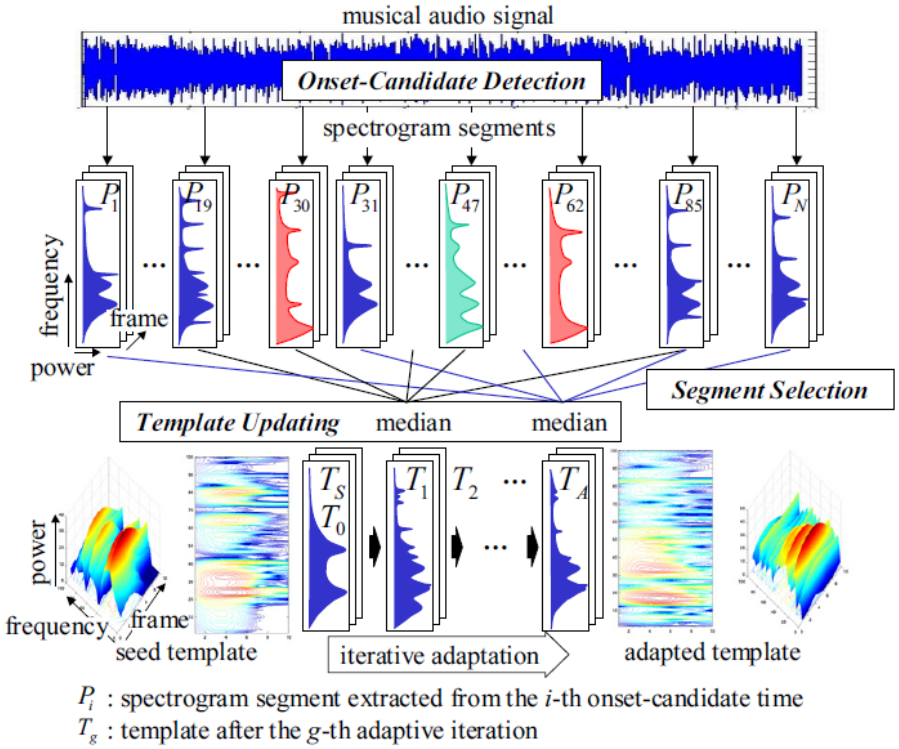


Figure 5.2: Overview of the template-adaptation method. [29]

The template adaptation method of Yoshii et al. uses a single initial so-called seed-template for each drum kit instrument; one for the kick drum, snare drum, and the hi-hat. Each seed-template is defined in the frequency domain as a fixed-time-length power spectrogram and serves as starting basis for the adaptation process, during which an iterative procedure adjusts the seed-template until it matches the desired drum sound of a given audio signal.

A general overview of the template adaptation method is shown in figure 5.2. The overall method can be divided into three functional blocks. First of all, the *Onset Candidate Detection* stage roughly detects onset candidates for a given musical piece and extracts corresponding spectrogram excerpts. The spectrogram segments are created by extracting fixed-time-length segments at the beginning of each onset candidate within the power spectrogram of the input signal. The extracted spectrogram segments and the seed template are then used in an iterative adaptation procedure, which involves the successive execution of the *Segment Selection* and *Template Updating* stage. The *Segment Selection* stage evaluates the reliability that each extracted spectrogram segment includes the drum sound spectrogram of the analysed template. Segments with high reliability are selected and subsequently used to update the seed template. Finally, the *Template Updating* stage creates an updated template by calculating the median power of each frame and frequency bin among the selected segments. The adapted template created by this procedure is then used as starting point in the next adaptive iteration.

5.4.1 Onset Candidate Detection

To reduce the computing time of the template matching process, an onset detection is performed. That makes it possible to extract a spectrum excerpt that starts not from every frame but every onset time. The onset detection aims to find the start times of all musical events within an audio signal. It is not possible to consider only specific instruments during onset detection, thus the detected onset times originate from all sound sources that are present in the input signal. The later processing is based on the results of the onset detection, because the template matching is only performed on segments that have been extracted according to the detected onset times. Therefore, it is important to minimize the detection failure of drum sound onsets. To prevent a lack of actual onsets in the first place, a high recall rate is preferred even if that leads to many false alarms.

The onset detection approach utilised by Yoshii et al. uses a peak-picking method to detect onset candidate times. Most onset detection algorithms use this approach, where the information in the time-domain is used to find local peaks in the slope of a smoothed amplitude envelope. This method is particularly well suited to music with drums. However, to minimize the effort of the implementation an already developed system called BeatRoot [4] was used as onset detector. BeatRoot is a well known beat tracking and visualisation system in the field of MIR and utilises an even more advanced onset detection algorithm. The algorithm is more sensitive than the peak-picking method in the time-domain, because it finds peaks in the spectral flux. Spectral flux is a measure of how quickly the power spectrum of a signal is changing, calculated by comparing the power spectrum for one frame against the power spectrum from the previous frame. This approach allows to detect even very soft hits of cymbals or drums.

In general, the time-frequency representation of the input signal is calculated with a Short Time Fourier Transform (STFT) using a Hamming window, but it is worth to mention that the recommended STFT-settings for the onset detection and the transcription process differ. The initial STFT processing for the onset detection is performed with a window length of 2048 samples (46 ms at a sampling rate of 44100 Hz) and a hop size of 441 samples (10 ms, or 78.5% overlap). The STFT processing for the transcription is performed with a window length of 4096 samples (93 ms at a sampling rate of 44100 Hz) and a hop size of 441 samples (10 ms, or 89.2% overlap). The reason for the different window lengths is that several tests showed that the onset detection of BeatRoot works better with a window length of 2048 samples, rather than 4096 samples, where some onsets were missed. Further, it should be noted that BeatRoot offers the option to normalize the spectrogram data either by the current frame energy or by an exponential average of the frame energy. By default BeatRoot utilizes the latter setting, which causes an undesired blurring of spectral features and further results in a decreased recognition accuracy of the transcription system. Thus, normalisation was turned off.

Seed Template Construction

A seed template T_S is a power spectrogram which is prepared for each drum type to be recognized. The time-length of seed template T_S is fixed. T_S can be defined as a time-frequency matrix denoted by $T_S(t, f)(1 \leq t \leq 10[frames], 1 \leq f \leq 2048[bins])$.

To cover a wide range of timbre variations with seed template T_S, multiple sounds of a single instrument are aggregated. The necessary drum sounds for this process were

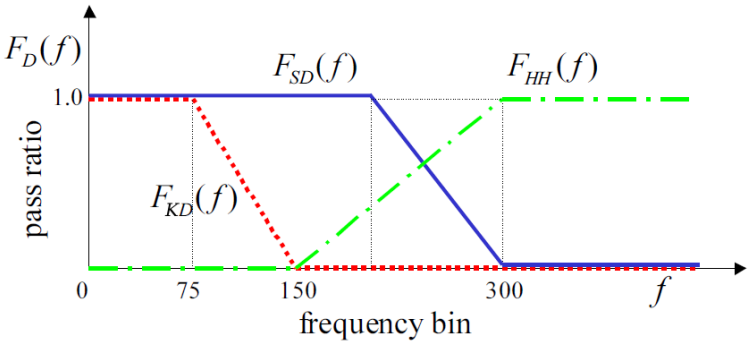


Figure 5.3: Filter functions for kick drum, snare drum and hi-hat. [29]

taken from a drum sample CD, which contains various real kick drum, snare drum, and hi-hat sounds. From this sound pool a set of six different drum samples is selected for each instrument, whereby only sounds produced by a normal stroke and only closed hi-hat sounds are considered. By analysing the selected samples with an onset detector, an onset time for each sample is detected. Starting from this onset time, a power spectrogram with the size of the seed template T_S is extracted from the samples. Therefore, multiple power spectrograms for an instrument are obtained, each of which is denoted as $S_i (i = 1, \ldots, N_S)$, where N_S stands for the number of extracted power spectrograms (number of selected samples). The final aggregation is done by selecting the maximum power at each frame and each frequency bin among the power spectrograms $\{S_i | i = 1, \ldots, N_S\}$

$$T_S(t, f) = \max_{1 \leq i \leq N_S} S_i(t, f). \tag{5.1}$$

To eliminate interfering artefacts and emphasize characteristic frequency parts in the audio signal for each instrument, a filter function $F_D(f)\{D|KD, SD, HH\}$, shown in figure 5.3, is introduced. By utilising a lowpass function for the kick (KD) and snare drum (SD) together with a highpass function for the hi-hat (HH), it is assumed that only typical frequency characteristics are examined in the later processing. In the iterative adaptation algorithm, let T_g denote a template being adapted after the gth iteration. Because T_S is the first template, T_0 is set to T_S. To obtain only characteristic frequencies of template T_g, the filter function F_D is applied. Thus, the filtered power spectrogram is the result of

$$\acute{T}_g(t, f) = F_D(f) T_g(t, f). \tag{5.2}$$

Spectrogram Segment Extraction

The ith spectrogram segment $P_i (i = 1, \ldots, N_O)$ is a power spectrogram extracted from each onset time o_i [ms] in the input signal, where N_O is the number of detected onsets. The size of each extracted segment is the same as seed template T_S, thus it can also be defined as time-frequency matrix. Depending on the currently analysed drum sound, each extracted segment is filtered by filter function $F_D(f)$. The filtered power spectrogram is

then denoted as

$$\acute{P}_i(t,f) = F_D(f)P_i(t,f). \tag{5.3}$$

5.4.2 Segment Selection

First, the reliability E_i that spectrogram segment P_i includes the spectral features of a specific drum sound, is calculated. Then, a descending order of spectrogram segments, with regard to the calculated reliabilities $\{E_i|i = 1,\ldots,N_O\}$, is prepared, to gather a selection of spectrogram segments with the highest correlation. The appropriate selection size is defined as a fixed ratio of the number of selected segments to the total number of spectrogram segments (number of detected onsets N_O). The proposed ratio is empirically set to 0.1, that means the number of selected segments is $0.1 \times N_O$.

The reliability E_i is defined as reciprocal of the distance D_i between template T_g and spectrogram segment P_i

$$E_i = \frac{1}{D_i}. \tag{5.4}$$

The distance measure D_i should satisfy that a high correlation between spectrogram segment P_i and drum sound spectrogram T_g results in a low distance. In other words, when the reliability that spectrogram segment P_i includes the drum sound spectrogram T_g is large, D_i should become small. The reciprocal of D_i is then used to map a small distance value D_i into large reliability value E_i.

The calculation of the distance measure for the hi-hat differs from the one for the kick drum and snare drum, thus both methods are described separately.

Recognition of Kick Drum and Snare Drum Sounds

The spectrogram templates of kick drum and snare drum can be characterized by salient spectral peaks. The positions of these peaks vary according to the actual timbre of the instrument. That is the reason why in the first adaptive iteration, where the seed template T_0 has never been adapted, the spectral peak positions of T_0 vary considerably to the actual drum sound of the given input signal. Additionally, characteristic spectral peaks in the spectrogram may be overlapped by other sounds present in the musical piece. In both cases typical spectral distance measures (e.g., Euclidean distance) cannot be applied. For different spectral peak positions between T_0 and P_i, such distance measures would make the distance D_i large even if spectrogram segment P_i includes the analysed drum sound. further, in the case of mistaken spectral peaks, the distance D_i would become small resulting in a wrong selection of spectrogram segments. Thus, in conclusion timbral differences or overlapping sounds would cause inappropriate results with a typical distance measure in the first iteration.

