DETECTING AND DESCRIBING PERCUSSIVE EVENTS IN POLYPHONIC MUSIC

Martin Haro Berois

Master Thesis submitted in partial fulfillment of the requirements for the degree:

Master in Information, Communication and Audiovisual Media Technologies.

Supervisor: Perfecto Herrera

UNIVERSITAT POMPEU FABRA

Music Technology Group

Barcelona
September 2008

DETECTING AND DESCRIBING PERCUSSIVE EVENTS IN POLYPHONIC MUSIC

Martin Haro Berois

Submitted for the degree of Master in Information, Communication
and Audiovisual Media Technologies
September 2008

Abstract

The present thesis deals with the automatic description of percussive events in "real world" polyphonic music. After taking a pattern recognition approach we evaluate "bag of frames" and "object-level temporal evolution" descriptors extracted from 153 frame-level features adding up a total of more than 1450 descriptors. Three binary instrument-level support vector machines models are built from a training set of more that 100 songs and 10 genres. We observe an improvement in the classification results when object-level temporal evolution descriptors are added to the feature set. Then we evaluate the binary models within a whole drum transcription system achieving comparable results with state of the art algorithms. Finally we present 17 mid-level percussion descriptors and evaluate their usefulness among MIR tasks like genre classification, danceability estimation and Western non-Western discrimination. We conclude that the presented percussion-related descriptors provide complementary information to "classic" descriptors that could help in the previously mentioned MIR tasks.

Declaration

The work in this thesis is based on research carried out at the Music Technology Group. No part of this thesis has been submitted elsewhere for any other degree or qualification and it all my own work unless referenced to the contrary in the text.

Acknowledgements

First of all I would like to thank Perfecto Herrera for his constant support during the whole process of the thesis. Without his guidance this work could not be done. I am also grateful to Xavier Serra for his support and the opportunity to be part of the music technology group. Furthermore, special greetings to Gerard Roma, Jordi Janer , Vasilis Pantazis and to the "aula 316 and beyond" people for their help, comments and suggestions.

Finally I would like to thank all my family, especially Ximena and Nibia.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> “If we use memory as a basis of comparison, the IBM 704 we first used in Bell labs back in 1967 would worth today 12 cents, and any today’s typical laptop would worth back in 67 ... 59 billion dollars.” -John M. Chowning. November 29, 2007

> “Any sufficiently advanced technology is indistinguishable from magic.” -Arthur C. Clarke, "Profiles of The Future", 1961 (Clarke’s third law)

1.1 Scope

During the last decade we have witnessed an exponential growth of music listeners. We are about to reach a point in history where all the music ever recorded could be stored in a single pocket-size gadget. Nowadays on-line music providers offer millions of songs to their costumers. Internet radios provide with twenty four hours of more or less personalized programs and P2P networks make easy to share music with people around the globe. This growth on music availability and music listeners comes with a downside: the titanic task of choosing, and discover, music from a collection of millions of songs [55].

The music information retrieval (MIR) field is becoming more and more efficient in providing tools for choosing, discovering and understanding the music we like. However we are still far from bridging the semantic gap between automatic music analysis (at signal level) and automatic music description (going from sound to sense [39]).

The main idea behind automatic music description is to extract, from a musical audio signal, a representation of what is happening inside the song. The goal is to get a system that automatically extracts relevant information (descriptors) from the signal. These music descriptions do not have the priority to be faithful scores. The point is to obtain “predicates” that apply to a given music excerpt and usually this information goes beyond traditional music scores. Examples of descriptors could be: pitch, duration, type of instruments, chords, overall structure of the song, genre, rhythm or main melody. Herrera et al. [28] call this way of understanding music transcription as the “descriptionist” approach , as different to the more traditional “transcriptionist” approach for which obtaining a faithful score is the final goal.

Today most advanced music transcription systems are still far, in terms of accuracy and flexibility, what skilled human musicians can do [33]. Since the beginning of this research field a great amount of work has been done in the transcription of pitched instruments. During the last decade the interest in the transcription of unpitched instruments has grown and most of the work has focused in the problem of drum[1] transcription [12] [28]. The aim of drum transcription systems is to obtain from an audio signal a representation of the type of percussion instrument played (instrument recognition), and when it has been played (temporal location). Even more, if we take a “descriptionist” approach, it could address the task of determine the amount of "percussivity" of the signal, the strength of the hits, the type of beating (e.g. brushes, sticks), etc.

The presence of unpitched percussion instruments[2] is at least as important as melodic instruments in Western music and even more important for non-Western compositions. A great number of important characteristics of musical signals can be extracted from percussion descriptors. This descriptors, alone or combined with complementary descriptors obtained from pitched events, can be used in order to determine key aspects of a given music signal like tempo, rhythm, genre, general

[1] The word “drum” it refers to the standard Rock/Pop drum kit found in Western music.

[2] The term unpitched percussive instruments refers to percussive instruments that do not transmit a clear sense of pitch (e.g. cymbals, snares, bass drums, etc.); xylophones or marimbas are not consider unpitched percussion because they transmit a sense of pitch to the listener.

structure, rhythmic patterns or mood. This information is relevant not only in the musicological and musicians-oriented fields but also in the development of more general applications like music browsing, query by rhythm, query by humming, query by beat-boxing, recommendation, similarity, cover song identification, etc.

This thesis intends to be a step forward in solving the problem of detecting and describing percussive events and can be understood as a continuation on the work done in [31] and [51].

1.2 Structure

The following chapters are structured as follows: in chapter 2 we introduce the scientific background i.e. description of drum kit instruments, the state of the art on percussion transcription, commonly used descriptors, etc. Chapter 3 presents the research problems, proposed experiments and units of study. In the following four chapters we develop the proposed experiments by introducing specific background for each experiment, its methodology and its main results and conclusions. Thus chapter 4 is devoted to database selection and normalization experiments, chapter 5 to experiments with object-level temporal evolution descriptors. In chapter 6 we evaluate the capabilities of our system to transcribe drum events and in chapter 7 we present and evaluate mid-level percussion descriptors. Finally chapter 8 is devoted to overall conclusions and further work.

Chapter 2

Scientific Background

In spite of the crucial role of percussion instruments in most of popular music styles and their increasing importance in the music of the twentieth century there is a lack of scientific research on percussion instruments not only from the point of view of acoustics, but also in music content processing and automatic instrument identification. One exception to this rule is the excellent work done by Thomas Rossing, one of the most important acousticians of the current times. Professor Rossing writes in the preface of his book *Science of percussion instruments*:

> "Although percussion instruments may be the oldest musical instruments (with the exception of the human voice), relatively little has been written about scientific research on this instruments.... Because the sounds of percussion instruments change so rapidly with time, their study and analysis require equipment that wasn't widely available until quite recently." [49]

2.1 The Drum kit

Due to the importance of the drum kit in western popular music we decide to concentrate our research efforts of detection and description of percussive instruments on this particular set of drums.

The standard drum kit is usually formed by (see figure 2.1):

- a bass drum or kick (1)

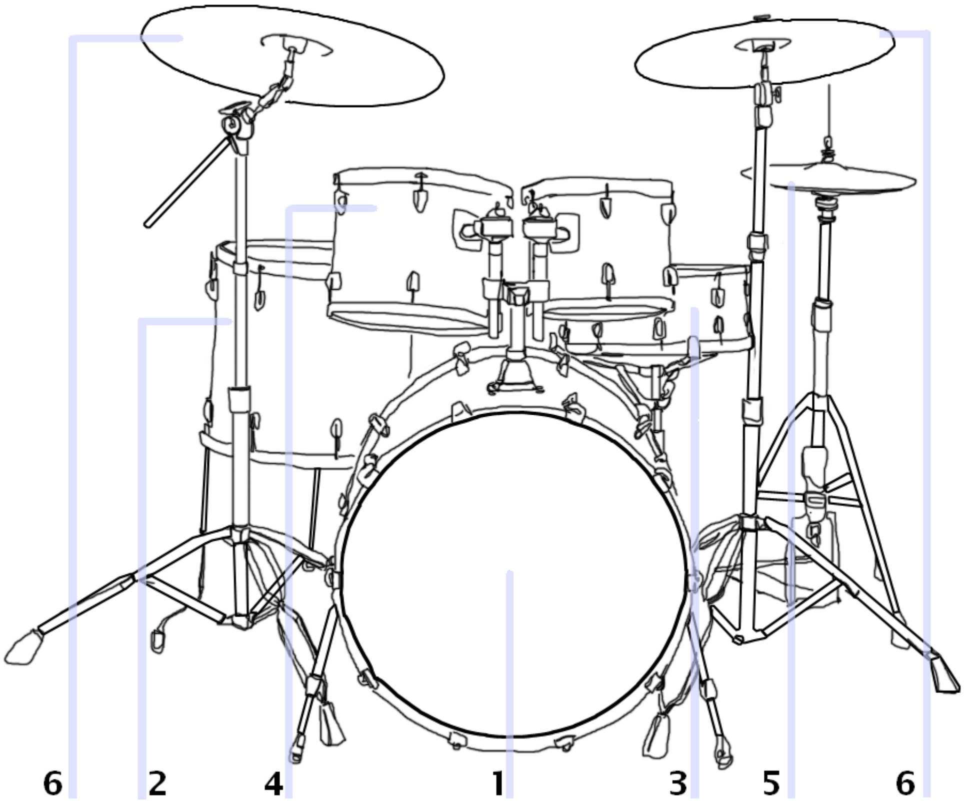


Figure 2.1: The drum kit. Source: wikimedia http://upload.wikimedia.org/wikipedia/commons/6/6d/Drum_kit_illustration_edit.png

- a snare drum (3)
- two tom-toms mounted on top of the bass drum (4)
- a floor tom (2)
- a hi-hat (5)
- a ride and a crash cymbal (6).

This set can vary according to the music style and drummer's personal preferences. Following the classical Hornbostel-Sachs categorization of musical instruments [65], drum kit instruments can be grouped in two categories namely: membranophones and idiophones.

Next we present an overview on each drum kit instrument (see [49] for a deeper analysis on the acoustics of these instruments).

Membranophones:

This category is formed by instruments which sound is produced by the vibration of a stretched membrane mounted over a hollow body. Bass drum, snare drum and tom-toms are included in this category and they can present one or two membranes each (one or two-head mounted).

Although these instruments are unpitched they can be “tuned” by combining stretching tensions on their heads. One-head instruments have more definite pitch than two-head ones.

- **Bass drum:** The bass drum or “kick” drum is, within the drum kit, the largest instrument with sizes going from 16" to 26" diameter and depths of 16" to 22". It is played using a pedal-operated mallet or beater and it produces a loud and short sound which energy is mainly below 150Hz.
- **Tom-toms:** Their sizes vary from 8” to 18” diameter. The small ones (normally 12” and 13”) are mounted over the bass drum while the big one (usually 16”) is mounted on three legs (floor tom). Tom-toms, as a consequence of their sizes, have a relative sense of pitch. The majority of their spectral energy can be found between 50 and 200 Hz.
- **Snare drum:** It is a two-head drum with the special characteristic of having strands of wire or gut (the snares) stretched over the low membrane. This characteristic confers its particular sound. Snares are typically 14” diameter and 5” to 8” deep, and can be made of metal, wood or acrylic material. Their energy spread over almost all frequencies but most of it is usually between 100 and 500 Hz.

Idiophones:

Within this category we found all percussion instruments that produce sound by the vibration of their own bodies. All drum kit’s cymbals are included here like hi-hats, ride, crash and splash cymbals among others.

- **Hi-hat:** It is a type of cymbal that consists of two plates mounted on a stand (one on top of the other) and operated by a foot pedal that clashes both plates. It can be struck when both plates are closed (with the pedal) or open (without the pedal) and the produced sound for each mode differs in frequency range and duration. Its timbre can also be modified by varying the gap between both plates. Hi-hat sizes are usually between 13" and 15" and, like all cymbals, their energy range goes through almost all the spectrum with most of the energy within the high frequency range.

- **Crash, Ride and Splash cymbals:** Within the cymbals' family there are many variations on sizes and sounds. Three commonly used cymbals are the Crash (with sizes from 13" to 19", loud and sharp sound) the Splash (from 6" to 12", short and splashy sound) and the Ride (18" to 24", sustained and shimmering sound). Their energy goes through all the spectrum on the first few milliseconds after the strike to concentrate on the high frequency range afterwards.

The main spectral differences among drum kit instruments can be seen in figure 2.2 where the x axis is represented in Barks[1] and the y axis corresponds to instrument's normalized energy. This plot is computed taking the average Bark values from isolated drum sounds (4,209 for bass drum, 3,043 for snare, 6,316 for hi-hat and 5,762 for cymbals). It shows the strong spectral differences between membranophones and idiophones, belonging the first class to the low level spectra and the latter to the high frequency ranges. Some subtle differences between bass drum, tom-toms and snares can be also appreciated from this averaged Bark values being bass drums the lower in frequency followed by tom-toms and snares. It is also interesting to remark the presence of an isolated small peak of energy in the snare drum values, belonging to Bark band number 21 (4,400 to 5,300 Hz). We speculate about the presence of this energy as being generated by the vibration of the actual snares. The spectral and time evolution for bass drum, snare drum and open hi-hat can be seen in figure 2.3.