To overcome these problems, spectral smoothing is applied to the spectrogram of seed template T_0 and each segment P_i. The intention is to create a rougher representation of the spectrogram in order to eliminate unwanted details. Thereby, it is possible to concentrate on the similarity of instrument sounds, even if an Euclidean distance measure is utilised. The precise peak positions between seed template T_0 and each spectrogram segment P_i become negligible, which prevents the unwanted increase of distance D_i even if P_i includes the analysed drum sound. The spectral smoothing is achieved by averaging a

number of consecutive frequency bin values over subsequent frames, thereby the resolution of the spectrogram is changed. Figure 5.4 and 5.5 depict this procedure, the proposed time resolution is 2 [frames] and the frequency resolution is 5 [bins].

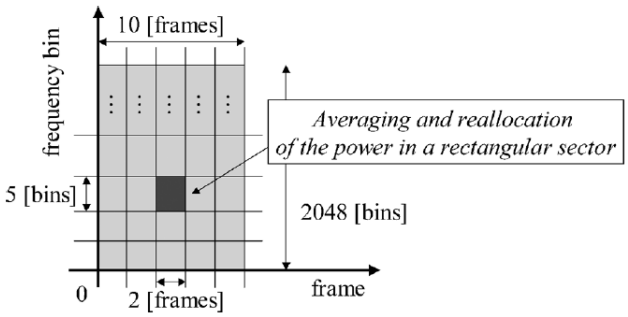


Figure 5.4: Spectral smoothing method for bass and snare drum to obtain a rougher spectrogram without details. This inhibits the undesirable increase of the distance between the seed template and spectrogram segments when using an Euclidean distance measure in the first iteration. [32]

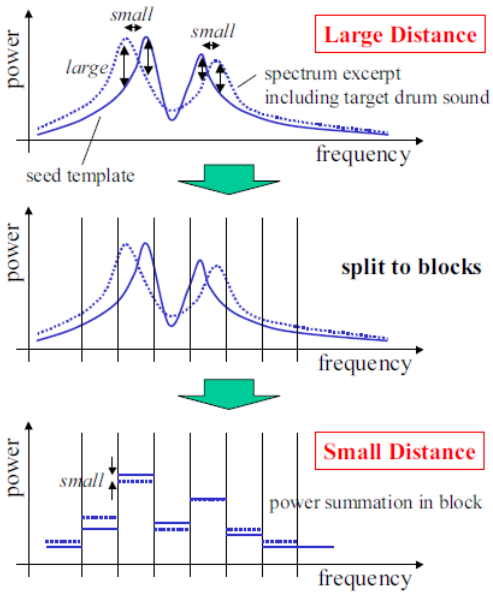


Figure 5.5: Different representation of the spectral smoothing approach. A lower quantization allows the comparison of similarities. [30]

Let $\widehat{T}_0$ and $\widehat{P}_i$ denote the smoothed seed template and a smoothed spectrogram segment. According to the previously selected resolution settings the time range is defined by $2t^{'} - 1 \leq t \leq 2t^{'}$ for $(1 \leq t^{'} \leq 5)$, and the frequency range is defined by $5f^{'} - 4 \leq f \leq 5f^{'}$

for $(1 \leq f' \leq 409)$. $\widehat{T}_0(t,f)$ is then calculated by

$$\widehat{T}_0(t,f) = \frac{1}{10} \sum_{t=2t'-1}^{2t'} \sum_{f=5f'-4}^{5f'} \acute{T}_0(t,f). \tag{5.5}$$

$\widehat{P}_i$ is calculated in the same manner. This operation describes the averaging and reallocation of the spectral power within a defined region. The time-frequency domain is split into rectangular cells of 2 [frames] and 5 [bins] in size. Afterwards, the power of each cell is averaged and reallocated to all bins inside the cell.

The spectral distance $D_i^{(0)}$ in the first iteration is defined as

$$D_i^{(0)} = \sqrt{\sum_{t=1}^{10} \sum_{f=1}^{2048} \left(\widehat{T}_0(t,f) - \widehat{P}_i(t,f)\right)^2} \tag{5.6}$$

Once the first iteration is finished, it is no longer necessary to apply spectral smoothing. The reason is that the spectral peak positions of the template $T_g (g \geq 1)$ are now adapted to the drum sound in the audio signal. From this time on it is desired to gather precisely calculated distance measures, in order to select the most appropriate segments for the template adaptation process. Therefore, the spectral distance $D_i^{(g)} (g \geq 1)$, which takes the differences of spectral peak positions into account, is defined as

$$D_i^{(g)} = \sqrt{\sum_{t=1}^{10} \sum_{f=1}^{2048} \left(\acute{T}_0(t,f) - \acute{P}_i(t,f)\right)^2} \qquad (g \geq 1) \tag{5.7}$$

Recognition of Hi-Hat Sounds

The hi-hat belongs to the instrument class of idiophones, which typically have a broad distribution of spectral energy across the frequency spectrum. For this reason, the power spectrogram of the hi-hat does not show any salient spectral peaks. As a consequence, the system of Yoshii et al. focuses on the spectral envelope of hi-hat sounds, rather than on precise spectral structures denoted by spectral peaks. To extract the spectral envelope and simultaneously ignore large variations of local spectral features, spectral smoothing is applied. The overall procedure to calculate the distance measure for the hi-hat is similar to the one for kick and snare drum. The only major difference is that the spectral smoothing is applied throughout all iterations, because the spectral envelope is used as recognition pattern. Therefore, the spectral distance D_i in any adaptive iteration is always calculated after spectral smoothing for template T_g and spectrogram segment P_i. To retain an adequate representation of the frequency characteristics for the hi-hat the spectral smoothing is done with a time resolution of 2 [frames] and a frequency resolution of 20 [bins]. In the end, the spectral distance D_i in any iteration between template T_g and spectrogram segment P_i is defined as

$$D_i^{(g)} = \sqrt{\sum_{t=1}^{10} \sum_{f=1}^{2048} \left(\hat{T}_0(t,f) - \hat{P}_i(t,f)\right)^2} \tag{5.8}$$

5.4.3 Template Updating

The result of the adaptation process is an adapted template that reflects the timbre of the drum sound contained in the input signal. The final adapted template is the result of several updated templates that have been created in intermediate stages of the adaptation process. Each update template is constructed by calculating the median power at each frame and frequency bin among all spectrogram segments that have been selected previously in the selection stage. The template updating for kick and snare drum differs from the hi-hat, thus both methods are described separately.

Recognition of Kick and Snare Drum

Let N_S denote the number of selected spectrogram segments, which is $0.1 \times N_O$. The updated template $\acute{T}_{g+1}$ is then calculated by

$$\acute{T}_{g+1}(t,f) = \underset{1 \leq i \leq N_S}{\text{median}} \acute{P}^{(i)}(t,f), \tag{5.9}$$

where $\acute{P}^{(i)}(i = 1, \ldots, N_S)$ stands for each selected spectrogram segment. The median function is utilised because it achieves two intended objectives at the same time: First, it points out similar spectral components among the selected spectrogram segments. Second, diverse spectral components among the selected spectrogram segments are suppressed. This can be explained in the following way: The spectral structure of a drum sound, defined through salient spectral peaks, is expected to be included in all selected spectrogram segments. At the same time spectral components caused by other sound sources appear at different positions among the selected spectrogram segments. Therefore, if the local power at a specific frame and frequency matches among many spectrogram segments, it is most likely that the local power represents the pure drum sound. This property can be well explained with the median function, because its main nature is to ignore outliers while underlining similarities. That makes sure that during the updating process the pure drum sound is highlighted by selecting the median of the local power while deviant spectral components become outliers and are not selected. Thus, the median leaves only common features across the selected segments. As a result, it is possible to obtain a template that has a spectrogram that is very close to the solo drum sound, even if other sound sources are included in the extracted segments. The whole procedure is depicted in figure 5.6.

Recognition of Hi-Hat

The updating process for the hi-hat is similar to the process for kick and snare drum. The only difference is that the updated template is based on smoothed spectrogram segments. The smoothing process makes sure that the stable median power can be obtained despite the fact that the local power for hi-hat sounds varies among the spectrogram segments. In other words, the shape of the spectral envelope is stable for smoothed segments, thus a stable median can be calculated. The updated and smoothed template $\hat{T}_{g+1}$ is obtained by

$$\hat{T}_{g+1}(t,f) = \underset{1 \leq i \leq N_S}{\text{median}} \hat{P}^{(i)}(t,f). \tag{5.10}$$

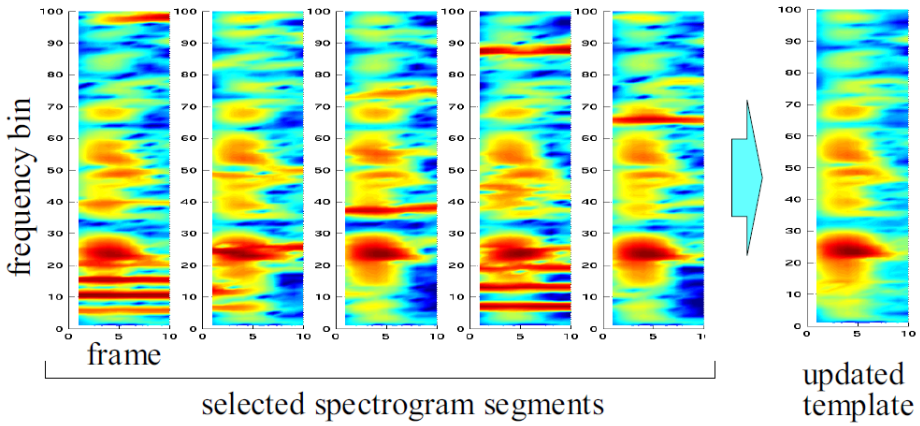


Figure 5.6: Template updating by selecting the median power at each frame and each frequency bin among selected spectrogram segments. [29]

5.5 Template Matching

This method detects all temporal locations of a drum sound by utilizing the spectral information of its adapted template. The described procedure of Yoshii et al. is able to detect onset times in a polyphonic audio signal, even if other musical instrument sounds overlap the analysed drum sound. To find the actual onsets, the spectrogram segments of all onset candidates are examined whether a particular drum sound is included or not. This determination is difficult because other sounds often overlap the analysed drum sound. Due to that, a general distance measure cannot be applied, because it would calculate an improper distance. To be more specific, the distance between an adapted template and a spectrogram segment including the analysed sound would become large, because of other sounds being present in a segment. To put it in another way, the overlapping of other sound sources makes the distance large, even if the drum sound spectrogram is included in a spectrogram segment.