[1] See section 2.3 for and explanation on Bark scale

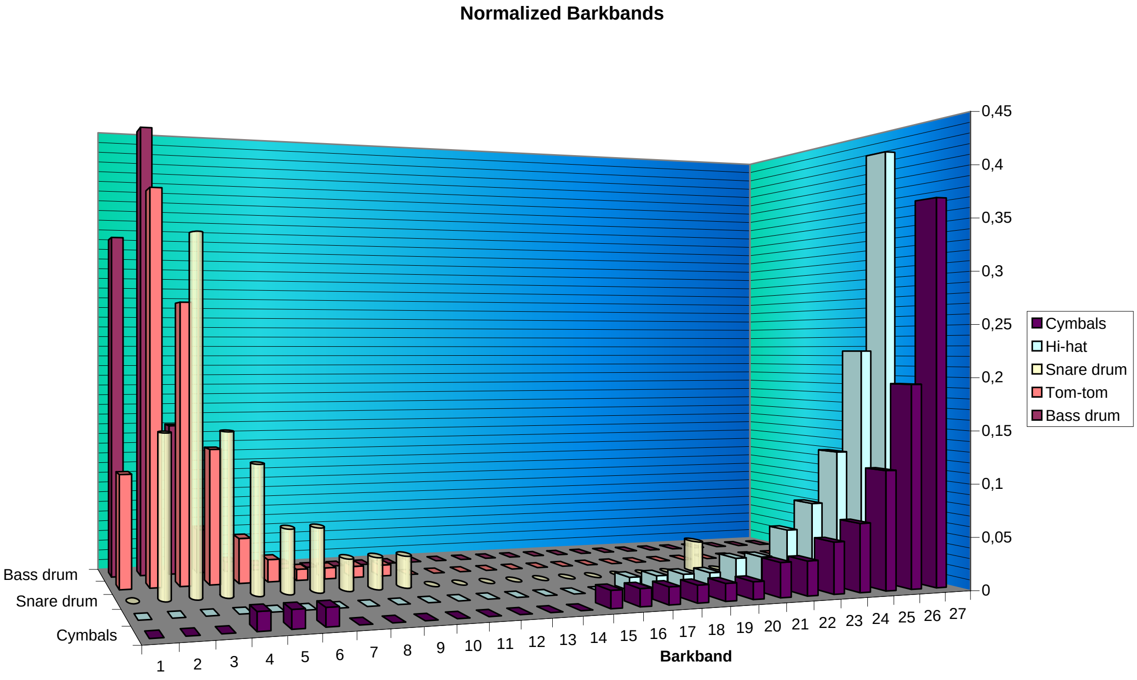


Figure 2.2: Average energy for drum kit instruments. x axis in Barkbands and y axis in normalized energy

2.2 State of the art on percussion transcription

According to [12] most of the work done in percussion transcription can be generally included in one of two main approaches: separation-based and pattern recognition-based systems. Separation-based algorithms try to isolate the percussion instruments from the sound mixture by means of different separation techniques, like Independent Subspace Analysis (ISA) or Sparse Coding. The goal is to obtain several streams containing sounds generated by a single percussion source. The next step in the transcription process is to label each stream with the name of the instrument (classification) and then apply an onset detector to get time information about each event within the stream. On the other side the pattern-recognition approach first segments the signal into meaningful events, by e.g. an onset detector, then performs an extraction of features within each event and finally tries to classify them using classification algorithms like SVM, K-NN, GMM or HMM (see figure 2.4).

Other distinctions between algorithms can be made like:

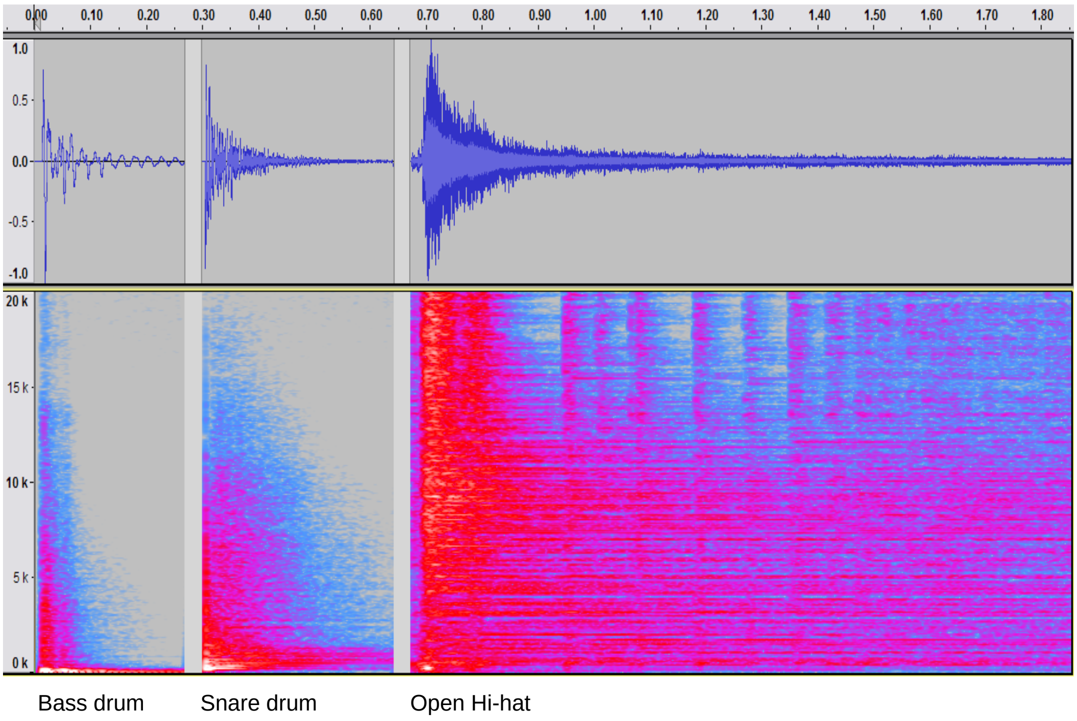


Figure 2.3: Waveform and spectrogram for bass drum, snare drum and open hi-hat.

- supervised (e.g. trained classifiers or instrument templates) vs. unsupervised (e.g. clustering and recognition of clusters).
- low-level signal analysis vs. musicological models.
- general (e.g. instrument templates) vs. local (e.g. using information extracted from the analyzed song).
- by its type of input (e.g. monophonic percussion, polyphonic percussion or polyphonic music).

2.2.1 Pattern Recognition Approaches

A general overview on this approach can be seen in figure 2.5 (adapted from [66]). Regarding sound this approach understands it as a whole and tries to detect "clues" (features) to recognize a particular instrument into the mix; Scheirer [53] calls this

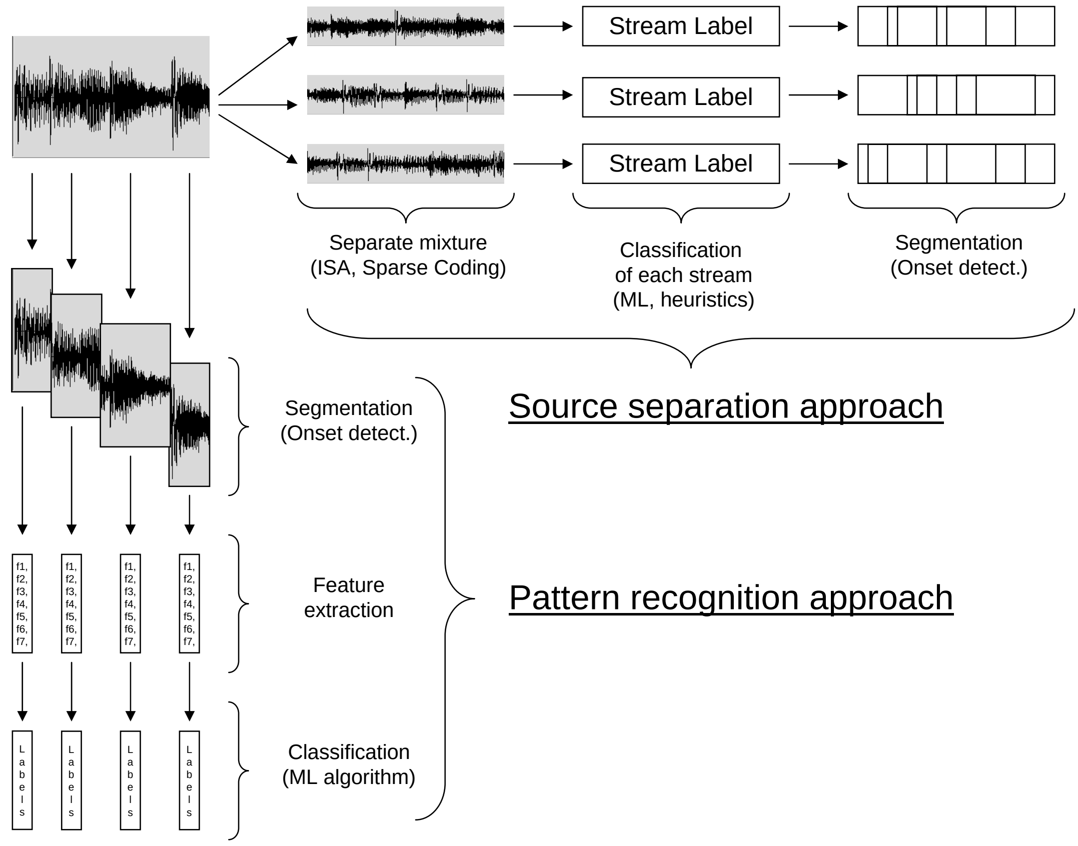


Figure 2.4: Two main approaches in percussion transcription.

approach "understanding without separation". The underline assumption is that the signal processing methods will provide enough information to characterize the aimed sounds even though they are "corrupted" with other concurrent sounds. This approach is also closer to human perception, since we, as human listeners, do not preserve independent waveforms as an intermediate representation of sound [53].

As mentioned above the first step in this kind of approach is the segmentation of the signal into meaningful events. Early works in drum transcription of monophonic drum sounds used information about the onsets, located based on rapid increases on amplitude. Then each event was classified by the system based on sub-band energy features [54]. Onset detectors based on psychoacoustic knowledge like [32] achieve very good results and are commonly used in this field. Other approaches like creating a Tatum grid and defining onsets according to the tick are also used for segmentation proposes [21].

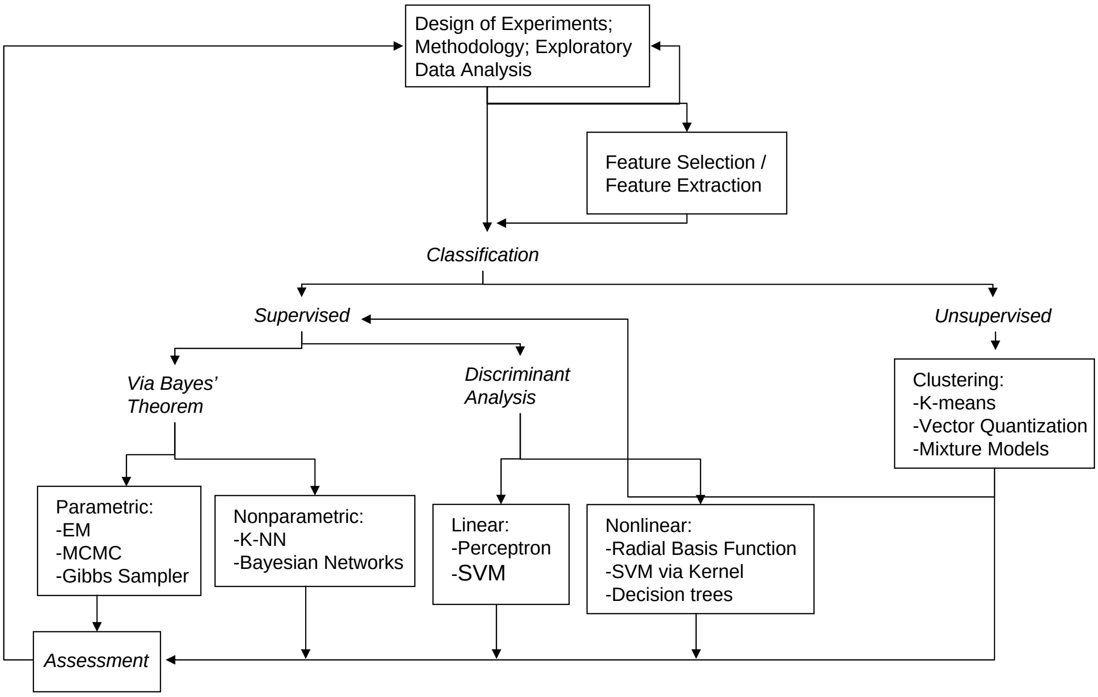


Figure 2.5: The Pattern Recognition Cycle. Adapted from [66].

The next step in a pattern-recognition approach is the extraction of features[2] and its selection. Finally a small set of features is obtained that is hopefully highly correlated with the class they represent and at the same time highly uncorrelated whit the rest of classes [31].

Once the list of features has been proposed and selected (by e.g. a feature selection algorithm) the next step is to train a machine learning algorithm like C4.5 or support vector machine (SVM). In this training process the selected features and the class labels are analyzed to build a model of every class (class modeling). By applying this model (or models) the classification task is finally achieved[3]. See [29,52] as examples on using the C4.5 algorithm for drum classification and [13, 14, 57, 61] as examples of using the SVM algorithm.

The transcription task becomes increasingly difficult as the input signal goes from

[2] See section 2.3 for a comprehensive explanation about audio features

[3] See section 4.1 for a more extensive explanation on feature selection, C4.5 and SVM algorithms

monophonic to polyphonic percussion and polyphonic music, because the level of "noise" (i.e. non-percussive sounds) is significantly higher in the polyphonic case [51].

A large-scale study on automatic classification of monophonic unpitched percussion sounds was presented by Herrera et al. [25] where up to 208 subsets of temporal and spectral descriptors were used and evaluated as features achieving a 15% error rate for more than thirty classes of percussion instruments. Gouyon and Herrera [21] achieved around 80 % of accuracy in transcription of drum tracks with a feature-based approach. Steelant et al. [57] presented the first attempt on using SVM for classification of percussive sounds. By this method they achieve, for isolated and overlapping drum sounds, F-measure[4] values of 0.95 and 0.98 for bass drum and snare drum respectively.

Regarding drum transcription in polyphonic music, Sandvold et al. [52] evaluated 25 CD-quality polyphonic songs using a combination of general and localized (record-specific) sound models. The correct classified instances by the general model (C4.5 with AdaBoost) were manually parsed to be used as localized training set. The average classification accuracy achieved by this system was: 95.06 % for bass drum, 93.1 % for snare and 89.17 for cymbals.