To overcome this problem, the distance measure approach from Goto et al. [15] is utilised. Goto's distance measure determines whether the spectral envelope of a spectrogram segment is likely to contain the spectral envelope of the adapted template. Thus it can calculate a distance even if other simultaneous sounds interfere with the drum sound.

Figure 5.7 shows an overview of the overall template matching approach. The procedure can be divided into three blocks which are passed subsequently. First, the Weight-Function Preparation stage applies a filter function to the adapted template. The filter function assures that only important frequency regions and characteristic frequency bins are taken into consideration. Next, the Power-Adjustment stage calculates the power difference between the adapted template and each spectrogram segment by combining local power differences at characteristic frequency bins to a single power difference value. Afterwards, a threshold is calculated among all power differences and it is considered that only segments below this threshold contain the target drum sound spectrogram. These spectrogram segments are then selected for further processing, where the power of each segment is adjusted to compensate the power difference. Afterwards, the Distance-Calculation stage

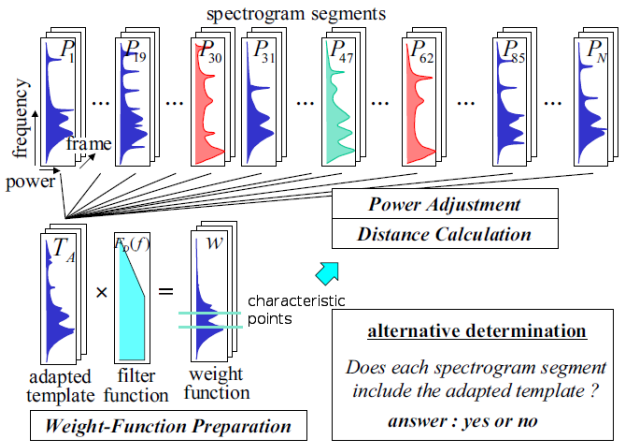


Figure 5.7: Overview of the template matching method. [30]

determines the distance between the adapted template and each adjusted spectrogram segment. Finally, the last step is to calculate a threshold among all distances which allows conclude whether a drum sound spectrogram is included in a spectrogram segment or not.

The template matching process is exemplified in more detail for the kick and snare drum. For the recognition of the hi-hat the processing is based on smoothed spectrogram segments, thus the described template matching process is compliant with the hi-hat by replacing [´] with [ˆ] in each expression.

5.5.1 Weight Function Preparation

A weight function is generated to represent characteristic frequency components within the adapted template. The weight function w is defined as

$$w(t, f) = \acute{T}_A(t, f), \tag{5.11}$$

where $\acute{T}_A$ specifies the adapted and filtered template. The weight function is utilised in the subsequent processing for the extraction of significant features.

It is assumed that the spectral peak positions of each frame give a good representation of the adapted template and are significant enough to serve as meaningful features. Thus, for each frame the position of an occurring peak is selected as characteristic frequency bin. Let $f_{t,k}(k = 1, \ldots, K_D)$ be the characteristic frequency bins in the adapted template, where $K_D(D = KD, SD, HH)$ is the number of characteristic frequency bins and $K_{KD} = 10$, $K_{SD} = 10$, $K_{HH} = 100$. $f_{t,k}$ is then determined by fulfilling the following conditions:

$$w(t, f_{t,k}) \geq w(t, f_{t,k} - 1) \tag{5.12}$$

$$w(t, f_{t,k}) \geq w(t, f_{t,k} + 1) \tag{5.13}$$

$$w(t, f_{t,k}) > \lambda_w \times \max_f w(t, f) \tag{5.14}$$

where λ_w is a constant set to 0.5, to satisfy that a characteristic bin is larger than 50% of the maximum magnitude within a frame. Finally, the k largest frequency bins among $w(t, f(t, k))$ are selected.

5.5.2 Power Adjustment

To correctly determine the onset times for a drum sound with Goto's distance measure, it is necessary to adjust the power of each segment to the adapted template. This ensures that Goto's distance measure can be applied even if the power of a drum sound included in a spectrogram segment is lower than that of the adapted template. Without the power adjustment Goto's distance measure would estimate an inappropriate distance between a spectrogram segment and an adapted template, because it could not determine whether the adapted template is included in the spectrogram segment or not. The power of each spectrogram segment is adjusted by a beforehand determined power distance under the assumption that the drum sound spectrogram segment is included in that spectrogram segment. Given the power differences of all spectrogram segments it can be assumed which spectrogram segments contain the drum sound spectrogram. A relatively low power difference supports the conclusion that a spectrogram segment includes the template spectrogram, whereas a relatively large power difference would indicate that a spectrogram segment does not include the template spectrogram.

To determine the power difference between each spectrogram segment and template, the local power difference at the previously selected characteristic frequency bins is calculated.

Power Difference Calculation

The local power difference $\eta_i(t, f_{t,k})$ at frame t and characteristic frequency bin $f_{t,k}$ is denoted with

$$\eta_i(t, f_{t,k}) = \acute{P}_i(t, f_{t,k}) - \acute{T}_A(t, f_{t,k}). \tag{5.15}$$

The overall power difference $\delta_i(t)$ at frame t is then determined as the first quartile of $\eta_i(t, f_{t,k})$

$$\delta_i(t) = \underset{k}{\text{first-quartile}}\, \eta_i(t, f_{t,k}) \tag{5.16}$$

$$K_i(t) = \underset{k}{\text{arg-first-quartile}}\, \eta_i(t, f_{t,k}) \tag{5.17}$$

where the variable $K_i(t)$ represents the value of k when $\eta_i(t, f_{t,k})$ is the first quartile. If the number of frames where the condition $\delta_i(t) \leq \Psi_\delta(t)$ is fulfilled exceeds a threshold R_δ, it is assumed that the adapted template is not included in the spectrogram segment. R_δ is set to 5 [frames] and $\Psi_\delta(t)$ is a threshold whose determination is described in section 5.5.4. The reason for selecting the first quartile among the power differences $\{\eta_i(t, f_{t,k})|k = 1, \ldots, K_D\}$ is that the first quartile takes possible outliers into consideration. Outliers may occur when the power at a characteristic frequency bin is affected by frequency components of other instruments. Picking out the first quartile prevents the selection of misleading power difference values, because accidental large power differences are ignored by the function and replaced by a more relevant power difference. Figure 5.8 depicts the power difference calculation process.

Some research during the implementation of this algorithm component showed that quartiles are simple in concept, but can be complicated in execution. The concept of quartiles is that some data is arranged in ascending order and divided into four roughly equal parts. The lower quartile is then the top ranked part containing the lowest data

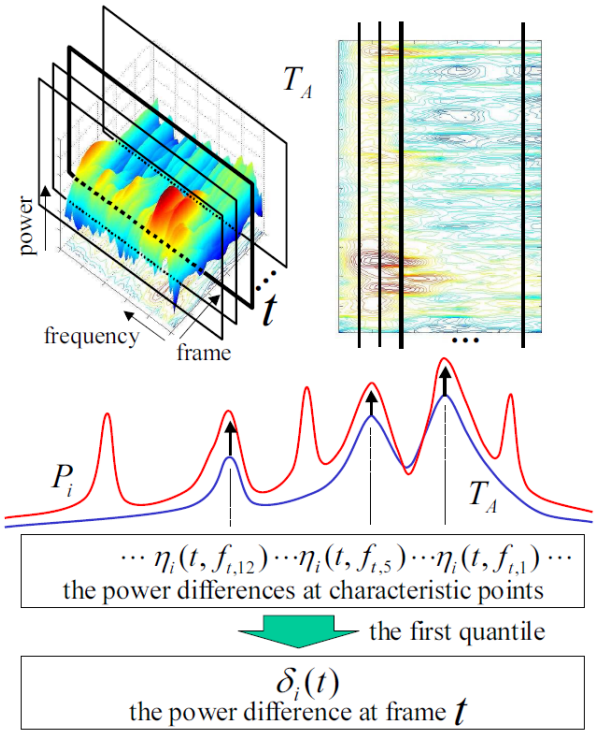


Figure 5.8: Overview of the power difference calculation process. [30]

values. What makes the handling of quartiles confusing is that the term quartile can have two meanings. The first definition refers to the subset of data values contained in each part. The second definition specifies a quartile as the cut-off value between two parts, which are often denoted as $\{Q_1, Q_2, Q_3\}$ (Q_1 is the lower quartile, Q_2 the median, Q_3 the upper quartile). Supposing this definition, statisticians do not agree on whether the quartile values should be points from the data set itself, or whether they can fall between two data points (as the median can when there are an even number of data points). Furthermore, statisticians are discordant how to deal with the median for the calculation of quartiles. Thus, numerous different quartile calculation methods exist[1]. A list of five methods which are commonly known is shown in table 5.1.

For the implementation the method of Mendenhall and Sincich (M&S) was chosen, because it is the only method where the upper and lower quartile values are always a single data point. This requirement can be derived from equation 5.17, where the first quartile should be a single data point to successfully map k. The formula to calculate the lower and upper quartile with the method of Mendenhall and Sincich is given in table 5.1.

The following example illustrates the calculation of the first quartile with the M&S method: Given a sample of five data points $S_5 = (5, 6, 7, 8, 9)$, the first quartile is calculated with $\frac{n+1}{4} = 1.5$. Due to the fact that the result is half an odd integer we round up and select the element on the second position to be the first quartile, hence $Q_1 = 6$.

[1] Journal of Statistics Education Volume 14, Number 3 (2006), `http://www.amstat.org/publications/jse/v14n3/langford.html` gives a description of over a dozen methods

Method	1^{st} quartile n odd	1^{st} quartile n even	3^{rd} quartile n odd	3^{rd} quartile n even
Tukey	$\frac{n+3}{4}$	$\frac{n+2}{4}$	$\frac{3n+1}{4}$	$\frac{3n+2}{4}$
Moore and McCabe	$\frac{n+1}{4}$	$\frac{n+2}{4}$	$\frac{3n+3}{4}$	$\frac{3n+2}{4}$
Mendenhall and Sincich	$\frac{n+1}{4}$*	$\frac{n+1}{4}$*	$\frac{3n+3}{4}$**	$\frac{3n+3}{4}$**
Minitab	$\frac{n+1}{4}$	$\frac{n+1}{4}$	$\frac{3n+4}{4}$	$\frac{3n+3}{4}$
Freund and Perles	$\frac{n+3}{4}$	$\frac{n+3}{4}$	$\frac{3n+1}{4}$	$\frac{3n+1}{4}$

*if the result is half an odd integer, round up
**if the result is half an odd integer, round down

Table 5.1: List of common methods for computing the positions of the first and third quartiles from a sample size n.[2]

Power Adjustment

The total power difference Δ_i is calculated by considering all local power differences $\delta_i(t)$ that satisfy $\delta_i(t) \geq \Psi_\delta(t)$, weighted by weight function w

$$\Delta_i = \frac{\sum_{\{t|\delta_i(t)\geq\Psi_{\delta(t)}\}} \delta_i(t) w(t, f_{t,K_i(t)})}{\sum_{\{t|\delta_i(t)\geq\Psi_{\delta(t)}\}} w(t, f_{t,K_i(t)})}. \tag{5.18}$$

If $\delta_i \leq \Theta_\Delta$ is satisfied, it can by assumed that the template is not included in the spectrogram segment, where Θ_Δ denotes a threshold that is automatically determined in section 5.5.4. Finally, the adjusted spectrogram segment denoted by $\tilde{P}_i$ is obtained with

$$\tilde{P}_i(t, f) = \acute{P}_i(t, f) - \Delta_i. \tag{5.19}$$

5.5.3 Distance Calculation

To calculate the distance between the adapted template $\acute{T}_A$ and an adjusted spectrogram segment $\tilde{P}_i$, an adapted version of Goto's distance measure is utilised. The adapted version calculates the distance with respect to the presence of multiple overlapping sounds. Goto's distance analyses the power difference at each time t and frequency bin f between the adapted template and the spectrogram segment. If the magnitude of $\tilde{P}_i(t, f)$ is larger than $\acute{T}_A(t, f)$, Goto's distance measure considers that $\tilde{P}_i(t, f)$ is a mixture of spectral components created by the drum sound and other sound sources. In that case, if the local

[2]Weisstein, Eric W. "Quartile." From MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/Quartile.html`

power distance is above a certain threshold, Goto's distance measure assumes that the currently analysed frequency component is contained in the spectrogram segment. This ensures that Goto's distance measure does not make the distance large when the spectral components of the drum sound are overlapped by those of other sounds. When a spectrogram segment includes the drum sound spectrogram, the power difference between $T_A(t,f)$ and $\tilde{P}_i(t,f)$ is expected to be relatively low, since the local power distance at frame t and frequency bin f is minimised. Therefore, the local distance measure denoted as $\gamma_i(t,f)$ is defined as

$$\gamma_i(t,f) = \begin{cases} 0, & \left(\tilde{P}_i(t,f) - \acute{T}_A(t,f) \geq \Psi_D\right) \\ 1, & \text{otherwise} \end{cases} \tag{5.20}$$

where $\Psi_D(D = KD, SD, HH)$ is a negative threshold with the purpose to ignore small variations of local spectral components. The proposed threshold values for each instrument are: $\Psi_{KD} = -12.5$ [dB], $\Psi_{SD} = -12.5$ [dB], $\Psi_{HH} = -5$ [dB]. Given the above definition, $\gamma_i(t,f)$ becomes zero when $\tilde{P}_i(t,f)$ is larger than $\acute{T}_A(t,f)$. In conclusion, the local distance is a binary classifier that indicates whether the frequency component at a particular position in the spectrogram segment has an equivalent in the adapted template.

To calculate a total distance value for a segment its local distance values $\gamma_i(t,f)$ are combined and transferred to the time-frequency domain by utilising the weight function w. Thus, the local distance for each segment, denoted by Γ_i, is obtained by

$$\Gamma_i = \sum_{t=1}^{10} \sum_{f=1}^{2048} w(t,f)\gamma_i(t,f). \tag{5.21}$$

To finally decide whether a drum sound occurred at a specific segment P_i, the total distance Γ_i is compared with a threshold Θ_Γ that is automatically determined in section 5.5.4. If it is satisfied that $\Gamma_i < \Theta_\Gamma$, it is assumed that the analysed drum sound occurred at the corresponding onset time of segment i.

5.5.4 Automatic Threshold Selection

To achieve good recognition results all previously mentioned thresholds are individually calculated for each input signal using Otsu's threshold selection method. In total 12 thresholds $(\{\Psi_\delta(t)|t = 1,\ldots,10\}, \Theta_\Delta, \Theta_\Gamma)$ are optimized. Otsu's method determines each threshold $(\Psi_\delta(t), \Theta_\Delta, \Theta_\Gamma)$ by separating a set of values $(\{\delta_i(t)|i = 1,\ldots,N_O\}, \{\delta_i|i = 1,\ldots,N_O\}, \{\Gamma_i|i = 1,\ldots,N_O\})$ into two classes of varying structural properties.

Finally, to balance the recall rate with the precision rate, the thresholds Θ_Δ and Θ_Γ are adjusted in the way that

$$\Theta_\Delta \rightarrow \lambda_\Delta \times \Theta_\Delta, \quad \Theta_\Gamma \rightarrow \lambda_\Gamma \times \Theta_\Gamma, \tag{5.22}$$

where λ_Δ and λ_Γ denote empirically determined balancing factors.

The threshold generation is an essential part of the algorithm, since it determines in a large proportion the success of the transcription system. Unfortunately, this part of the algorithm is described very briefly in [32], thus the implementation is based on personal assumptions. These assumptions include the use of a histogram for the calculation of each threshold and the supposition that the matching procedure is based on the refinement of

picked onsets. It should be mentioned that it was decided to abandon the use of balancing factors (see equation 5.22) in the implementation, because it seemed curious to manipulate calculated thresholds in hindsight with a constant.

Otsu's Method

Otsu's threshold selection method is well known in the field of computer vision or more specifically in image processing. Probably the most popular application of Otsu's method in this context is the reduction of grey-level images to binary images. Otsu's algorithm assumes that the pixels in an image can be divided into two classes (e.g. foreground and background) and calculates the optimum threshold separating those two classes. The grey-level image is then converted to a binary image by turning all pixels below the threshold to white, and all pixels above the threshold to black. The algorithm operates directly on the grey-level histogram which shows the distribution of the tonal intensity levels. The algorithm further assumes that the histogram is bimodal which means that it contains two classes. The calculation of the optimal threshold is then based on the simple idea of finding the threshold that minimizes the within-class variance, which is defined as the weighted sum of variances of each cluster

$$\sigma^2_{\text{Within}}(t) = q_1(t)\sigma^2_1(t) + q_2(t)\sigma^2_2(t) \tag{5.23}$$

where the class probabilities are estimated as

$$q_1(t) = \sum_{i=1}^{t} P(i) \tag{5.24}$$

$$q_2(t) = \sum_{i=t+1}^{N} P(i) \tag{5.25}$$

and the class means are given by

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)} \tag{5.26}$$

$$\mu_2(t) = \sum_{i=t+1}^{N} \frac{iP(i)}{q_2(t)} \tag{5.27}$$

where [1,N] is the range of intensity levels (number of histogram bins). Finally, the individual class variances are:

$$\sigma^2_1(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \tag{5.28}$$

$$\sigma^2_2(t) = \sum_{i=t+1}^{N} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)} \tag{5.29}$$

Given these equations, the threshold selection method is fully described. The optimal threshold is calculated by running through the full range of t values and picking the one

value that minimizes $\sigma^2_{\text{Within}}(t)$. However, computing the within-class variance for each of the two classes for each possible threshold requires a lot of computation. To ease the calculation the relationship between the within-class and the between-class variance can be exploited. The total variance σ^2 is given by

$$\sigma^2 = \sigma^2_{\text{Within}}(t) + \sigma^2_{\text{Between}}(t). \tag{5.30}$$

Since the total variance is always constant and independent of t, the effect of changing the threshold is merely to move the contributions from one term to the other. Thus, minimizing the within-class variance is the same as maximizing the between-class variance. If the within-class variance is subtracted from the total variance the between-class variance is

$$\begin{aligned} \sigma^2_{\text{Between}}(t) &= \sigma^2 - \sigma^2_{\text{Within}}(t) \\ &= q_1(t)[\mu_1(t) - \mu]^2 + q_2(t)[\mu_2(t) - \mu]^2 \end{aligned} \tag{5.31}$$

where μ is the combined mean. The between-class variance is simply the sum of weighted squared distances between the means of each class and the overall mean. After substituting $\mu = q_1(t)\mu_1(t) + q_2(t)\mu_2(t)$ and simplifying the between-class variance is

$$\begin{aligned} \sigma^2_{\text{Between}}(t) &= q_1 q_2(t)[\mu_1(t) - \mu_2(t)]^2 \\ &= q_1[1 - q_1(t)][\mu_1(t) - \mu_2(t)]^2. \end{aligned} \tag{5.32}$$

where $[1-q_1(t)]$ is a substitute for q_2. Now, the between-class variance depends only on the difference between the means of the two clusters and the probability of the first cluster. The benefit from calculating the between-class variance is that the quantities of $\sigma^2_{\text{Between}}(t)$ can be computed recursively while running through the range of t values, because the computations are not independent while changing from one threshold to another. q_1 and the cluster means $\mu_1(t), \mu_2(t)$ can be updated as elements move from one cluster to the other while t increases. By using simple recurrence relations the optimal threshold which maximizes the between-class variance can be found while successively testing each threshold:

$$\begin{aligned} q_1(t+1) &= q_1(t) + P(t+1) && \text{with initial value } q_1(1) = P(1) \\ \mu_1(t+1) &= \frac{q_1(t)\mu_1(t) + (t+1)P(t+1)}{q_1(t+1)} && \text{with initial value } \mu_1 = 0 \\ \mu_2(t+1) &= \frac{\mu - q_1(t+1)\mu_1(t+1)}{1 - q_1(t+1)} \end{aligned}$$

5.6 Graphical User Interface (GUI)

The system described above has been implemented in Java, whereby BeatRoot (Release 0.5.6[3]) has been used as starting basis in order to reduce the programming effort. It seemed beneficial to combine the beat tracking capabilities of BeatRoot with drum transcription

[3]available from `http://www.elec.qmul.ac.uk/people/simond/beatroot/index.html`

capabilities since both applications are closely related. This approach also saved the burden of implementing low-level audio file operations from scratch and allowed the use of an existing GUI. The original GUI includes the rendering of audio data as a waveform and spectrogram and was enhanced to display the transcribed onset times for the kick drum, snare drum, and the hi-hat (Figure 5.9). The detected percussion events can be saved as a list of time and event pairs into a simple text file. Furthermore, the application was enhanced to allow the concurrent or separate playback of the audio file, the transcribed drum sounds, and detected beat times. For evaluation purposes the application supports the processing of large file amounts through command line batch processing.