In [61] a whole automatic transcription system was presented. The evaluation of this algorithm was done within the scope of the MIREX 2005 drum detection contest[5]. Starting from the audio file an onset-based classification is preformed using SVM as machine learning algorithm. F-measure results of 0.688, 0.555 and 0.601 were obtained for bass drum, snare drum and hi-hat respectively.

In [14] SVM are again used as classification algorithm for a database of ten songs played by two drummers and mixed four times (80 signals). This algorithm use a band-wise harmonic/noise decomposition as pre-processing step to enhance the presence of unpitched instruments. A localized adapted model like the one presented in [52] was also evaluated. This system achieves an overall F-measure of 0.84 for bass drum and snare drum sounds.

Paulus and Kapuri [42] use hidden Markov models (HMM) with temporal and

[4] See sub-section 4.1.3 for an explanation on the F-measure

[5] See section (6.1) for an overview on the MIREX contest.

spectral features to achieve an overall F-measure of 0.697 for a complete transcription system [6]

In [17] a whole transcription system is presented combining the source-separation and the pattern-recognition approaches. A set of features is computed from the original audio signal and from a “drum enhanced” track obtained by source separation. These feature vectors are then processed by a C-SVM algorithm achieving, for balanced mixtures between drums and music tracks, F-measures of 0.695 for bass drums, 0.583 for snare drums and 0.755 for hi-hats. The experimental database is formed by 28 songs from the ENST_wet database [7].

As can be seen from this review, the pattern recognition approach is well suited for classifying isolated and polyphonic percussion events (i.e. drum samples and drum loops). Regarding automatic transcription of drum events in polyphonic music, this approach has delivered moderate results (no more than 70 %) and there is still a lot of room for improvement. Following the previously described algorithm steps, several areas can be improved in order to obtain better results. Since this is a serial process any improvement on any area would help on increasing the performance of the whole system. From our point of view it would be necessary to work on:

- better training databases: since relatively small databases (no more than 50 songs) were used in the literature to train the systems. It is highly probable that this songs are not enough to represent the variability in terms of timbre and styles (genres) of the entire song universe.

- more efficient event detectors: this would reduce the gap between “pure” classification results and “real world” transcription systems.

- better features: specially designed to extract relevant information from the sound events (e.g. to describe the temporal evolution of descriptor values)

- better machine learning algorithms: up to date best results are obtained by

[6] See section (6.1) for further explanation on this paper.

[7] See section 3.2 for an overview on this database

using the SVM algorithm but it would be interesting to try new algorithms (e.g. to include probabilistic information on the decision process).

In addition, the introduction of high level information (e.g. musicological knowledge) into the system could help to improve its performance. Another important aspect is to work with publicly available databases, like ENST or MAMI, to better compare results among authors. We believe the pattern recognition approach, alone or combined with others, is a powerful tool to describe percussive events in polyphonic music. This believe is supported by previous results on isolated and polyphonic drum events and by perceptual evidence since, as mentioned before, human listeners do not preserve independent waveforms as an intermediate representation of sound [53]. In words of neuroscientist Daniel Levitin [36](pag. 103):

> “The brain extracts basic, low-level features from the music..., using specialized neural networks that decompose the signal into information...It does this through a process of feature extraction, followed by another process of feature integration.”

2.2.2 Separation-based Approaches

This approach has the hypothetical advantage that once each stream has been separated from the mix (the difficult part), the identification process can be easily achieved because high rates of accuracy on identifying percussion instruments from monophonic drum sounds has been reached e.g. in [29]. Generally, in terms of variation of pitch, duration and timbre within a particular song, percussion instruments have more stable behavior than pitched sounds. This characteristic is important for achieving good results with source separation techniques, since these techniques assume stability of spectra.

The most commonly used technique for source separation is Independent Component Analysis (ICA) [5]. In basic ICA the discrimination power is equal to the number of sensors (in this case microphones), not the usual case for commercial music recordings. The number of sensors is normally two in stereo recordings or one

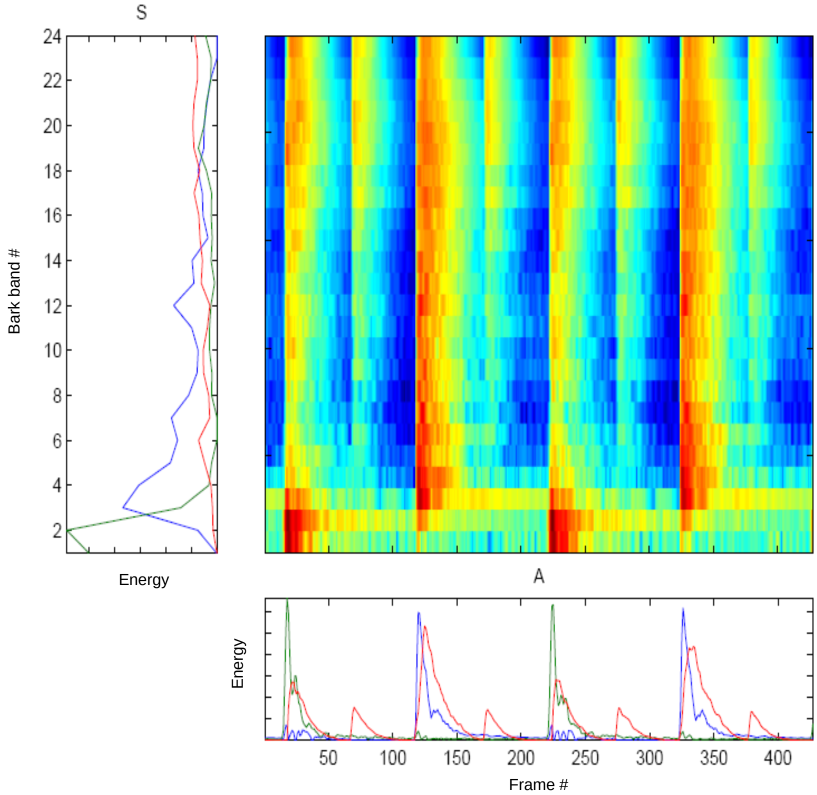


Figure 2.6: Drum loop decomposition into three sources, Bark band frequency resolution. From Paulus poster `http://www.cs.tut.fi/sgn/arg/paulus/eusipco05_paulus_poster.pdf`.

if the “pan” is centered (single-channel) like in most bass drum and snare sounds in studio recordings.

Several methods for source separation from single-channel recordings have been developed and used in drum transcription. These methods are mainly based on the idea that an input signal can be viewed as a sum of sources with fixed spectrum and time-varying gains (see Equation (2.1) and figure (2.6)).

$$X_{\mathrm{t}}(f) \cong \sum_{n=1}^{N} a_{n,t} S_n(f) \tag{2.1}$$

Where X is the power spectrum in frame t, N is the number of sources, n the source index, f the discrete frequency index, a the gain of source n in frame t and S is the fixed spectrum of source n.

Several systems for estimating $a_{n,t}$ and S_n have been developed presupposing different criteria like the independence of sources (ISA), the non-negativity of the matrices (NMF) or the sparseness of sources (Sparse coding).

Independent Subspace Analysis (ISA) [5] can be viewed as a relaxed version of ICA in the sense that it can be implemented with single-channel inputs. The authors propose grouping components by partitioning a matrix of independent component crossentropies (ixegram). The ixegram measures the mutual similarities of components in an audio segment and clustering the ixegram yields the source subspaces and time trajectories.

In Prior Subspace Analysis (PSA) [11] the matrix S (fixed spectrogram) is obtained from previous estimations (training data) then the gain matrix can be calculated by multiplying X by the pseudo-inverse of S. The result is passed through ICA to obtain a set of independent amplitude basis functions. The main problem with PSA is that the calculated matrix may have negative values and that does not happens with real data (see [9,10]).

Non-negative Matrix factorization (NMF) [34,56] is a method based in the assumption of non-negativity of the matrices (a suitable assumption when working with spectrogram). This assumption allows to estimate S and the gain matrix by minimizing a cost function in an iterative algorithm. This method can achieve good results without prior knowledge (blind separation), see [23] as an example on separation of drums and pitched instruments.

A mix between NMF and PSA, called non-negative spectrogram factorization (NSF), was proposed by Paulus and Virtanen [43]. Other combination of techniques is the Non-negative sparse coding (NNSC) [30] that combines NMF and sparse coding.

2.2.3 Other Approaches

Approaches like [68] and [69] use matching template spectrogram achieving very good results (e.g. the Adamast system implemented with this approach was the winner system in the MIREX contest of drum transcription from polyphonic music). The main idea is to obtain, from a large training database of sounds, a template spectrogram representation of a particular percussion instrument. Thus, when analyzing a song, a template-adaptation algorithm is applied on every onset. A distance measure for template matching is used to avoid the spectral overlapping of other sounds. The evaluation results in the Audio Drum Detection Contest were 72.8%, 70.2%, and 57.4% in transcribing bass drums, snare drums, and hi-hats, respectively. Dittmar [6] combines source-separation (Non-Negative ICA) with spectrogram templates to achieve, also within the MIREX contest, F-measure values of 0.606, 0.581 and 0.585 for bass drum, snare drum and hi-hat.

In table 2.1 a summary of the previously seen works on drum transcription in polyphonic music is shown. As we can see from the presented evidences, the problem of drum transcription is not a solved one, and there is a lot of room for improvement. Most of the work has been focused either on the separation-based approach or the patter-recognition approach although other approaches like matching template algorithms may also achieve good results.

The main problem with the separation-based approach is that source separation techniques are still far from reliable extraction of individual sources in most of the "real-world" sounds. Since the rest of the process depends on this first separation step, taking a second approach give the impression to be a better way to deal with the problem in the meanwhile.

Regarding the pattern-recognition approach achieved results are still far from optimal transcription (between 60 and 69 %). As seen before, there are still many things that can be done to improve the classification results like: using more representative training databases (this can be achieved by working with sample songs with representative variability on timbre and genre), improving event detectors, extracting better features and working with better machine learning algorithms.

Other approaches like matching templates achieve similar results as pattern-recognition. It could be also interesting to combine different strategies in order to get an expert system.

In the present thesis we will follow the pattern-recognition approach to evaluate a large set of sounds and new descriptors to determine its potential to describe percussion events in polyphonic music.

2.3 Commonly used descriptors

In order to analyze audio with data mining algorithms we first need to extract relevant features from the signal. Ideally these features will describe the sound by producing a downsampled collection of multivariate time series [38]. Starting from the time-domain representation of the audio we can either directly compute "temporal" descriptors or first transform the signal, e.g. into frequency, Mel or Bark representations, and then compute descriptors from this new representation. It is also a common practice to aggregate this "low-level" descriptor into "mid-level" representations of the sound (e.g. MFCC's, Bark band ratios, etc.) Many different sound descriptors have been proposed in the literature but there is still no consensus on the most relevant ones for discriminating unpitched sounds in polyphonic music [17]. In this thesis we compute a set of thirty-two descriptors that can be roughly grouped into four categories namely:

- Temporal descriptors: computed from the time-domain representation of the audio
- Spectral descriptors: computed from the frequency representation of the signal obtained from the Short-Time Fourier Transform (STFT)
- Perceptual descriptors: computed from perceptual representation of the signal like Mel and Bark scales.
- Tonal descriptors: mid-level representation of the tonal content.

Next an overview on the used descriptors is presented (see [44] , [25] , [51] and the MPEG-7 standard on audio descriptors[8] for a more comprehensive explanation).

[8] see web page: `http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm`

Authors	# songs	Approach	Main Algorithms	Overall result	bass drum	snare	hi-hat	Measure
Sandvold et al. [52]	25	Patt-Rec	local. models	92.44	95.06	93.10	89.17	Accuracy
Tanghe et al. [61]	50	Patt-Rec	SVM	61.47	68.80	55.50	60.10	F-measure
Gillet and Richard [14]	20	Patt-Rec	SVM+local.	84.00	82.30	84.20	—	F-measure
Paulus and Klapuri [42]	45	Patt-Rec	HMM	69.70	79.50	65.50	66.00	F-measure
Gillet and Richard [17]	28	S. Sep+Patt-Rec	C-SVM	67.80	69.50	58.30	75.50	F-measure
Helen and Vitranen [23]	100	S.Sep	NMF	93.00	—	—	—	Correct class
Dittmar [6]	50	S. Sep+Sp. Temp	NN ICA	58.80	60.60	58.10	58.50	F-measure
Yoshii et al. [68]	50	Sp. Temp	Temp. match.	67.00	72.80	70.20	57.40	F-measure
Yoshii et al. [69]	70	Sp. Temp	Temp. match.	62.48	82.92	58.28	46.25	F-measure

Table 2.1: Summary of drum transcription systems in polyphonic music. Almost all works classify bass drum, snare drum and hi-hat sounds, except for Helen and Vitranen [23] (where only discrimination between drum and pitched sounds is performed) and Gillet and Richard [14] (where only bass drum and snare drum are detected). Patt-Rec means Pattern recognition, S. Sep. means source separation and Sp. Temp means spectral template.

Temporal descriptors:

- **zero-crossing rate (ZCR):** it counts how many times a signal changes from positive to negative values (cross the zero axis).
- **lpc:** the values for the eleven linear predictive coefficients of the signal.

Spectral descriptors:

- **spectral centroid:** is the center of gravity of the spectrum.
- **spectral complexity:** it measures the complexity of the instrumentation of the audio piece. Normally, a big number of musical instruments increases the complexity of the spectrum (see [58]).
- **spectral crest:** it measures the noisiness of the spectrum by computing the ratio between the max value and the arithmetic mean of the spectrum.
- **spectral decrease:** it is a measure of the amount of decrease of the spectral amplitude.
- **spectral dissonance:** it measures the roughness of the sound by means of dissonance curves obtained from perceptual experiments (see [45]).
- **spectral energy:** It is the total spectrum energy at a given frame.
- **spectral energyband low:** the spectrum energy between 20 and 150 Hz.
- **spectral energyband middle-low:** the spectrum energy between 150 and 800 Hz.
- **spectral energyband middle-high:** the spectrum energy between 800 and 4.000 Hz.
- **spectral energyband high:** the spectrum energy between 4.000 and 20.000 Hz.