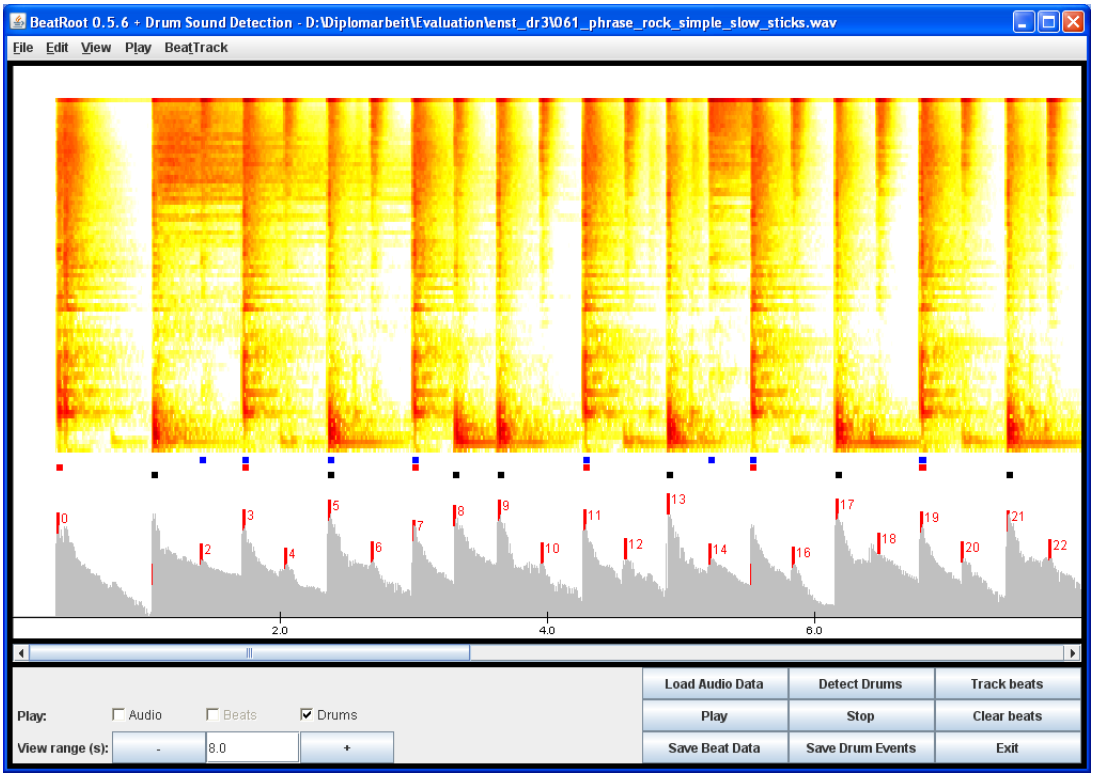


Figure 5.9: A screenshot of the implemented drum transcription system showing an 8-second excerpt from a drum track, with the spectrogram (top), the transcribed drum sound onsets marked as coloured boxes (black: kick, red: snare, blue: hi-hat), amplitude envelope marked with detected onsets (short vertical lines), and the control panel (bottom).

Chapter 6

Evaluation

This chapter summarizes the tests that were performed in order to evaluate the proposed transcription system of Yoshii et al., or more precisely its deduced implementation. The primary focus lies on how implementation-specific parameters influence the quality of the transcription. Given this intention and the detailed explanation of the algorithm several questions of interest arise:

- What impact does the number of predefined templates have?
  Is there an optimal number of predefined templates for an instrument?
- What effect does the choice of different templates have?
  Are some templates better than others?
- What influence do the filter functions have?
  Which filter settings are the best?
- What happens when templates with different playing techniques are mixed?

The objective of this chapter is to answer these questions in several tests, whereby each of the stated points is handled in a single test case. The test cases are carried out in the same order as above with the intention to utilise the results of each test case in all subsequent test cases to finally gain a fully optimized transcription system. Before each test case is presented in detail, a description of the utilised performance measures and the compiled test corpora is given.

6.1 Performance Measures

To quantify the performance of an information retrieval system, there are several commonly used scoring metrics. Generally, these metrics measure how well the retrieved information matches the expected information. To determine the recognition quality of the implemented transcription system, the experimental results are evaluated by three performance metrics, namely recall rate, precision rate, and f-measure. These performance metrics derive from the general metrics utilised in information retrieval and are described in more detail below:

recall rate: The recall of a system is the proportion of retrieved information that is relevant to all the relevant information. The recall can be seen as a measure of completeness. Therefore, the formula to measure the recall rate for the drum sound transcription system is given by

$$\text{recall rate} = \frac{\#(\text{correctly detected onsets})}{\#(\text{actual onsets})}. \tag{6.1}$$

precision rate: The precision of a system is the proportion of retrieved information that is relevant to all the retrieved information. The precision can be seen as a measure of exactness or fidelity. Thus, the formula to measure the precision rate for the drum sound transcription system is given by

$$\text{precision rate} = \frac{\#(\text{correctly detected onsets})}{\#(\text{detected onsets})}. \tag{6.2}$$

f-measure: The weighted harmonic mean of precision and recall, the so called f-measure or balanced f-score is

$$\text{f-measure} = \frac{2 \cdot \text{recall rate} \cdot \text{precision rate}}{\text{recall rate} + \text{precision rate}}. \tag{6.3}$$

6.2 Ground Truth

In order to test algorithms that can automatically detect drum events in music signals, ground truth data is needed. The compilation of ground truth and especially the manual annotation of truth files is a time consuming task, thus it is desirable to use an already existing ground truth. This approach also supports a transparent and systematic comparison of different systems, because these can be compared directly.

Research on percussive instrument transcription is relatively new, compared to melody-related research, thus publicly available ground truth for drum processing is rare. In fact, most of the work to date on automatic drum transcription is evaluated by ground truth that has been compiled by researchers themselves. As this ground truth collections are limited in size and their exchange is problematic due to copyright issues, the need for a large and freely available ground truth collection emerged. A more recently published ground truth collection that satisfies this need is the Ecole Nationale Supérieure des Télécommunications (ENST)-drums database[1]. The database was especially created to support the development and evaluation of drum transcription systems, thus it is part of the ground truth utilised in the later test cases. Beside that, it also seemed an obvious choice to use the ground truth from the MIREX 2005 Audio Drum Detection task, at which the proposed transcription system of Yoshii et al. was originally evaluated.

6.2.1 ENST-Drums Database

The ENST-drums database is a large audiovisual database for automatic drum transcription and is freely available for research purposes. The database contains recordings of three

[1] `http://perso.telecom-paristech.fr/~gillet/ENST-drums/`

professional drummers specialized in different music genres. For each drummer five different kinds of sequences have been recorded: individual strokes, phrases, soli, pre-recorded accompaniment, and synthetic accompaniment. In the individual stroke sequence each element of the drum kit is played several times with a few seconds of silence between each hit. The phrases sequence, which provides the majority of data, contains recordings of various popular styles of different tempi and complexity levels. Highly complex phrases, which incorporate all drum instruments of the drum kit, are recorded in the soli sequence. In the two accompaniment sequences, the drummers play along with pre-recorded or MIDI-generated background music. Depending on the musical style, the recordings within each sequence are played either with sticks, rods, brushes, or mallets, which in further consequence increases the diversity of investigable drum sounds. The total duration of audio material recorded per drummer is around 75 minutes. All drum sequences are fully annotated in form of a text file containing a list of time and event pairs. Each drummer plays his his own drum kit and is recorded on eight separate audio tracks which are mixed down to two stereo mixes. The dry mix is created without any processing by simply panning and adjusting the level of each instrument. On the wet mix each instrument is processed by an appropriate equalization and compression. More details concerning the ENST-drums database can be found in [12].

6.2.2 MIREX 2005 Test Set

The participants, or more precisely the proposed algorithms, of the Audio Drum Detection task at MIREX 2005 were evaluated on a test corpus of more than 50 files. The utilised test corpus at that time contained live and sequenced music of many genres with various drum sounds and was a combination of three annotated music collections from Christian Dittmar (CD), Koen Tanghe (KT) [26] and Masataka Goto (MG). Unfortunately, attempts to utilise the complete test corpus failed. Only a training set of 23 files, including a few files from each collection, could be initially collected. However, Christian Dittmar was kind enough to additionally provide his collection of 20 files, hence in total a set of 39 annotated audio files could be compiled. The length of each audio file of the CD and KT collection is 30 seconds, within the MG collection the durations range from about 3 to 6 minutes. Like in the ENST-drums database, all files are annotated in form of a text file containing a list of time and event pairs.

6.3 Test Cases

The test cases in this section are intended to support the evaluation and selection of implementation-specific parameters. Each test case contains a description of the test objective, the steps necessary to perform the test, and a discussion about the obtained test results.

In order to allow an extensive assessment of the algorithm specific parameters for each drum type it was decided to evaluate the kick drum, snare drum, and hi-hat in each test case individually in two different scenarios. On the one hand on audio signals containing only drum tracks and on the other hand on real music signals. Given that the ENST-drums database contains mostly drums-only recordings and the MIREX 2005 test set contains only real world recordings, it was a natural decision to use both ground

truth collections separately for the evaluation in the previously described manner. The ENST-drums database just needed to be recompiled to contain the wet-mix phrases of all three drummers (135 files in total), whereas the MIREX 2005 ground truth could be left unchanged. The utilised drum sound samples for the test cases were taken from a commercial drum sample CD which contains various real kick drum, snare drum, and hi-hat sounds.

For each test case the following conditions apply: First, in the evaluation of a specific instrument, only musical pieces that include that instrument are considered. Second, the maximum time variance for a positive onset detection is ± 25 [ms], that means if the difference between a detected onset time and the actual onset time is within this range, it is assumed that the detected onset time is correct.

6.3.1 Number of Templates

To maximise the coverage of possible timbre variations, the proposed algorithm of Yoshii et al. combines different sounds of an instrument to a single so-called seed template.

The goal of this test case is to clarify how the number of selected sound samples utilised in this process affects the transcription performance. For this purpose n different sound samples have been defined and the results of all k possible combinations of these, without repetition and regard to order, are averaged and examined. This selective operation equals to the binomial coefficient in combinatorics, where

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!} \tag{6.4}$$

describes the number of ways that k objects can be chosen from among n objects (unordered combinations without repetition). The reason for this selective approach and the aggregation of performance measures is to suppress the influence of and between the used samples.

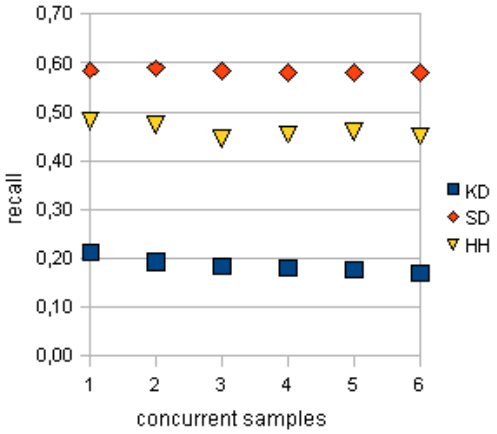
The following example serves as an illustration for the evaluation process: Given $n = 3$ samples labelled as S_1, S_2, S_3 and the goal to retrieve k-subsets we get:

$$\begin{array}{ll} \{S_1\}, \{S_2\}, \{S_3\} & \text{for } k = 1 \\ \{S_1, S_2\}, \{S_1, S_3\}, \{S_2, S_3\} & \text{for } k = 2 \\ \{S_1, S_2, S_3\} & \text{for } k = 3 \end{array}$$

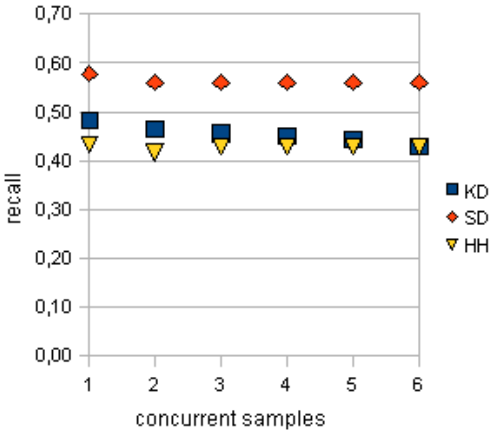
The sample combinations, obtained this way, are then used as input for the transcription system. In order to finally evaluate the performance of each subset, the performance results of all elements (combinations) of subset k are summed up and divided by the total number of combinations $\binom{n}{k}$ of that subset.

To keep the computational costs reasonable for this test case, the performance results of $(1 \leq k \leq 6)$ simultaneously chosen samples from a set of $n = 6$ samples were evaluated. Figure 6.1 shows the performance results for each transcribed instrument obtained for the ENST-Drums and MIREX 2005 ground truth.

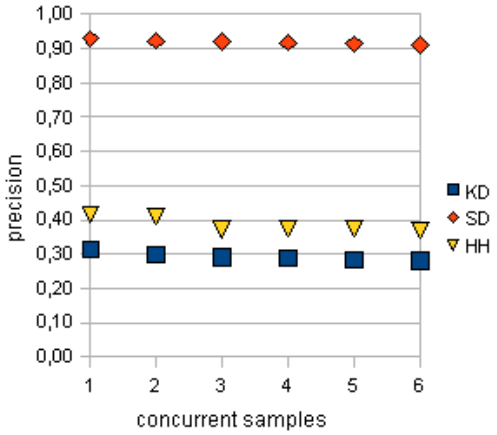
The results suggest that the transcription performance does not necessarily increase with the number of concurrently used samples, contrariwise the performance is more likely to decrease. It is also notable that the results between the monophone and polyphone ground truth seem to be independent. In the monophone corpus the f-measure values



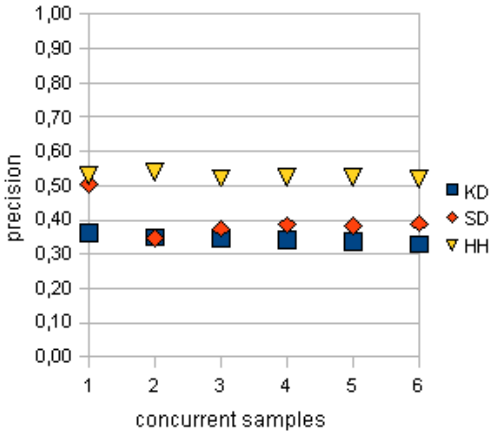
(a) recall, ENST-drums ground truth



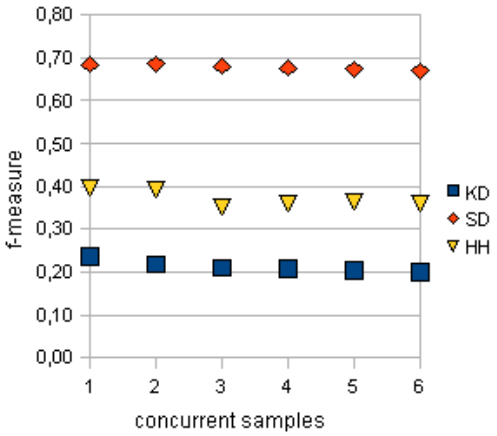
(b) recall, MIREX 2005 ground truth



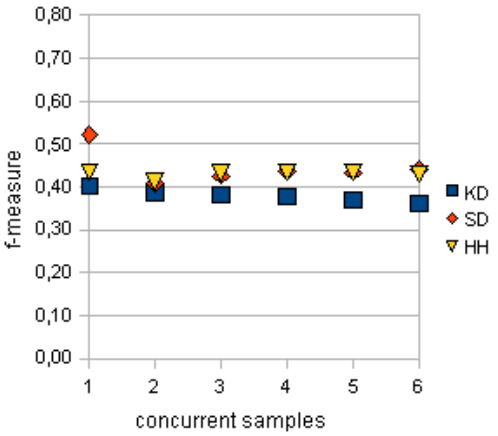
(c) precision, ENST-drums ground truth



(d) precision, MIREX 2005 ground truth



(e) f-measure, ENST-drums ground truth



(f) f-measure, MIREX 2005 ground truth

Figure 6.1: The performance scores according to the number of concurrently used samples.

hardly change, whereas in the polyphone corpus the f-measure values are subject to larger fluctuations. However, a common observation is that regardless for which instrument and ground truth, the f-measure has its maximum in the case of a single utilised sample. That suggests, that the transcription system performs best with only one sample.

This result stands in contradiction to the expected result, where the performance should increase with the number of concurrently used samples until it should suddenly drop because of overfitting. The overfitting is expected, because the algorithm of Yoshii et al. creates the seed template by selecting the maximum power at each frame and each frequency bin among the power spectrograms of the predefined sound samples (see equation 5.1). Logically, the combination of a large number of samples, which have their power maxima at different positions in the spectrogram, would create a power spectrogram with a flat line of maxima so no characteristic features would be left.

In order to find an explanation for the obtained results, it is advantageous to first recall the template adaptation procedure or more specifically the segment selection approach given in section 5.4.2. In the first iteration of the adaptation procedure, where the seed template has never been adapted, spectral smoothing is applied (see equation 5.5) in order to allow an adequate distance calculation. This process could be the cause for the obtained results, as it blurs the spectral information of the seed template, which in further consequence seems to make the aggregation of samples rather dispensable.

6.3.2 Template Selection

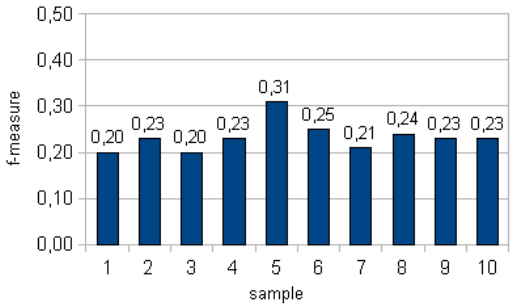
The objective of this test case is to identify how different samples affect the performance of the transcription. For this purpose the transcription system has been fed with 10 different samples, of which the first 6 samples are identical to the ones used in the first test case. The resulting performance measures for each instrument obtained for the ENST-drums and the MIREX 2005 ground truth are depicted in figure 6.2.

It can be seen that the performance results within each instrument type vary between the used samples. That indicates that the transcription performance depends on the used samples. The variance for the f-measure is in most cases within the range of a few percent, though differences up to 15% could be observed in figure 6.2(d). Unlike the first test case, it seems that there is a correlation between the results for monophone and polyphone ground truth which in further consequence suggests that generally applicable samples exist.

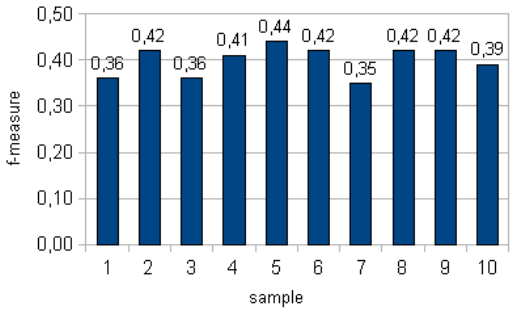
The results of this test case concur with the expected results, as it is understandable that different samples with different frequency spectra lead to different selections of segments for the seed-template aggregation in the first iteration of the template adaptation phase. However, this effect should be minimized through the use of spectral smoothing (see equation 5.5), which is for the most part confirmed by the low variations that have been obtained for the f-measure.

6.3.3 Filter Characteristics

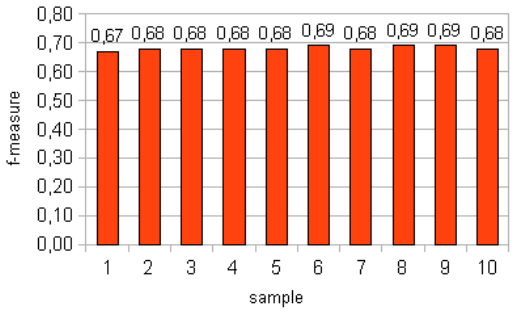
Yoshi et al. use a lowpass filter for the kick and snare drum and highpass filter for the hi-hat in their transcription algorithm. The cutoff frequency and the rate of frequency rolloff for the filters are defined in bins (see figure 5.3). With a sampling frequency of 44100 Hz and a window length of 4096 samples for the STFT the bin spacing is approximately 10.8



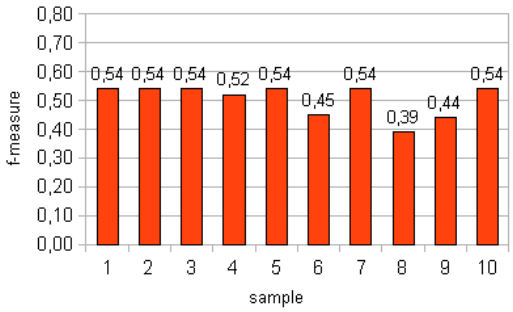
(a) kick, ENST-drums ground truth



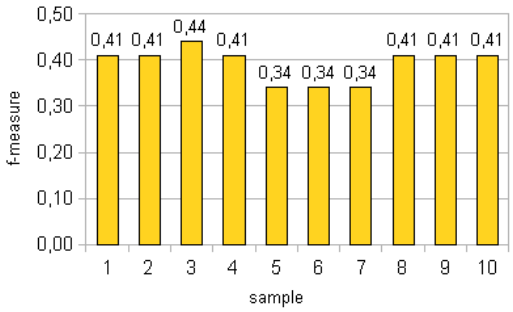
(b) kick, MIREX 2005 ground truth



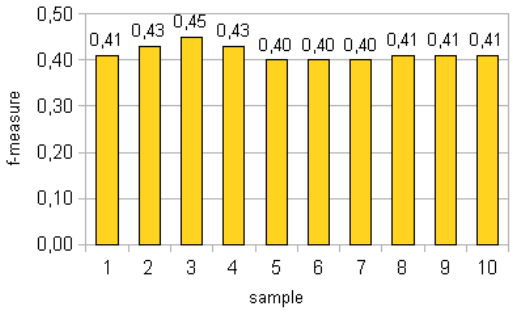
(c) snare, ENST-drums ground truth



(d) snare, MIREX 2005 ground truth



(e) hi-hat, ENST-drums ground truth



(f) hi-hat, MIREX 2005 ground truth

Figure 6.2: The sample-dependent f-measure scores obtained for the ENST-drums and MIREX 2005 ground truth.

Hz, thus the recommended cutoff frequencies for each instrument are: kick drum 810Hz (bin 75), snare drum 2160 Hz (bin 200), and hi-hat 1620 Hz (bin 150).