- **spectral flatness db:** it characterizes the shape of the spectral envelope. It is computed by the ratio between the geometric mean and the arithmetic mean per frequency bands (in our case bark bands). For tonal signals flatness db is close to one and for noisy signals is close to zero .
- **spectral flux:** it is a measure of how quickly the power spectrum changes from frame to frame. It is obtained by comparing the power spectrum of one frame against the power spectrum of the previous one.
- **spectral high frequency content (hfc):** it is a measure of the amount of high frequency content of a signal. It is computed by adding the magnitudes of the spectral bins, but multiplying each magnitude by its own position value (proportional to the frequency). See [40]
- **spectral kurtosis:** it gives a measure of the flatness of a distribution around its mean value. It is computed like the fourth order statistical moment, but taking the spectrum as the histogram of the signal.
- **spectral pitch:** it is represented as the fundamental frequency of the analyzed sound. It works for monophonic sounds and it is computed using the YinFFT method developed by Paul Brossier (see [4]).
- **spectral pitch instantaneous confidence:** it is a measure of pitch confidence derived from the YinFFT algorithm [4]. It gives an idea about how much a certain pitch is affecting the total spectrum. An output near to one means that exist just one pitch in the mixture, an output near to zero indicates multiple or not distinguishable pitches.
- **spectral pitch salience:** It is given by the ratio of the highest peak to the zero-lag peak in the autocorrelation function. Ideally non-pitched sounds have a mean pitch salience value close to zero while harmonic sounds have a value close to one (see [47]).
- **spectral rms:** the root mean square spectrum energy.

- **spectral rolloff:** it is the frequency value who left 95% of the spectrum energy below its value.

- **spectral skewness:** it is a measure of the asymmetry of a distribution around its mean value. A negative skewness indicates a signal spectrum with more energy in the high frequencies. A positive skewness indicates a signal spectrum with more energy in the low frequencies. A skewness equal to zero indicates a symmetric spectrum. For silence or constants signal, skewness it is also zero. It is computed like the third standardized moment but assuming the spectral values as the histogram representation.

- **spectral spread:** it is defined as the variance of a distribution around its mean value. It is equal to the second order central moment being, for this case, the spectral centroid the first central moment.

- **spectral strongpeak:** It is defined as the ratio between the spectrum maximum magnitude and the bandwidth of the maximum peak in the spectrum above a threshold (half its amplitude). It reveals whether the spectrum presents a very pronounced maximum peak (see [21]).

Perceptual descriptors:

- **barkbands:** It is a vector containing 27 Bark band values of a Spectrum. For each Bark band the power-spectrum (magnitude-squared) is summed. For better resolution the first two bands [0..100] and [100..200] are divided in two (according to [25]). The Frequency edges (in Hz) are: 0, 50, 100, 150, 200, 300, 400, 510, 630, 770, 920, 1080, 1270, 1480, 1720, 2000, 2320, 2700, 3150, 3700, 4400, 5300, 6400, 7700, 9500, 12000, 15500 and 20500.

- **barkbands ratio:** taking the sum of the Bark band vector (27 values) on each frame as 100%, each Bark band value is normalized to determine its contribution to the total.

- **barkbands kurtosis:** it is computed like the spectral kurtosis but taking the Bark band values instead of the spectral ones.

- **barkbands skewness:** it is computed like the spectral skewness but taking the Bark band values instead of the spectral ones.

- **barkbands spread:** it is computed like the spectral spread but taking the Bark band values instead of the spectral ones.

- **MFCC:** the standard 13 Mel-Frequency Cepstrum Coefficients

- **MFCC ratio:** taking the sum of the MFCC coefficients (13 values) for each frame as 100%, each MFCC coefficient is normalized to determine its contribution to the total.

Tonal descriptors:

- **Harmonic Pitch Class Profile (HPCP):** it is a 36 dimensional vector which represents the intensities of each of the frequency bins of an equal-tempered scale (see [18]).

2.4 The "Ceiling Glass" problem

Comparing the obtained results on classification of monophonic sounds with those obtained for polyphonic sounds, it is clear that results for the latter are worst most of the time. What could explain this low performance in polyphonic audio?

Recent studies have detected some very interesting issues that appear to be the cause for this low performance namely:

- Problems derived from the "bag-of-frames" (BOF) approach.

- The "Ceiling Factor".

- The need for better features.

The classic approach in pattern recognition algorithms, as described earlier, is to cut an audio waveform into frames (usually 50 ms. with 50 % overlap [2]) and then to compute a big number of "general" low-level features within this frame (as seen in section 2.3). After, each sound event is described by a collection of frame-wise features [35] and these feature vectors are delivered to a machine learning algorithm (like SVM, K-NN, etc.) to perform the classification or clustering task. This approach is usually called the bag-of-frames (BOF) since the temporal relationship between frames is lost within each sound event.

The BOF approach has proved to be effective in classification and clustering of isolated sound events and soundscapes [2]. However in the case of polyphonic audio the time cues within sound events seem to be an important aspect in detecting such events but these time cues are vanished in the BOF approach [35].

This classic approach (BOF) also seems to have reached a performance ceiling since the fact of increasing the complexity of the classification algorithms does not improve significantly their performance [3].

An additional problem suggested in the literature in close relationship with the BOF problem is the requirement for better features that take into account the temporal evolution of the sound events[9] [26] [47] [8].In [41] the concept of "analytical feature" is introduced where features are built for each specific problem using an heuristic function generation process.

[9] In some cases the first and second derivative between frames are computed, but these are still a short-time measures that do not account for mid-level changes on the temporal evolution of descriptors.

Chapter 3

Research problems, proposed experiments and units of study

In the previous chapters we present an overview on the problem of drum transcription. As can be seen, from the state of the art section, this is not a solved task and we are still far from perfect transcriptions. In chapter 1 we introduce, following the ideas presented in [28], that score-like transcription is just one area of the automatic "description" process. In this thesis we will address this "descriptionist" path since, to our understanding, very useful information could be derived from "imperfect" transcriptions (e.g. describing the "percussivity" of a song, the type of drum instrument played, etc.). In section 2.2 we see that the pattern recognition approach is one of the better suited approaches to derive these percussion-level descriptors. In this chapter we present some research problems derived from this approach and we also plan some experiments in order to move towards their solution.

3.1 Research problems

As seen from the state of the art on drum transcription in polyphonic music we can detect three main research problems:

- The need for studies evaluated on larger databases as done by Herrera et al. [25] for monophonic percussion sounds. See subsection 2.2.1 for an explanation on this topic.

- The need for mid-term temporal-evolution descriptors in order to gain information about the evolution of the frame-level descriptors going beyond the classical bag-of-frames (BOF) approach. As seen in section 2.4 this gained information could help in “breaking” the “glass ceiling” observed after using traditional BOF descriptors.

- The need for evaluate the descriptive capabilities of a system working with imperfect detection levels. To take a “descriptionist” approach instead of a “transcriptionist” and try to derive song-level useful information from these imperfect descriptions. In other words: can we derive useful information from this imperfect transcriptions, or do we have to wait for more accurate results?

3.1.1 Proposed Experiments

In order to answer the previously detected research problems we propose four sets of experiments to be evaluated in the following chapters:

- **Set 1 - Database selection and normalization** (chapter 4): in these experiments we will try to obtain a representative database of drum events and a common normalization criteria. In order to do that we will first evaluate three annotated-percussion databases to choose the one with best database-crossvalidation results. Then, in order to gain confidence about the representativeness of the selected database, we will compare classification results by using different normalization strategies. We expect, for a representative database, that normalization parameters extracted from this database have comparable ranges with those obtained from much bigger sets.

- **Set 2 - Object-level temporal evolution descriptors** (chapter 5): in these experiments we will compute and evaluate the performance of object-level temporal evolution descriptors. We will try to determine if by adding temporal information, that represent the time evolution of frame-level descriptor values, the classification results are increased.

- **Set 3 - Drum transcription system** (chapter 6): in these experiments we will determine the transcription capabilities of the previously trained models against a large number of polyphonic songs. We will evaluate "standard" transcription results (i.e. what and when a percussive event is produced) and we will also evaluate "relaxed" transcription results (i.e. what and how many percussive events within a song are). The results obtained from these experiments will lead us to propose several mid-level percussion descriptors.

- **Set 4** - **Mid-level percussion descriptors** (chapter 7): in these experiments we will compute and evaluate the description capabilities of mid-level percussion descriptors extracted from the output of the previously evaluated drum transcription system. We will try to determine if these new descriptors could help in other MIR areas such as genre classification or danceability estimation.

3.2 Units of study

Looking for collections of songs with proper annotations on percussive events we found three databases that suit our needs. Two of them are publicly available and were used in previous studies on drum transcription; the other one is an in-house annotated collection of songs.

Next we describe the main characteristics of each database.

- **ENST-Drums database:** This is the largest publicly available drum database (for a detailed overview on this database see [15]). It contains recordings from three different drummers and drum sets playing single hits, drum phrases and complete songs covering various styles. It was carefully annotated with the assistance of video recordings. The authors provide "dry" (without sound effects) and "wet" drum recordings tracks on one side and its music accompaniments on the other. In our case, since we want to detect drum events in "real world" music, we work only with "wet" sounds. Given that the drum and music tracks are provided in separate files our mixing strategy is to merge them directly (without further changes of sound levels). Afterward we segment 30 seconds of each song, and its labels, to be used as working collection. At the end we obtain 28 songs with real accompaniments and 36 songs with MIDI (provided in audio format) accompaniments adding up a total of 64 songs. We call this database “ENST_wet”.

- **MAMI database:** This database is a collection of 52 music fragments extracted from commercial CD’s. Each 30 seconds long fragment was annotated by Tanghe et al. (see [62]) within the context of the Musical Audio Mining (MAMI) project using MIDI files to label drum events. Since only the MIDI annotations and a list of the songs is provided, we purchase the audio files. We manage to gather 48 songs, we align them with the MIDI files and then we transform each MIDI event into a text label. This database is one of the three databases used during the MIREX 2005 drum contest.

- **Sandvold’s database:** This is an in-house database of 30 annotated 20-second audio excerpts, extracted from commercial CD’s (see [51]). It was created for evaluation purposes on drum transcription systems and it has four annotated categories: bass drum, snare, cymbals (including hi-hats) and miscellaneous percussion.

As can be seen in figure 3.1 genre distribution within each database is more or less broad (i.e. songs are not particularly focused on one genre). A broad genre distribution is important to get a representative training database.

In order to compatibilize the three databases we need to "translate" their labels into a common label dictionary (see appendix A.1 for more information) . The number of instances per database can be seen in table 3.1

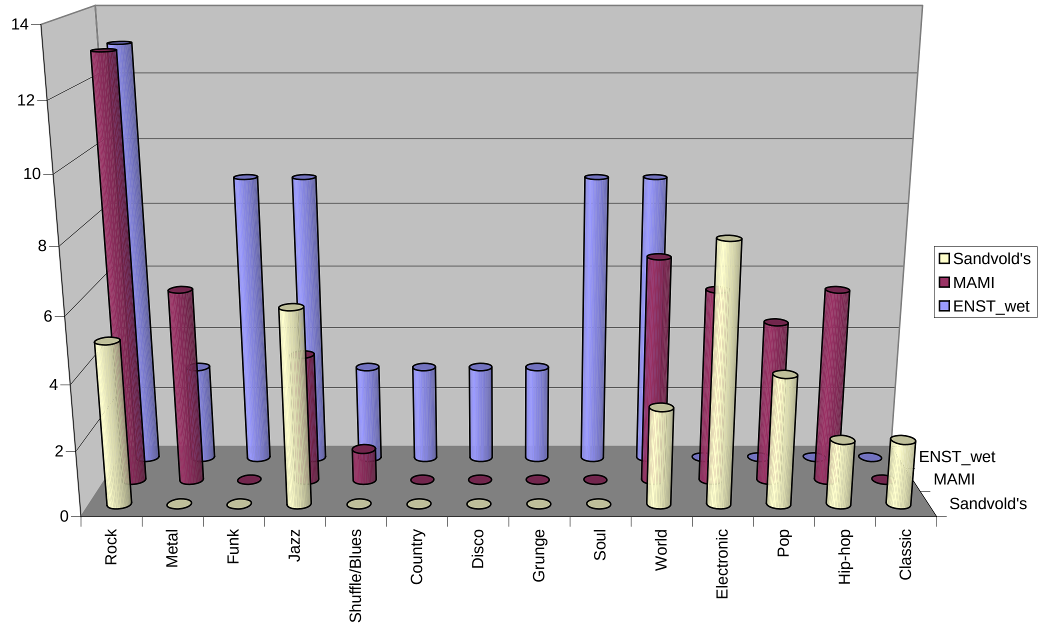


Figure 3.1: Genre distribution per database, y axis represents the number of songs.

Instrument	ENST_Wet	MAMI	Sandvold's	Total
bass drum	3,771	2,149	487	**6,407**
snare	3,763	1,459	433	**5,655**
hi-hat (closed)	3,790	2,487	0	**6,277**
hi-hat (open)	1,444	679	0	**2,123**
ride cymbal	1,169	518	0	**1,687**
crash cymbal	127	166	0	**293**
splash cymbal	0	40	0	**40**
other cymbals	431	0	618	**1,049**
tom toms	353	204	0	**557**
rim shot	20	101	0	**121**
cross stick	196	0	0	**196**
cowbell	84	75	0	**159**
other drum	0	1,716	1,026	**2,742**
total	**15,148**	**9,594**	**2,564**	**27,306**

Table 3.1: Number of drum instances per database.

Due to the number of instances and the musical importance of each instrument within the drum kit we decide to work with the following instruments: bass drum, snare drum, hi-hat (open and close) and cymbals (including hi-hat).