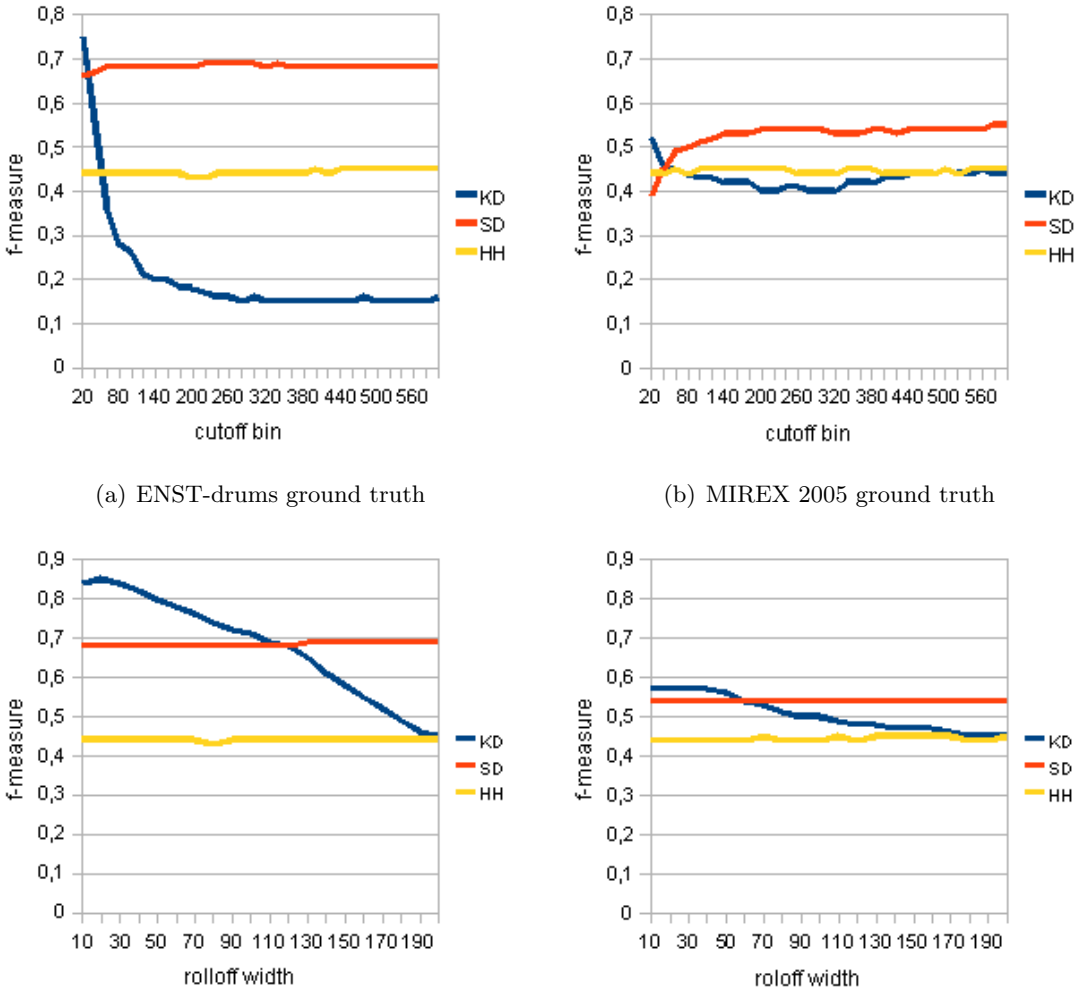
The objective of this test case is to clarify how the filter characteristics affect the performance of the transcription. Before this question can be answered, it is necessary to determine the characteristic frequency ranges for kick drum, snare drum, and hi-hat in order to define appropriate test settings. As it turns out, no definite answer for the frequency ranges of these instruments can be given. Research in this context revealed that there are a few frequency range estimations for the mentioned instruments, but those are not consistent and beyond that very rough. Thus, it was assumed that the timbre variations of the drums allow only a proper estimation for a broad frequency range. An equitable approximation could look like this: kick drum 30 Hz - 6 kHz, snare drum 70 Hz - 14 kHz, cymbals 300 Hz - 14 kHz.

Given the description of the template adaptation and matching algorithm it is supposed that an exact definition of the frequency ranges is not significant, rather it should be ensured that the filters cover a reasonable timbre spectrum. Further, it is reasonable to conclude that the filter ranges of the instruments should not overlap, so that during the processing of segments characteristic frequency bins are not mistaken. Interestingly, this conclusion is not shared by Yoshii et al. which is embodied in the use of a lowpass filter for kick and snare drum. It is believed that this approach was chosen for the sake of convenience with the assumption that the characteristic frequencies of the snare drum are more prominent then those of the kick drum.

On further reflection, it was decided to evaluate the effects of different cutoff frequencies first and subsequently use the obtained results to study the effects of different rolloff settings. To identify proper cutoff frequencies for each instrument filter first, the performance results have been logged while the cutoff frequencies subsequently altered from bin 20 to bin 600 with a step size of 20. Because of the insights of the previous two test cases, only one sample for each instrument has been used during this evaluation. The samples that have been chosen in this context to fit the monophone and polyphone ground truth as effectively as possible are: kick no. 5, snare no. 5, hi-hat no. 3.

The results for the experimental filter settings, obtained in the described manner, are depicted in figure 6.3(a) for the ENST-drums ground truth and in 6.3(b) for the MIREX ground truth. It can be seen that the transcription performance improved considerably for the kick drum due to the lower cutoff frequency of the lowpass filter. Additionally, the results suggest that it is best to keep the cutoff frequency as low as possible. For the snare drum, when looking at figure 6.3(b), it can be seen that the recommended value for the cutoff frequency (bin 200) fits very well since the f-measure reaches its full potential at that mark. A different behaviour is observable for the hi-hat, where the filter settings do not seem to have much influence. This may be due to the large frequency spectrum that is covered by the filter and the smoothing which is additionally applied in the detection of the hi-hat. Within the scope of these observations, the cutoff frequencies for the snare drum and hi-hat have been kept and the cutoff frequency for the kick drum has been lowered to bin 20 (216 Hz).

To identify a proper rolloff width for each instrument filter, the previously determined cutoff frequencies have been applied while the rolloff width subsequently altered from bin 10 to bin 200 with a step size of 10. The results for the rolloff width settings are depicted in figure 6.3(c) for the ENST-drums and in 6.3(d) for the MIREX ground truth.



(a) ENST-drums ground truth

(b) MIREX 2005 ground truth

(c) ENST-drums ground truth

(d) MIREX 2005 ground truth

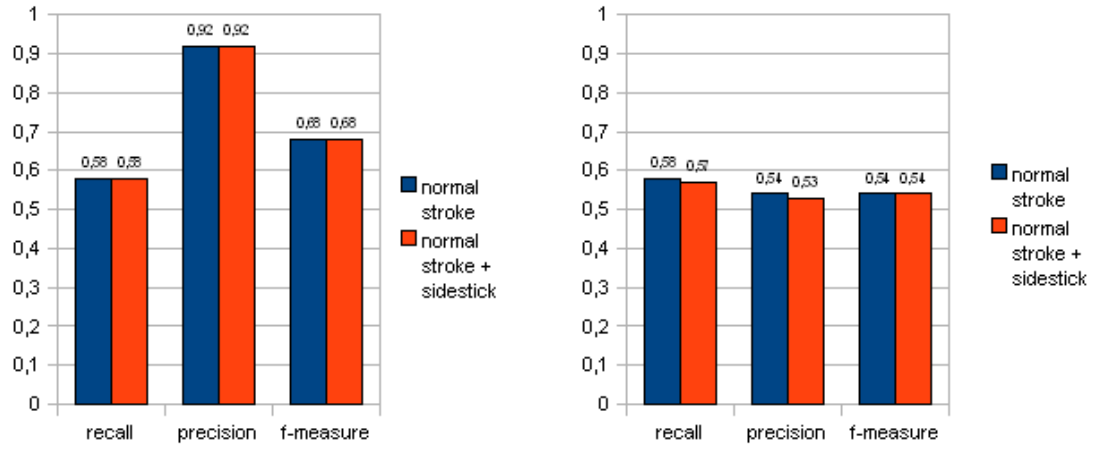
Figure 6.3: The influence of different filter settings on the f-measure scores obtained for the ENST-drums and MIREX 2005 ground truth.

The conclusions that can be drawn from the graphs are that the exact rolloff settings for the snare and hi-hat are quite unimportant, whereas the rolloff settings for the kick drum do play an important role. In fact, it is favourable to use the lowest possible rolloff setting. This result supports the previous observation for the cutoff frequency of the kick drum which implies that a focus on very low frequency bins gains the best results.

6.3.4 Mixture of Different Playing Technique Samples

The final test case deals with the question how the transcription system performs when different playing style samples are mixed. This question is in particular interesting for the snare drum, since it is often played in different ways.

To approach the prefacing question, it was decided to evaluate the performance results of the snare drum in two different scenarios. The first scenario uses a normal stroke snare drum sound sample as input for the transcription system, whereas the second scenario uses that same sample plus another one containing a sidestick stroke snare drum sound. The performance results of both scenarios, tested for the ENST-drums and MIREX 2005 ground truth, are depicted in figure 6.4.



(a) ENST-drums ground truth

(b) MIREX 2005 ground truth

Figure 6.4: Performance scores comparison between a normal snare stroke sample and its mixture with a sidestick stroke sample obtained for the ENST-drums and MIREX 2005 ground truth.

The obtained results suggest that the mixture of different playing style samples has very little impact on the recognition performance. This can be explained by the algorithm design which adapts the seed template to that sound of a given audio file which is most similar to it. That further implies that the algorithm is tied to a single sound and is not able to concurrently adapt to different variations of a drum sound generated by different playing techniques. Therefore, the mixture of different playing style samples is unrewarding.

6.3.5 Discussion

Several conclusions can be drawn from the above test cases. First, it seems not productive to mix different samples in order to cover a broader range of timbre variations, because in the first adaptive iteration the seed sample is blurred anyway. Second, it appears that the transcription performance is dependent from the given samples, thus some samples are more suitable for transcription than others. Moreover, it is even likely that generally applicable samples exist, this is underlined by the fact that the obtained performance for the evaluated samples correlated between the ENST-drums and MIREX 2005 ground truth. Third, the results suggest that the lowpass cutoff frequency and the rolloff settings for the kick drum should be lower in order to avoid disruptive influence from the snare drum. Fourth, the mixture of different playing style samples does not improve the transcription performance. This is due to the adaptive algorithm design, which implies that only a single sound can be adapted.

The results further suggest that the most difficult detection task among the three instruments is the one for the hi-hat. The reason is probably that cymbals have a very broad frequency range and thus they are more vulnerable to overlapping sounds. Further, there are different playing styles for the hi-hat ranging from closed to open and usually these occur in a mixed way during drum performances. Since the current implementation uses only a single hi-hat template, that template could not cover all spectral variations of those playing styles.

Some general properties of the transcription system should also be mentioned: The algorithm always assumes that all analysed drums are present. In other words, the algorithm is not able to detect that a drum sound does not occur in a given signal. If a drum is not present, the adaptation procedure simply selects the sound that is most similar to that drum. Another property of the algorithm is that the transcription results for a song and an excerpt of that same song can be different, because the available data for the threshold calculation differs. It is also worth to mention that the algorithm is only capable to adapt to a single timbre variation of a drum sound. That means when drum sounds greatly vary within a musical piece, due to playing techniques or electronic effects, the algorithm is not able to adapt to the individual variations.