At the end of this process we obtain a large set of polyphonic audio excerpts of

30 seconds length (mono, 16 bit, 44100 Hz). These audio segments coming from the three databases add up a total of 142 songs labeled with four (possibly concurrent) tags namely: bass drum, snare, hi-hat and cymbals. The final number of instances can be seen in table 3.2

Instrument	**ENST_Wet**	**MAMI**	**Sandvold's**	**Total**
bass drum	3,771	2,149	487	**6,407**
snare drum	3,763	1,459	433	**5,655**
hi-hat	5,234	3,166	0	**8,400**
cymbals	6,961	3,890	618	**11,469**

Table 3.2: Final number of instances per database.

Chapter 4

Database selection and normalization

In this chapter we evaluate the capabilities of the previously described databases to be used as training databases for machine learning algorithms.

We also perform a set of experiments to determine the best normalization criteria to be applied on training and testing databases.

4.1 Background

4.1.1 Classification techniques

C4.5

The C4.5 [46] algorithm belongs to the "binary trees" classification family which generates classifiers expressed as decision trees. These trees are constructed top-down using the concept of information gain. Starting with the presumable most informative feature (the one that maximally reduces entropy), branches are created from the different values of this descriptor. Then the training examples are sorted to the appropriate descendant node. The whole process is recursively repeated with all the descendant nodes. Once built the tree can be pruned to avoid overfitting [24]. The attributes can be either numerical or nominal and it can handle multi-class classifications.

These algorithms are fast, robust to noisy data and their outputs can be easily summarized into interpretable trees of descriptors. This characteristic is very important when we want to "learn" about the descriptors involved in the classification process.

Support vector machines

Support Vector Machines (SVM) [64] are based on statistical learning theory. The SVM algorithm finds the hyperplane with maximum soft-margin for the given training set [27]. Once the separating hyperplane $f(x)$ is found, new data instances x_n can be easily classified by evaluating the sign of the function $f(x)$. If $f(x_n) > 0$ then x_n belongs to the positive class, otherwise it belongs to the negative class [67]. If the training data is not linearly separable it can be mapped it into a new (hopefully linear) hyperspace. This can be done by choosing the correct kernel function. Common kernel functions are polynomial (of various degrees) and radial basis functions (rbf). Support vector machines are better suited for binary classification tasks but multi-class SVM can also be found. Nowadays support vector machines are considered a must try on any machine learning application due to their robustness and accurate results [67].

Within this thesis, when dealing with SVM we will use the LibSVM[1] implementation found in the Rapid miner[2] software.

4.1.2 Feature selection

When the number of features is too high the whole classification process can be damaged. Hence it is advisable to perform an algorithm-independent feature selection process to detect the most informative features avoiding, at the same time, the overfitting of the training data. This kind of feature selection algorithms are also called "filters" as opposite to "wrappers" that use the learning algorithm itself to evaluate the appropriateness of features.

Next we describe two feature selection algorithms used in this thesis:

[1] See: LibSVM web site `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

[2] Rapid miner is an open source data mining software, see rapid-i web site `http://rapid-i.com/content/view/10/32/lang,en/`

- **Correlation-based Feature Selection (CFS):** this algorithm evaluates subsets of attributes by computing the ratio between how predictive is the selected group and how intercorrelated are their attributes. This ratio gives a heuristic of the "merit" of the subset [22]. The best subsets are those with low intracorrelation and, at the same time, high class-correlation (correlated with the class attribute).

- **ReliefF:** is a feature weighting algorithm[3] . Every feature weight is iteratively estimated according to its capability to discriminate between neighboring patterns. On every iteration, a random pattern is selected and then two nearest neighbors are found, one from the same class (called NH) and the other from the other class (called NM) [60]. Hence this algorithm score individual features rather than feature subsets as the CFS algorithm.

Herrera et al. [29], within the scope of isolated percussion classification, have found better results on using ReliefF as feature selection algorithm for instance-based classifiers and CFS for decision-trees algorithms.

4.1.3 Evaluation

There are many evaluation measures in the information retrieval field, but three of the most commonly used are: precision, recall and the F-measure.

- **Precision:** is the fraction of retrieved items that are correct.

$$P = \frac{C}{D} \tag{4.1}$$

where C is the number of correct results and D is the number of detected results

- **Recall:** is the fraction of items that are correctly retrieved.

$$R = \frac{C}{GT} \tag{4.2}$$

where C is the number of correct results and GT is the total number events (ground truth).

[3] A real-valued number is assigned to each feature to indicate its relevance

- **F-measure:** is the weighted harmonic mean of precision and recall, it summarize a trade-off between both values[4].

$$F - measure = \frac{2P.R}{P + R} \tag{4.3}$$

Following the evaluation criteria decided by the MIR community for the MIREX 2005 drum contest we decide to evaluate all our experiments using the F-measure. This measure it is also used as evaluation criteria in [17], [61], [57] and [14].

4.2 Methodology

Since we want to detect and describe percussive sounds in polyphonic music using a pattern-recognition approach, we need to start from the best possible database to be used as a training set for the system. It is also recommended to have a second database the sounds of which have been excluded from the training process to be used as an independent test.

Another important decision, when working with machine learning algorithms dealing with more than two classes, is whether to work with one multi-class model or several binary-models. In our case, since we are working with percussive events that can occur at the same time, we decide to train n binary classifiers instead of one model with 2^n possible classes (being n the number of instruments to detect). In this context we will have one trained model in charge of detecting the presence, or absence, of one particular instrument (e.g. snare or not-snare). Although the process of training several models is more expensive, in terms of time and computational cost, by taking this path we avoid building "mixed" classes for all possible co-occurring combinations of instruments like e.g. snare+hi-hat, bass drum+snare, etc (see [13] for comparison of both strategies in classifying drum loops).

[4] This measure is also known as balanced F-measure or $F_1 - measure$ since the genreal F-measure formula is $F_\eta = (1 + \eta^2)(P.R)/(\eta^2.P + R)$ for all $\eta > 0$

4.2.1 Database selection

As seen in section 3.2 we have three different candidates to be our training database. Due to its size (30 songs) we decide to left Sandvold's database as independent testing database. The training capabilities of ENST_wet, MAMI are evaluated. A mix database is also constructed and evaluated (ENST_wet+MAMI). In this case we split the database using 90% to train and reserving 10 % of the database to be used as testing set.

The aim of this set of experiments is to determine which of the databases is better to work with as training set.

The general framework of the experiment is as follows:

1. Frame based feature extraction: temporal, spectral and perceptual descriptors[5] are computed by analyzing a windowed signal of 2048 frames with a hop size of 512 frames. For the tonal descriptors we increase the window size up to 4096 frames and the hop size up to 1024. The whole song is processed to store all the frame-level descriptor values into a file. All these frame-level descriptors are computed using Essentia[6]

2. Onset detection using an in-house implementation of the onset detector algorithm proposed by Brossier in [4].

3. The mean, variance, minimum and maximum values for each descriptor is computed for every onset plus 150 milliseconds (12 frames) or until the next onset if it is located before 150 milliseconds. At the end we get a feature vector of 612 values for each onset.

4. Every onset is labeled according to the following strategy: every label that is within a threshold of ±30 milliseconds from the onset value is assigned to this onset (this strategy generates multiple-labels per onset).

[5] See section 2.3 for further information.

[6] Essentia is an audio analysis and music matching tool developed by the music technology group (MTG), see essentia's web site: `http://mtg.upf.edu/technologies/essentia` for further information.

5. Four databases are constructed from the multi-labeled onsets (one per instrument) containing one binary labeled (e.g. snare, non-snare) feature vector per onset.

6. These steps are repeated for every database (ENST_wet, MAMI and Sandvold's[7]).

7. Several combinations of training and testing sets are evaluated using the C4.5 decision tree algorithm implemented in Weka[8] setting the "minimum number of objects" parameter to 5% of the number of instances of each database to avoid extremely large trees. All training models are computed with 10-fold cross-validation[9]

8. The database with better classification results is selected to be the training set.

4.2.2 Normalization

Taking the selected database from the first set of experiments (subsection 4.2.1) we use a state-of-the-art algorithm to evaluate its potential as training set. We understand "potential" as the possibility to derive, from the training database, models that perfom well with "real world" songs. In order to do that an "ideal" training database has to be a manageable sub-set of songs representative of the song universe, i.e. allowing extrapolations from the training database to the population (song universe). As mentioned in section 2.2 of this thesis the best results are obtained by using support vector machines (SVM). The only inconvenience with this (an other) machine learning algorithm is that it requires normalized input values (all its values between 0 and 1).

[7] see 3.2 for an explanation on these databases

[8] Weka is an open source software for machine learning see: `http://www.cs.waikato.ac.nz/ml/weka/`

[9] In the 10-fold cross-validation process the whole database is split into 10 equal sub-sets. Then 9 of these sets are used for training purposes while the remaining one is used for evaluation. This training-testing process is preformed 10 times using the 10 sets as a testing set.

In order to normalize a set of values we first need to determine its possible maximum and minimum values. A commonly used practice is to extract these parameters from the training database and then apply these normalization coefficients also to the test database. In this case we are assuming that the training database is a good representation of the universe of possible songs (and values), a desirable, but not always correct assumption.

In this section we want to determine if a general normalization criteria can be applied to avoid database-specific normalizations and to evaluate the representativeness of our training database.

To derive "general normalization" parameters we estimate the maximum and minimum values of the descriptors (section 2.3) from an in-house collection of more than 5,000 songs. To determine these values we first compute the mean and standard deviation for each descriptor values. Then to estimate the minimum we subtract two times the standard deviation to the mean value and add two times the standard deviation to the mean to derive the maximum normalization parameter (see equations 4.4 and 4.5). We avoid computing this for the descriptors which values were already between zero and one.

$$Min = mean - 2StdDev \tag{4.4}$$

$$Max = mean + 2StdDev \tag{4.5}$$

As in subsection 4.2.1 we plan a set of experiments to evaluate the classification performance of the database. In this case we first perform a CFS feature selection (see subsection 4.1.2 for an explanation on this algorithm) in a 10-fold manner keeping only the features selected on more than 5 folds[10]. We use SVM as learning algorithm, 10-fold validation for model construction and database cross-validation for testing purposes. Three evaluations are performed on each instrument:

- Each database is normalized by its own maximum and minimum values (self

[10] After the feature selection we kept 42 attributes for bass drum, 71 for snare, 31 for hi-hat and 27 for cymbal.

normalization).

- The normalization coefficients are extracted from the training database and then applied also to the test database (train normalization).
- The parameters extracted form the general database are used to normalize both train and test databases (general normalization).

A standard set of parameter for the SVM algorithm is applied in all the experiments (C = 10, gamma = 0.2 and rbf kernel)

4.3 Results and Discussion

4.3.1 Results

Database selection results

In spite of the reasonable good results obtained for both databases in the 10-fold validation scenario, the database cross-validation results are not as consistent as in the previous scenario. When the ENST_wet database is used as training set good results are obtained for bass drum and snare but not for hi-hat and cymbals in the remaining databases (database cross-validation). In the case of training with MAMI good results are achieved for bass drum and hi-hat but not for snare and cymbals (see table 4.1 and figure 4.1).

C4.5		**Tested on**		
Trained on	**Instrument**	**ENST_wet**	**MAMI**	**Sandvold's**
ENST_wet	bass drum	0.781	0.718	0.696
MAMI	bass drum	0.775	0.782	0.821
ENST_wet	snare	0.718	0.688	0.633
MAMI	snare	0.467	0.760	0.555
ENST_wet	hi-hat	0.712	0.436	—
MAMI	hi-hat	0.624	0.653	—
ENST_wet	cymbals	0.704	0.342	0.552
MAMI	cymbals	0.505	0.620	0.456

Table 4.1: ENST_wet vs. MAMI. Classification results per instrument using each database as training set. Results for 10-fold self-validation and database cross-validation (F-measure values).

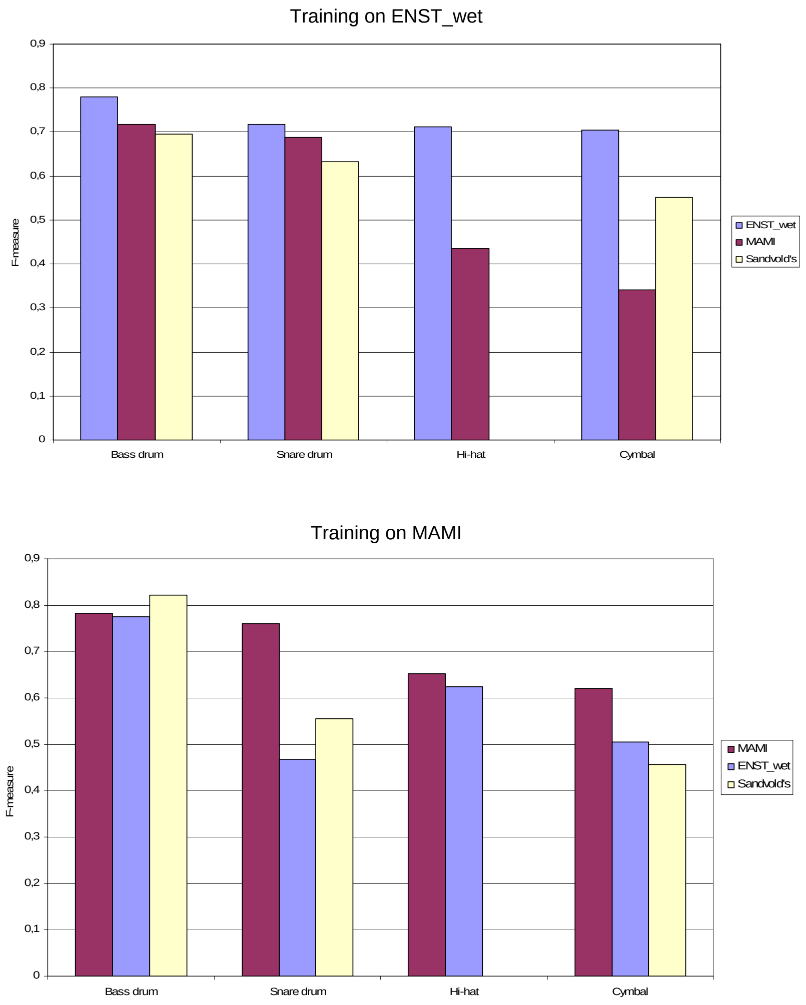


Figure 4.1: Cross-validation results per instrument. Above: training on ENST_wet. Below: Training on MAMI.

In order to avoid these fluctuating results we decide to mix both databases with the intention of increase the representativeness of hi-hat, snare and cymbals examples. As mentioned before we split the mixed database and use 90% of it to train and 10 % to test. Within the training set we train the model by 10-fold cross-validation.

The obtained results for this new database are more homogeneous and in all cases overpass the 60 % value for the F-measure in the database cross-validation (see table 4.2). This new database contains more than 14,200 instances (see appendix

C4.5		F-measure				
Trained on	**Instrument**	**90%MIX**	**ENST_wet**	**MAMI**	**Sandvold's**	**10%MIX**
90%MIX	bass drum	0.779	0.781	0.776	0.786	0.751
90%MIX	snare	0.718	0.712	0.786	0.611	0.712
90%MIX	hi-hat	0.684	0.726	0.656	—	0.711
90%MIX	cymbals	0.674	0.719	0.660	0.641	0.692

Table 4.2: Mixed database evaluation.

A.2 for details) and it seems to be the best option to be used as training set for the system. To be more confident with these results the same experiment is performed five times with random splits (90-10) of the mixed database. The box-plot of these experiments can be seen in figure 4.2. The maximum distance between the lower and upper quartile is observed in cymbal results, being this distance less than 8 %. This means that F-measure classification results are more or less independent from the 90-10 split point.

The average number of descriptors selected by the decision tree algorithm is 5 for bass drum, 8 for snare drum and 9 for hi-hat and cymbals. The most frequently selected features per instrument are:

- **Bass drum:** low Bark bands, MFCC, spectral energy low and spectral flux.
- **Snare drum:** mid Bark bands , temporal lpc, MFCC and spectral flatness.
- **Hi-hat:** high Bark bands, temporal lpc, MFCC, spectral spread and spectral flatness db.
- **Cymbals:** high Bark bands, temporal lpc, MFCC, spectral spread and spectral kurtosis.

Normalization results

The results for the normalization experiments can be seen in table 4.3 and show almost no differences between normalization strategies (except for the bass drum performance in “self” normalization which performance is clearly damaged by this approach). The small differences between “general” and ”training” normalization criteria strongly support our assumptions about the representativeness of the “90MIX”

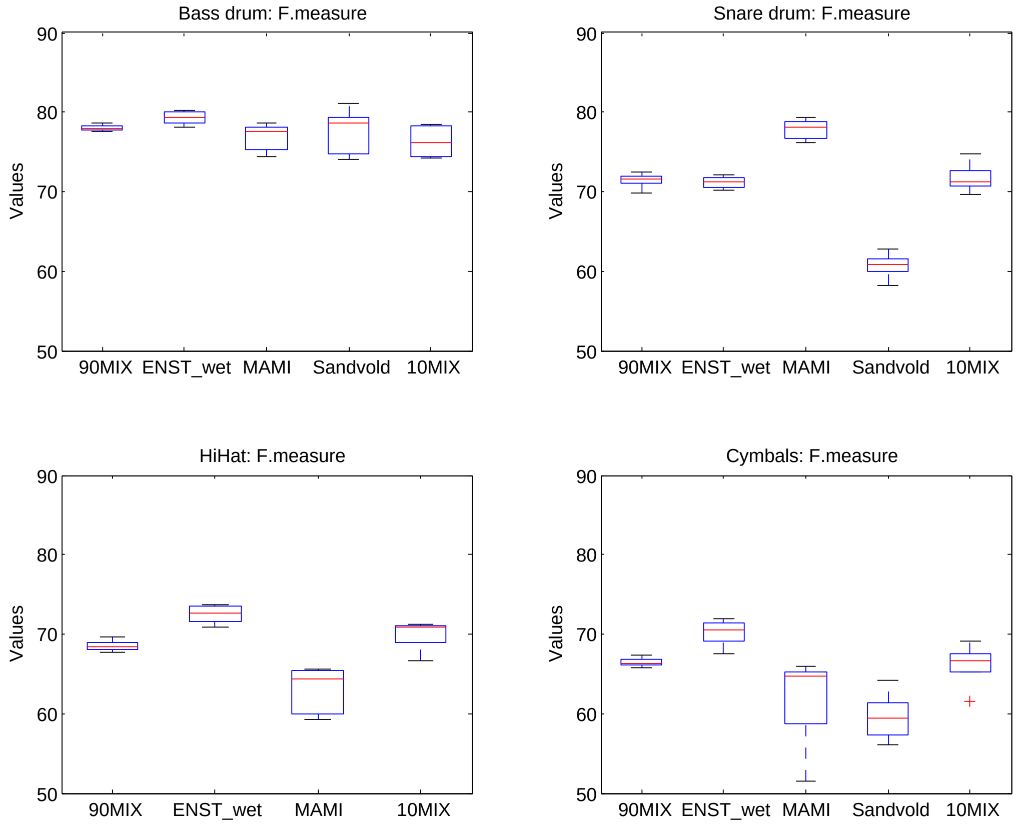


Figure 4.2: Five trials box-plot: F-measure classification results for the 5 random splits (90-10) per instrument. The red line corresponds to the median, the blue box represents the inter-quartile interval (1^{st} to 3^{rd} quartile), the black lines correspond to the maximum and minimum values (assuming no outliers) and the red + stands for outliers.

LibSVM			F-measure		
Instrument	**Trained on**	**Normalization**	**90%MIX**	**Sandvold**	**10%MIX**
bass drum	90MIX_self_norm	self	0.822	0.707	0.769
bass drum	90MIX_self_norm	training	0.822	0.806	0.781
bass drum	90MIX_general_norm	general	0.804	0.821	0.793
snare	90MIX_self_norm	self	0.791	0.598	0.743
snare	90MIX_self_norm	training	0.791	0.608	0.791
snare	90MIX_general_norm	general	0.786	0.579	0.784
hi-hat	90MIX_self_norm	self	0.733	—	0.734
hi-hat	90MIX_self_norm	training	0.733	—	0.725
hi-hat	90MIX_general_norm	general	0.730	—	0.737
cymbals	90MIX_self_norm	self	0.705	0.641	0.678
cymbals	90MIX_self_norm	training	0.705	0.641	0.682
cymbals	90MIX_general_norm	general	0.700	0.676	0.682

Table 4.3: Normalization evaluation. (SVM same parameters).

database. Because the normalization parameters extracted from this database (about 130 songs) are compatible (i.e. they can be interchanged without degrade the classification results) with the parameters extracted from the "general" database (a set of more than 5,000 songs).

4.3.2 Main Conclusions

At the end of these experiments we can conclude that the 90%MIX database is the best option to be used as training set. This assumption is supported by the fact that database cross-validation results using this training set are more stable (overpassing 60 % F-measure) than using ENST_wet or MAMI only. We reserve 10 % of this database and Sandvold's database for database cross-validation in future experiments.

In the normalization experiments small differences between training-based and general-based normalization criteria are detected. For this reason we are more confident about the representativeness of our database. Therefore, we adopt the "general" normalization criteria to be used in future experiments.

Chapter 5

Object-level temporal evolution descriptors

In the previous chapter we have selected three databases to work with, the 90%MIX database to be used as training set, and the 10%MIX and Sandvold's databases to be used as testing databases. A normalization criteria has also been established by taking the "general" normalization parameters to normalize the values of the descriptors.

In this chapter we evaluate the performance of the bag-of-frames (BOF) approach to classify bass drum, snare drum, hi-hat and cymbal sounds in polyphonic music. We also try to determine if by adding object-level temporal evolution descriptors for each feature time series improved classification results are obtained. We understand the term "object" as: every sound event starting on an onset and finishing 150 milliseconds before (or in the next onset if this new onset it is located before 150 ms.)

From now on all the experiments will use the same databases and normalization criteria previously defined.

5.1 Background

As mentioned in section 2.4 the traditional bag-of-frames (BOF) approach in pattern-recognition algorithms means to calculate a general set of frame-level feature values and then compute within each feature time series some statistical descriptor like its mean (and sometimes its variance). An array of those descriptors is thus used as a condensed representation of the analyzed sound event or song.

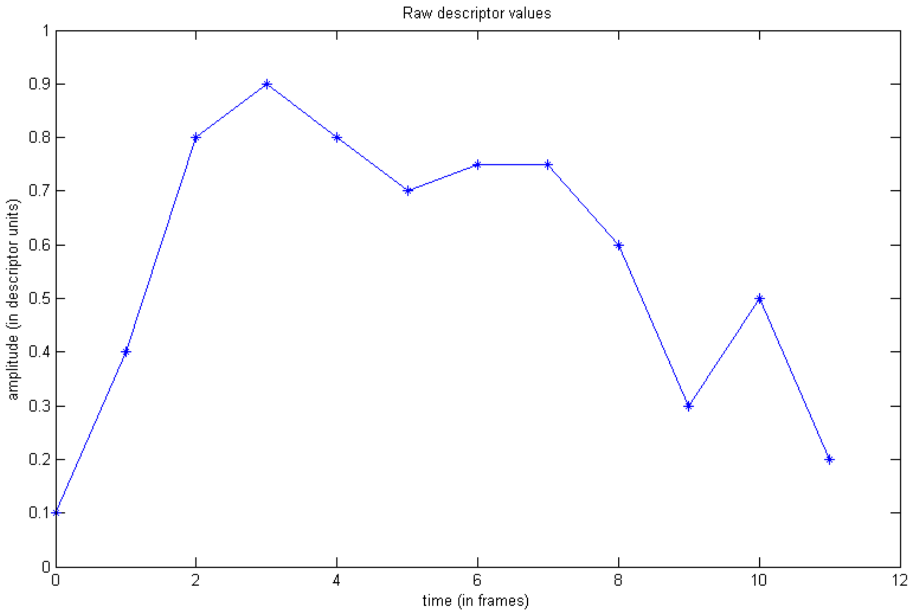
Although the BOF approach has proven to be sufficient for classification of isolated sounds and soundscapes it seems to present some limitations when dealing with polyphonic music (see section 2.4). In order to avoid the limitations of this approach, i.e. the neglection of temporal-evolution information, some new descriptors have been proposed in the recent literature. In [48] Ricard and Herrera implemented a system for automatically describe morphological characteristics of sound events (sound objects). The goal of this system was to describe each sound according to four morphological criteria including: dynamic profile (energy temporal-evolution), pitchness, pitchness profile (temporal-evolution of pitchness) and pitch profile (temporal-evolution of pitch). These morphological descriptors were discrete labels automatically assigned to each song after a feature extraction and classification process. In [38] and [37] a large set of temporal, and non-temporal statistics were computed from low-level feature time series. A total of 154 descriptors were extracted including the first four statistical moments (mean, standard deviation, skewness and kurtosis), autocorrelation, spectral centroid, spectral bandwidth, spectral slope, modulation, together with nonlinear time series descriptors like the mean and the standard deviation of the distances and angles in the phase space. The final number of computed descriptors was close to 70,000. After a novel feature selection process these descriptors were used to model timbre distances and genre classification.

For the case of percussion, only few authors have investigated the use of object-level temporal evolution descriptors and when these descriptors were used refer only to temporal-evolution of energy or amplitude values. Herrera et. al [29] segmented isolated drum sounds into attack and decay, then several descriptors were computed for each sound segment like log attack-time, temporal centroid, and zero-crossing rate. Some general descriptors for the whole drum event were also computed like MFCCs and energy descriptors. In Tindale et. al [63] isolated snare sounds were classified using only temporal descriptors extracted from the attack section of the sound. This descriptors included: zero-crossings, attack time, RMS and temporal centroid. Paulus and Klapuri [42] used narrow-band features to describe the temporal evolution together with "traditional" spectral features to transcribe drum events in polyphonic music. These narrow-band features were computed from the energy evolution of several frequency sub-bands. In [50] the authors employed their automatic feature “creator” introduced in [41] to classify pandeiro sounds. In this system basic signal processing operators, like FFT or filters, and spectral operators, like spectral centroid or spectral skewness, were combined by an evolutionary algorithm to get a very large set of features (about 10^{20}).

In previous experiments we compute the mean, variance, minimum and maximum value for each descriptor for every onset plus 150 milliseconds (12 frames), or until the next onset if the distance between onsets is less than 150 ms. In this chapter besides these previously computed object-level descriptors we decide to compute several descriptors from the temporal evolution of every frame-based descriptor described in section 2.3. These object-level descriptors can be grouped in two categories: amplitude-related object descriptors and time-related object descriptors. This categorization can be seen in figure 5.1 where we plot an example of the frame-based evolution of one hypothetical raw feature values, designating the x axis as amplitude (in descriptor’s units) and the y axis as time (in frames).

Amplitude-related object descriptors:

mean:	0.567
var:	0.069
min:	0.100
max:	0.900
skewness:	0.427
kurtosis:	0.344



Temporal-related object descriptors:

temporal skewness:	0.544	slope:	0.459
temporal kurtosis:	0.377	norm. attack:	0.908
temporal centroid:	0.435	norm. decay:	0.262
max. norm. position:	0.333	attack:	0.587
min. norm. position:	0.083	decay:	0.475

Figure 5.1: Object-level descriptors: toy example.

Amplitude-related object descriptors:

This category includes the previously defined descriptors mean, variance, minimum and maximum plus:

- **skewness:** it is a measure of the asymmetry of a distribution around its amplitude mean value. The skewness value is normalized between 0 and 1, thus a skewness value below 0.5 indicates a signal with more high-amplitude values. A skewness value above 0.5 indicates a signal with more low-amplitude values. It is the third order statistical moment.