Chapter 7

Conclusions and Future Work

The information derived from percussive events is an important prerequisite, not only for the creation of musical scores, but also for the extraction of semantic meaningful metadata, such as tempo and musical meter. Drum events provide important clues about the rhythmical organisation of a musical piece and for many music genres nowadays, rhythmic structures have become at least equally important as melodic or tonal structures.

In this thesis an overview of the current state of the art in unpitched percussion transcription has been given. This encompassed both event-based and separation-based systems. Although the proceedings in the area of percussive instrument transcription evolved considerably and showed promising results over the last years, it is still a fact that the human auditory system is far away from being mimicked by computers.

To date, the best transcription results are gained with systems that focus only on a few instruments: kick drum, snare drum, and the hi-hat in the transcription of drums-only signals, and kick and snare drum in the transcription of polyphonic music. This approach is not surprising, because it reduces the complexity of the overall transcription problem. Nonetheless, this simplification does consider the most important instruments, and therefore represents a good starting point for more general systems in the future.

Recent work on percussion transcription shows a tendency to the usage of adaptive approaches, which take certain characteristics of the input signal into account when attempting transcription, both in event-based and separation-based systems. Work in this field is driven by the assumption that a system that is trimmed to individual signals is more likely to produce a successful transcription than a system which makes use of general models. This approach can be seen as an attempt to master the large timbre variations of drums. The presented work as well as the implemented transcription system in this thesis serve as example for this progress.

The obvious direction for future work lies in extending the abilities of transcription systems to detect more drums. New methods or adaptations of existing approaches are needed to deal with an increased number of different types of percussion instruments. Today's methods, or more precisely their mathematical models, do not allow a reliable distinction between all instruments in a drum kit. Future improvements should also target the refinement of detection methods in order to allow the discrimination of subtle but significant differences, such as between open and closed hi-hats, as well as between different types of cymbals and playing techniques (like sidestick or rimshot on the snare drum). The room for improvements is large, considering that many systems only deal with low-level

recognition. The predictability of percussion patterns could be exploited by means of musicological modelling, leading to better transcription systems.

In conclusion, this work has shown a number of possibilities to solve the problem of drum transcription. The implemented transcription system demonstrates that pattern-recognition techniques in combination with instrument model adaptation can be used to transcribe the kick drum, snare drum, and hi-hat for drums-only and real music signals. An extensive evaluation revealed assets and drawbacks of the implementation and showed that the consideration of the following list of insights on implementation specific settings can save unnecessary expenses and improve the overall performance of the transcription system: First, it seems not productive to mix different samples in order to cover a broader range of timbre variations because of the smoothing process during the template adaptation procedure. Second, it appears that the transcription performance is dependent on the given samples and it is likely that generally applicable samples exist. Third, the lowpass filter for the kick drum should focus on very low frequency bins to avoid disruptive influence from the snare drum. Fourth, due to the adaptive algorithm design only a single sound can be adapted, therefore the mixture of different playing style samples is unrewarding.

The presented work shall be understood as a first approach to solve the problem of percussive instrument transcription and it is hoped that future work will further enhance the accomplishments to date.

Appendix A

Abbreviations

CASA	Computational Auditory Scene Analysis
CD	Compact Disc
ENST	Ecole Nationale Supérieure des Télécommunications
GMM	Gaussian Mixture Model
GUI	Graphical User Interface
HMM	Hidden Markov Model
ICA	Independent Component Analysis
ISA	Independent Subspace Analysis
ISMIR	International Conference on Music Information Retrieval
MFCC	Mel Frequency Cepstral Coefficient
MIDI	Musical Instrument Digital Interface
MIR	Music Information Retrieval
MIREX	Music Information Retrieval Evaluation eXchange
NNSC	None-Negative Sparse Coding
NMF	Non-Negative Matrix Factorisation
PCA	Principal Component Analysis
PSA	Prior Subspace Analysis
RMS	Root Mean Square
STFT	Short Time Fourier Transform
SVM	Support Vector Machine

Bibliography

[1] C. Dittmar. Drum detection from polyphonic audio via detailed analysis of the time frequency domain. In *1st Music Information Retrieval Evaluation eXchange (MIREX)*, 2005.

[2] C. Dittmar, K. Dressler, and K. Rosenbauer. A toolbox for automatic transcription of polyphonic music. In *Audio Mostly 2007 2nd Conference on Interaction with Sound*, Illmenau, Germany, September 2007.

[3] C. Dittmar and C. Uhle. Further steps towards drum transcription of polyphonic music. In *Audio Engineering Society 116th Convention*, Berlin, Germany, May 2004.

[4] S. Dixon. An interactive beat tracking and visualisation system. In *Proceedings of the International Computer Music Conference*, pages 215–218, 2001.

[5] D. FitzGerald, E. Coyle, and B. Lawlor. Prior subspace analysis for drum transcription. In *Audio Engineering Society 114th Convention*, Amsterdam, Netherlands, March 2003.

[6] D. FitzGerald, B. Lawlor, and E. Coyle. Sub-band independent subspace analysis for drum transcription. In *Internatinal Conference on Digital Audio Effects*, Hamburg, Germany, 2002.

[7] D. FitzGerald, R. Lawlor, and E. Coyle. Drum transcription in the presence of pitched instruments using prior subspace analysis. In *Irish Signals & Systems Conference*, Limerick, Ireland, July 2003.

[8] D. FitzGerald, R. Lawlor, and E. Coyle. Drum transcription using automatic grouping of events and prior subspace analysis. In *4th European Workshop on Image analysis for Multimedia Interactive Services*, pages 306–309, 2003.

[9] D. FitzGerald and J. Paulus. Unpitched percussion transcription. In Klapuri and Davy [20], pages 131–162.

[10] O. Gillet and G. Richard. Automatic labelling of tabla signals. In *Internatinal Conference on Music Information Retrieval*, Baltimore, USA, October 2003.

[11] O. Gillet and G. Richard. Automatic transcription of drum loops. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, volume 4, pages iv–269–iv–272 vol.4, 2004.

[12] O. Gillet and G. Richard. Enst-drums: an extensive audio-visual database for drum signals processing. In *Proceedings of the 7th International Conference on Music Information Retrieval*, pages 156–159, 2006.

[13] 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.

[14] M. Goto and Y. Muraoka. A beat tracking system for acoustic signals of music. In *ACM International Conference on Multimedia*, pages 365–372, San Francisco, USA, October 1994.

[15] M. Goto and Y. Muraoka. A sound source separation system for percussion instrument. *The transactions of the Institute of Electronics, Information and Communication Engineers*, 77(5):901–911, 1994.

[16] F. Gouyon, F. Pachet, and O. Delure. On the use of zero-crossing rate for an application of classification of percussive sounds. In *International Conference on Digital Audio Effects*, Verona, Italy, December 2000.

[17] S. Hainsworth. Beat tracking and musical metre analysis. In Klapuri and Davy [20], pages 101–129.

[18] P. Herrera, A. Dehamel, and F. Gouyon. Automatic labeling of unpitched percussion sounds. In *Audio Engineering Society 114th Convention*, Amsterdam, Netherlands, March 2003.

[19] P. Herrera, A. Yeterian, and F. Gouyon. Automatic classification of drum sounds: A comparison of feature selection methods and classification techniques. In *International Conference on Music and Artificial Intelligence*, pages 69–80, Edinburgh, Scotland, September 2002.

[20] A. Klapuri and M. Davy, editors. *Signal Processing Methods for Music Transcription.* Springer, New York, 2006.

[21] J. Paulus and T. Virtanen. Drum transcription with non-negative spectrogram factorisation. In *European Signal Processing Conference*, Antalya, Turkey, September 2005.

[22] J. K. Paulus and A. P. Klapuri. Conventional and periodic n-grams in the transcription of drum sequences. In *IEEE International Conference on Multimedia and Expo*, volume 2, pages 737–740, Baltimore, USA, July 2003.

[23] V. Sandvold, F. Gouyon, and P. Herrera. Percussion classification in polyphonic audio recordings using localized sound models. In *International Conference on Music Information Retrieval*, Barcelona, Spain, October 2004.

[24] J. Sillanp, A. Klapuri, J. Seppnen, and T. Virtanen. Recognition of acoustic noise mixtures by combined bottom-up and top-down processing. In *European Signal Processing Conference*, 2000.

[25] K. Tanghe, S. Degroeve, and B. De Baets. An algorithm for detecting and labeling drum events in polyphonic music. In *MIREX evaluation campaign*, 2005.

[26] K. Tanghe, M. Lesaffre, S. Degroeve, M. Leman, B. De Baets, and J.-P. Martens. Collecting ground truth annotations for drum detection in polyphonic music. In *Internatonal Conference on Music Information Retrieval*, pages 50–57, 2005.

[27] D. Van Steelant, K. Tanghe, S. Degrove, M. Baets, B. De Leman, and J.-P. Martens. Classification of percussive sounds using support vector machines. In *Machine Learning Conference of Belgium and The Netherlands*, Brussels, Belgium, January 2004.

[28] T. Virtanen. Sound source separation using sparse coding with temporal continuity objective. In *International Computer Music Conference*, Singapore, Malaysia, 2003.

[29] K. Yoshii, M. Goto, and H. G. Okuno. Adamast: A drum sound recognizer based on adaptation and matching of spectrogram templates. In *Matching of Spectrogram Templates, ISMIR 2004*, pages 184–191, 2004.

[30] K. Yoshii, M. Goto, and H. G. Okuno. Automatic drum sound description for real-world music using template adaptation and matching methods. In *International Conference on Music Information Retrieval*, Barcelona, Spain, October 2004.

[31] K. Yoshii, M. Goto, and H. G. Okuno. Drum sound identification for polyphonic music using template adaptation and matching methods. In *ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing*, Jeju, Korea, 2004.

[32] 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, January 2007.

[33] A. Zils, F. Pachet, O. Delerue, and F. Gouyon. Automatic extraction of drum tracks from polyphonic music signals. In *International Conference on Web Delivering of Music*, Darmstadt, Germany, December 2002.

CURRICULM VITAE

Personal Information

Name	Alex Maximilian Wöhrer
Date of Birth	30th May 1983
Nationality	Austrian

Education

since 2006	University of Vienna Master's Program for Media Computer Science
2003 – 2006	FH Technikum Wien (University of Applied Sciences) B.Sc. in Business Informatics
1997 – 2002	HTBLA Wien Donaustadt (Higher Technical Education Institute) Faculty for Computer Engineering
1993 – 1997	Höhere Internatschule des Bundes Wien (Secondary School)
1989 – 1993	Volksschule Deutsch Wagram (Elementary School)

Cross-Cultural and Social Experience

2004 – 2005	University of Dalarna, Sweden Semester abroad within the ERASMUS programme
2002 – 2003	Samariterbund Gruppe Floridsdorf-Donaustadt (Civilian Service) Paramedic

Teaching Experience

2008 - 2009	University of Vienna Tutor for courses on Mobile Computing and Networked Systems