- **kurtosis:** it is the fourth order statistical moment. It gives a measure of the flatness of a distribution around its amplitude mean value. The output value is normalized between 0 and 1.

Time-related object descriptors:

- **temporal_skewness:** it is a measure of the asymmetry of a distribution around its temporal mean value. The skewness value is normalized between 0 and 1, thus a skewness value below 0.5 indicates a signal with more "energy" in the last frames. A skewness value above 0.5 indicates a signal with more "energy" in the first frame values. It is computed like the third statistical moment but taking the time series of descriptor values as a histogram.

- **temporal_kurtosis:** it gives a measure of the flatness of a distribution around its temporal mean value. It is computed like the fourth order statistical moment, but taking the time series as the histogram of the signal. The output value then is normalized between 0 and 1.

- **temporal_centroid:** it is the temporal center of gravity of the spectrum.

- **maximum normalized position:** is the position (in time) for the maximum value of the time series normalized by its length.

- **minimum normalized position:** is the position (in time) for the minimum value of the time series normalized by its length.

- **slope:** for this descriptor we first compute the slope of the linear regression of the data (x axis in frames, y axis in feature's units). Then, in order to get limited-range values we compute the angle of this slope by calculating the arc tangent of this value. The output value is normalized between 0 and 1, thus a 0 degree slope is equal to 0.5.

- **normalized attack:** first the position for maximum point of the time series is found. Then the slope descriptor for the time series that goes from the initial point to the maximum point is computed. When computing the slope the x axis is a normalized array going from 0 to 1.

- **normalized decay:** first the position for maximum point of the time series is found. Then the slope descriptor for the time series that goes from this maximum point to the end is computed. When computing the slope the x axis is a normalized array going from 0 to 1.

- **attack:** the same as in the normalized attack but, when computing the slope value, instead of normalizing the x axis from 0 to 1 an array of values in seconds is used. This array is obtained by multiplying the number of frames of the time series by the sample rate and dividing this value by the hop size (in samples).

- **decay:** the same as in the normalized decay but, when computing the slope value, instead of normalizing the x axis from 0 to 1 an array of values in seconds is used. This array is obtained by multiplying the number of frames of the time series by the sample rate and dividing this value by the hop size (in samples).

5.2 Methodology

In order to evaluate the bag-of-frame (BOF) and the object-level temporal-evolution descriptors we first build a set of databases namely:

- **strict-BOF:** in this case we compute only the mean value for the descriptors used in section 2.3 obtaining a 153 feature vector for each event.

- **best-time-evolution:** starting from the same list of frame-level descriptors we compute the whole set of descriptors described in the previous section (amplitude and temporal-related object descriptors) except the mean values. This database can be seen as all-but-mean set of descriptors. Then by applying a ranker algorithm[1] we select the best 153 descriptors to equalize the number of descriptor with the "strict BOF" database.

[1] In this case we use the InfoGainAttributeEval from Weka. This feature selection algorithm evaluates every attribute by measuring information gain with respect to the class.

- **BOF:** this is the same database used in chapter 4. Starting from the frame values we compute the mean, variance, minimum and maximum values.

- **BOF+TDESC:** this database includes all BOF descriptors plus all object-level temporal-evolution descriptors summing up to 2,449 descriptors.

Once completed the previous step, we compare the binary classification results obtained from the databases using decision trees and support vector machines (SVM) learning algorithms.

First we evaluate the strict-BOF set against the best-time-evolution database using the decision tree algorithm (5% MinNumObj). Then with the same algorithm and parameters we compare the BOF database against the BOF+TDESC database.

Since it is not recommended to try the SVM algorithm with a very large set of descriptors we perform a 10-fold CFS feature selection, as done in subsection 4.2.2, on both BOF and BOF+TDESC databases. Then we select the same number of "best" features for both databases to compare the classification results for the "selected-BOF" vs. "selected-BOF+TDESC" databases. In this last experiment we try to determine the best classification values for the "selected-BOF" and the "selected-BOF+TDESC" databases by performing a grid parameter search on the SVM parameters. This grid search is performed by iterating over several hundred of possible values for C and gamma and three kernels i.e. linear, polynomial (degree two), and rbf.

5.3 Results and Discussion

5.3.1 Results

Strict-BOF vs. best-time-evolution

The results for the first set of experiments can be seen in table 5.1. From these results we can observe that best-time-evolution descriptors (best_T_DESC) outperforms strict-BOF in almost all instruments and databases (except for cymbals in Sandvold's database where no improvement is shown). The number of selected descriptors seems to be smaller for the best_T_DESC databases, specially for the case of the snare drum where only 2 descriptors (instead of 8 to the strict-BOF case) are selected by the algorithm to classify sound events i.e. slope of the barkband #4 and max value for barkband_ratio #21. The average improvement on using best-time-evolution descriptors is 4.45 % for all instruments and databases. This suggest that using descriptors from the "time-evolution" database is better than using only mean descriptor values.

C4.5			F-measure		
Instrument	**Train on**	**# Descriptors**	**90%MIX**	**Sandvold's**	**10%MIX**
bass drum	90MIX_Strict BOF	6	0.759	0.712	0.781
bass drum	90MIX_best_T_DESC	4	**0.781**	**0.783**	**0.799**
snare	90MIX_Strict BOF	8	0.677	0.594	0.686
snare	90MIX_best_T_DESC	2	**0.704**	**0.631**	**0.733**
hi-hat	90MIX_Strict BOF	6	0.684	—	0.709
hi-hat	90MIX_best_T_DESC	5	**0.754**	—	**0.756**
cymbals	90MIX_Strict BOF	7	0.679	**0.640**	0.690
cymbals	90MIX_best_T_DESC	5	**0.742**	**0.640**	**0.763**

Table 5.1: Classification results for strict BOF and "best" object-level temporal-evolution descriptors. See section 5.2 for further explanations.

BOF vs. BOF+TDESC

When comparing the BOF database (mean, variance, minimum and maximum) with the BOF+TDESC database we also observe that by adding object-level temporal evolution descriptors an improvement on the classification results is obtained. This improvement is observed not only in the 10-fold inter-database results but also in the cross-database evaluations (except for the snare in Sandvold's database where no change is shown). For this evaluation the average performance is increased by 2.95 % when using BOF+TDESC descriptors with almost the same number of selected descriptors. These results can be seen in table 5.2.

Selected BOF vs. selected BOF+TDESC

The classification results for the SVM algorithm (grid search) are shown in table 5.3. Here the number of features in both databases are almost equal and are selected

C4.5			F-measure		
Instrument	**Train on**	**# Descriptors**	**90%MIX**	**Sandvold's**	**10%MIX**
bass drum	90MIX_BOF	6	0.774	0.730	0.776
bass drum	90MIX_BOF+TDESC	6	**0.781**	**0.791**	**0.802**
snare	90MIX_BOF	8	0.708	**0.623**	0.722
snare	90MIX_BOF+TDESC	8	**0.731**	0.622	**0.739**
hi-hat	90MIX_BOF	4	0.717	—	0.720
hi-hat	90MIX_BOF+TDESC	5	**0.762**	—	**0.749**
cymbals	90MIX_BOF	6	0.720	0.630	0.684
cymbals	90MIX_BOF+TDESC	6	**0.741**	**0.640**	**0.763**

Table 5.2: Classification results for BOF and BOF+TDEC descriptors. See section 5.2 for further explanations.

by the CFS feature selection algorithm. As in the previous tests we witness an improvement when adding object-level temporal evolution descriptors to a BOF database. In this case the overall performance of the BOF+TDESC_CFS is 3.46 % above the BOF_CFS results. It is also interesting to notice that snare classification results decrease, by adding temporal-evolution descriptors, about 2.5 % on average for both 90% and 10%MIX databases. At the same time an improvement of 12.1 % is produced for snare classification results on Sandvold's database.

LibSVM w/grid search		F-measure		
Instrument	**Train on**	**90%MIX**	**Sandvold's**	**10%MIX**
bass drum	90MIX_BOF_CFS	0.825	0.773	0.823
bass drum	90MIX_BOF+TDESC_CFS	**0.834**	**0.812**	**0.835**
snare	90MIX_BOF_CFS	**0.795**	0.566	**0.806**
snare	90MIX_BOF+TDESC_CFS	0.778	**0.687**	0.773
hi-hat	90MIX_BOF_CFS	0.757	—	0.771
hi-hat	90MIX_BOF+TDESC_CFS	**0.806**	—	**0.802**
cymbals	90MIX_BOF_CFS	0.722	0.638	0.713
cymbals	90MIX_BOF+TDESC_CFS	**0.770**	**0.671**	**0.793**

Table 5.3: BOF vs. BOF+temporal descriptors. LibSVM with grid parameter search.

In order to build the 90MIX_BOF_TDESC_CFS database we select (by CFS feature selection) 56 descriptors for bass drum, 77 for snare drum, 38 for hi-hat and 51 for cymbals[2]. We claim that the addition of object-level temporal evolution descriptors to a "classic" BOF database improves the classification results for these

[2] See appendix A.3 for a full list of selected features

instruments. The next question is how many temporal evolution descriptors remain after the feature selection process. In figure 5.2 we show the ratio between amplitude and temporal-related object descriptors (as defined in section 5.1) after the feature selection process. About 60 % (on average) of the selected descriptors belong to the temporal evolution category.

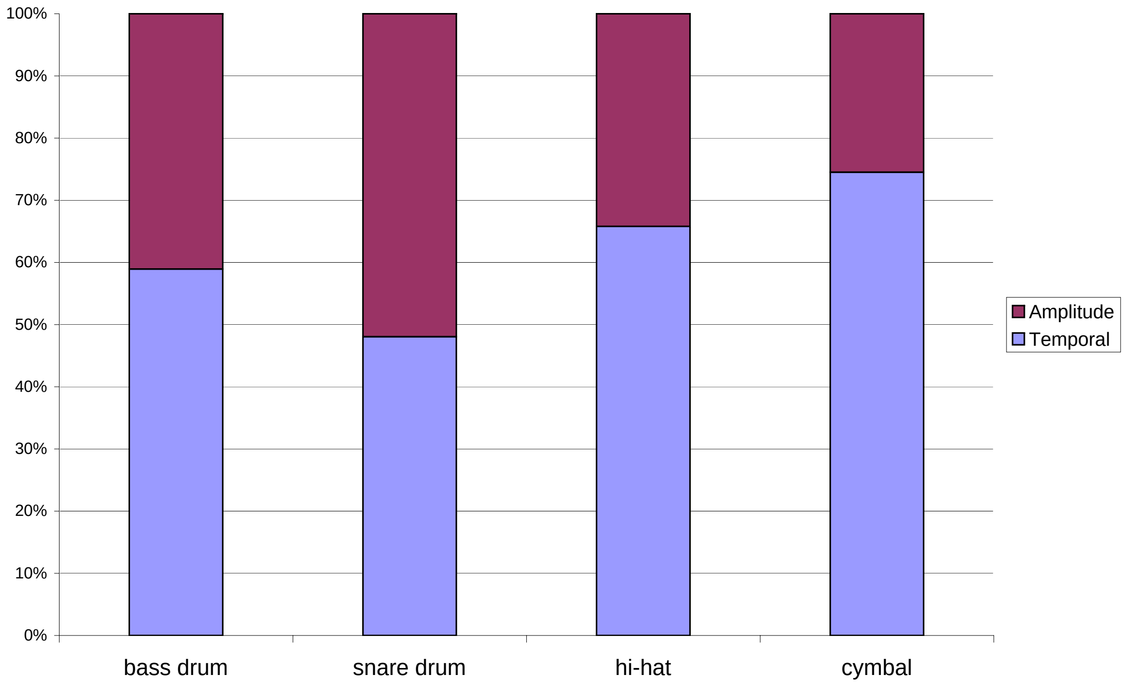


Figure 5.2: Amplitude-related vs. Temporal-related object descriptors per instrument after feature selection.

5.3.2 Main Conclusions

In this chapter we have shown that adding descriptors for the temporal evolution of the time series generated by frame-level descriptors (object-level temporal evolution descriptors) leads to an improvement in the classification results of about 3.6 % on the overall average F-measure.

The best results are obtained using support vector machines with a relative small sub-set of features (selected by the CFS algorithm). With the exception of snare drums and cymbals on Sandvold's database the obtained classification results are near 80 %.

It is worth noticing that these object-level temporal descriptors are about 60 % of the final list of selected features. It is also remarkable that the presence of Bark band derived descriptors in this final list is also about 60% [3].

It seems that with the addition of object-level temporal descriptors we do not reach to "break" the glass ceiling but at least we start to "scratch" its surface.

In the next chapter we will see how the models learned on this chapter perform in a whole drum transcription system.

[3] 64.3 % for bass drums, 62.33 % for snares, 57.9 % for hi-hats and 50.9 % for cymbals

Chapter 6

Drum transcription system

In the previous chapter we obtain three support vector machines models to classify bass drum, snare and hi-hat sounds within polyphonic music. Good classification results are produced by adding object-level temporal evolution descriptors to the classic BOF descriptors.

In this chapter we evaluate the behavior of these models within a whole transcription system. Three databases are analyzed (ENST_wet, MAMI and Sandvold's) adding up a total of 142 songs (30 seconds length). Up to our knowledge, regarding the number of songs and genres, this is the largest evaluation performed on a drum transcription system for polyphonic music.

6.1 Background

Up to date the main evaluation on drum transcription systems was performed in the Music Information Retrieval Evaluation eXchange (MIREX) within the context of the 2005 ISMIR conference [1]. The goal of this contest was to detect drum events produced by bass drums, snares and hi-hats in polyphonic audio. The evaluated data-set was integrated by 50 files (30 second-length) of both live and sequenced music. The overall results of this contest can be seen in table 6.1[2].

[1] see MIREX web site: `http://www.music-ir.org/mirex/2005`

[2] for a detailed view see drum results MIREX 2005 web page `http://www.music-ir.org/evaluation/mirex-results/audio-drum/index.html`

		Average F-measure			
Rank	**Participant**	**Total**	**Bass drum**	**Snare**	**Hi-hat**
1	Yoshii, Goto, & Okuno	0.670	0.728	0.702	0.574
2	Tanghe, Degroeve, & De Baets 3	0.611	0.688	0.555	0.601
3	Tanghe, Degroeve, & De Baets 4	0.609	0.686	0.562	0.590
4	Tanghe, Degroeve, & De Baets 1	0.599	0.677	0.542	0.588
5	Dittmar, C.	0.588	0.606	0.581	0.585
6	Paulus, J.	0.499	0.527	0.430	0.587
7	Gillet & Richard 2	0.443	0.598	0.428	0.334
8	Gillet & Richard 1	0.391	0.533	0.317	0.343

Table 6.1: MIREX 2005: drum detection contest. Overall results.

For the 2007 ISMIR Paulus and Klapuri [42] presented an evaluation of 45 songs from RWC Pop database[3] using HMM with some temporal descriptors (as described earlier in section 5.1) obtaining an overall F-measure of 0.697 and instrument-level results of 0.795, 0.655 and 0.660 for bass drum, snare drum and hi-hat respectively.

In a very recent paper Gillet and Richard [17] evaluate the performance of their transcription system against 28 songs from the ENST_wet database[4] achieving F-measure values of 0.695 for bass drums, 0.583 for snare drums and 0.755 for hi_hats[5] and an overall average F-measure of 0.678.

6.2 Methodology

The experiment set-up for evaluating the transcription capabilities of our system is as follow:

1. Input the 30 seconds audio excerpts extracted from the three previously described databases (ENST_wet, MAMI and Sandvold's) and their ground truth labels.

2. Perform an onset detection using an in-house implementation of the onset detector algorithm proposed by Brossier in [4].

[3] See [20] for a detailed view on this database

[4] See subsection 2.2.1 for an explanation on this paper

[5] These results are for the case of balanced mixtures between drums and music tracks

3. Compare the detected onsets[6] with the original labels to describe the performance of the onset detector.

4. Compute the descriptors used by each model[7] on every onset plus 150 ms (or until the next onset). Apply the models to every set of descriptors to obtain the predicted labels for every onset.

5. Evaluate the results of the predicted labels against the ground truth annotations. As in the MIREX 2005 contest a range of ±30 milliseconds from the true times is allowed.

6. Evaluate the total number of predicted labels per instrument against the total number of ground truth labels per instrument. This evaluation is called relaxed transcription and its goal is to evaluate the description capabilities of the system at a song-level resolution.

6.3 Results and Discussion

6.3.1 Results

Onset detection results

The onset detector performance can be seen in table 6.2. The overall performance per database is as follows:

ENST_wet 0,718, MAMI 0,610, Sandvold's 0,840 and the total average performance is 0,723[8]. It is important to notice that this is not the true onset detector performance since we are comparing ground truth labels against labeled onsets. But this measures show the expected "top" performance for each database and instrument. Since both Sandvold's and MAMI's songs were extracted from commercial

[6] Like in previous experiments onsets are labeled according to the following strategy: every label that is within a threshold of ±30 milliseconds from the onset value is assigned to this onset.

[7] The models are the ones computed in the previous chapter. For every instrument the "90MIX_BOF+TDESC_CFS" database is used as training database for three SVM algorithms, one for bass drum, one for snare drum and one for hi-hat.

[8] Since we are dividing the number of corrected labeled onsets by the total number of labels we are computing a "recall" measure

Instrument	database	ground truth	detected onsets	onset / g. truth
bass drum	ENST_wet	3771	2717	0.759
bass drum	MAMI	2150	1252	0.582
bass drum	Sandvold's	490	398	0.793
bass drum	**all**	**6411**	**4367**	**0.706**
snare	ENST_wet	3766	2028	0.602
snare	MAMI	1464	813	0.619
snare	Sandvold's	438	357	0.887
snare	**all**	**5668**	**3198**	**0.667**
hi-hat	ENST_wet	5234	4180	0.793
hi-hat	MAMI	3176	1708	0.629
hi-hat	**all**	**8410**	**5888**	**0.723**

Table 6.2: Onset detector performance per instrument.

CD's we can hypothesize that the onset performance difference is due to the differences in the labeling process. In Sandvold's the annotations were done based on pre-detected onsets while annotations in MAMI database were done basically "by ear" (see [62]). It is also worth to notice that Sandvold's labels do not include every drum instance since only pre-detected onsets were labeled when building this database. Therefore it could be the case that our onset detector detects real drum events that are un-labeled. Since we do not have a ground truth label for those events our transcription system will mark the prediction on this onset as incorrect.

The best onset recall measure reported in the MIREX drum contest was 0.725 for the "Tanghe, Degroeve, & De Baets 1" algorithm. This measure is comparable with the one obtained by our system.

Transcription results

Transcription results are shown in tables 6.3 and 6.4. The "ground truth", "detected" and "correct" columns show the sum of all songs results for each database. The "precision", "recall" and "F-measure" columns show the average results for all songs on every database. The "all" measures are computed after evaluating a "bag" of all 142 songs and its labels and not by simply averaging the results of each database.

By analyzing individual instrument results for "all" songs we observe a better performance for bass drum (69.9 %) followed by snare drum (65.2 %) and hi-hat (62.6 %). As expected, over-detection rates are present on Sandvold's database (see subsection 6.3.1), this could decrease the performance on this database. We also notice over-detection on MAMI's snare drums, that could be due to label alignment problems or specific biases of the model.

		Total results			**Average results**		
Instrument	**database**	**ground truth**	**detected**	**correct**	**precision**	**recall**	**F-measure**
bass drum	ENST_wet	3771	2615	2298	0.873	0.656	**0.710**
bass drum	MAMI	2150	1938	1108	0.552	0.509	**0.514**
bass drum	Sandvold's	490	655	354	0.573	0.735	**0.605**
bass drum	**all**	**6411**	**5208**	**3760**	**0.701**	**0.558**	**0.699**
snare	ENST_wet	3766	2054	1635	0.765	0.508	**0.558**
snare	MAMI	1464	2050	746	0.412	0.565	**0.445**
snare	Sandvold's	438	694	238	0.350	0.616	**0.363**
snare	**all**	**5668**	**4798**	**2619**	**0.623**	**0.550**	**0.652**
hi-hat	ENST_wet	5234	4436	3998	0.858	0.743	**0.785**
hi-hat	MAMI	3176	2342	1371	0.487	0.530	**0.412**
hi-hat	**all**	**8410**	**6778**	**5369**	**0.621**	**0.652**	**0.626**

Table 6.3: Average transcription results per instrument.

database	**precision**	**recall**	**F-measure**
ENST_wet	0.832	0.636	**0.684**
MAMI	0.484	0.535	**0.457**
Sandvold's	0.462	0.676	**0.484**
all	**0.648**	**0.587**	**0.659**

Table 6.4: Average transcription results per database.

Considering database-level results the best results are obtained on ENST_wet (68.4 %) and the worst on MAMI (45.7 %). The overall performance of the system over 142 songs is 0.659.

A comparison between our system and state-of-the-art systems described in section 6.1 can be seen in table 6.5. If we compare our results with the MIREX 2005 drum transcription contest, our system would be ranked in second place after "Yoshii, Goto and Okuno" system (0.67 overall result). Compared with Paulus and Klapuri [42] our system preforms only 3.8 % below. This could be considered as a good result taking into account that Paulus and Klapuri's results come from "pop" songs only, thus with less variability than our database that includes several genres. To better compare the performance of our system against Gillet and Richard's system [17], since their results are presented over 28 songs coming from the minus_one ENST_wet database, we also compute the transcription results for minus_one ENST_wet only database[9]. The performance of our system over ENST_wet only database (see “our sys. ENST” in table 6.5) is 2.1 % worst than Gillet and Richard’s overall result, obtaining +3.5 % for bass drum, -7.9% for snare and -1.8% for hi-hat.

It is worth notice that within this state-of-the-art list the only “classic” pattern recognition system is the one presented by Tanghe et. al [61]. This algorithm uses a pretty sophisticated onset detector followed by a standard feature extractor and support vector machines classifier with normalization coefficients extracted from the test database. Our system outperforms Tanghe’s by 4.8 % in the overall measure. This difference could be mainly explained by the presence of better descriptors (possibly temporal descriptors) and a bigger training database in our system.

At this stage we can claim that our system’s performance could be among the top ranks compared with state-of-the-art algorithms.

	F-measure						
	overall	bass drum	snare	hi-hat	# songs	presented on	comments
Yoshii et. al	0.670	0.728	0.702	0.574	50	MIREX 2005	1st place
Tanghe et. al	0.611	0.688	0.555	0.601	50	MIREX 2005	2nd place
Paulus & Klapuri	0.697	0.795	0.655	0.660	45	ISMIR 2007	RWC-pop db
Gillet & Richard	0.678	0.695	0.583	0.755	28	IEEE 2008	m_one ENST db
our system	0.659	0.699	0.652	0.626	142		
our sys. ENST	0.657	0.730	0.504	0.737	28		m_one ENST db

Table 6.5: Transcription results compared with state-of-the-art systems.

Since simple average measures are not fully informative we decide to plot a histogram with the F-measure results for all 142 songs per instrument. These histograms are shown in figure 6.1 where we can easily see that better results are obtained for bass drums and hi-hats and more results close to the chance-level performance are obtained for snare drums. This information can not be derived from average results only.

[9] As in Gillet and Richard’s paper, in this case, we consider as valid a time deviation of up to 50 ms.

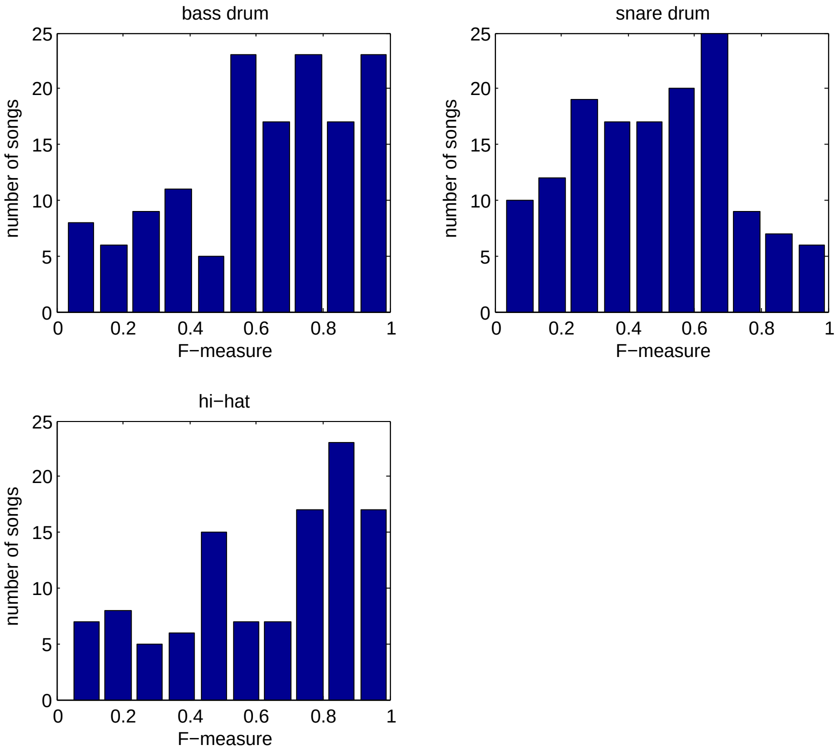


Figure 6.1: Transcription results: F-measure distribution over all songs (142 - 30 sec. excerpts).

Relaxed transcription results

Taking into account that our final goal is to derive song-level percussion descriptors, it is useful to know the performance of our system to estimate the total number of drum events per song (e.g. how many bass drum, snare drum or hi-hat strikes a particular song has). Thus if we are able to correctly estimate these values we could construct accurate song-level percussion descriptors relaying on those correctly estimated values. These descriptors could be used to characterize a song as having, for example, a lot of snare drum, no hi-hat, etc. Hence we decide to evaluate, in these sets of experiments, the performance of our system to detect the correct number of drum events per song. In order to do that we now consider as "correct" the total number of e.g. hi-hats events in the whole audio file discarding time-information.

The results for these experiments can be seen in tables 6.6 and 6.7. We call these evaluations as "relaxed transcription". By looking at instrument level results we detect better classification results on bass drum (0.822) and hi-hat (0.794) followed by snare results (0.698). This can also be seen in the histogram plots (figure 6.2) where a big number of songs are near 100% performance.

Instrument	**database**	**ground truth**	**detected**	**correct**	**precision**	**recall**	**F**
bass drum	ENST_wet	3771	2615	2586	0.991	0.737	**0.802**
bass drum	MAMI	2150	1938	1805	0.948	0.864	**0.875**
bass drum	Sandvold's	490	655	454	0.742	0.936	**0.778**
bass drum	**all**	**6411**	**5208**	**4845**	**0.925**	**0.822**	**0.822**
snare	ENST_wet	3766	2054	2005	0.941	0.620	**0.685**
snare	MAMI	1464	2050	1409	0.730	0.973	**0.795**
snare	Sandvold's	438	694	363	0.561	0.899	**0.570**
snare	**all**	**5668**	**4798**	**3777**	**0.789**	**0.798**	**0.698**
hi-hat	ENST_wet	5234	4436	4360	0.969	0.846	**0.890**
hi-hat	MAMI	3176	2342	2123	0.792	0.776	**0.667**
hi-hat	**all**	**8410**	**6778**	**6483**	**0.893**	**0.816**	**0.794**

Table 6.6: Relaxed average transcription results per instrument.

database	**precision**	**recall**	**F**
ENST_wet	0.967	0.734	**0.792**
MAMI	0.823	0.871	**0.779**
Sandvold's	0.652	0.918	**0.674**
all	**0.869**	**0.812**	**0.771**

Table 6.7: Relaxed average transcription results per database.

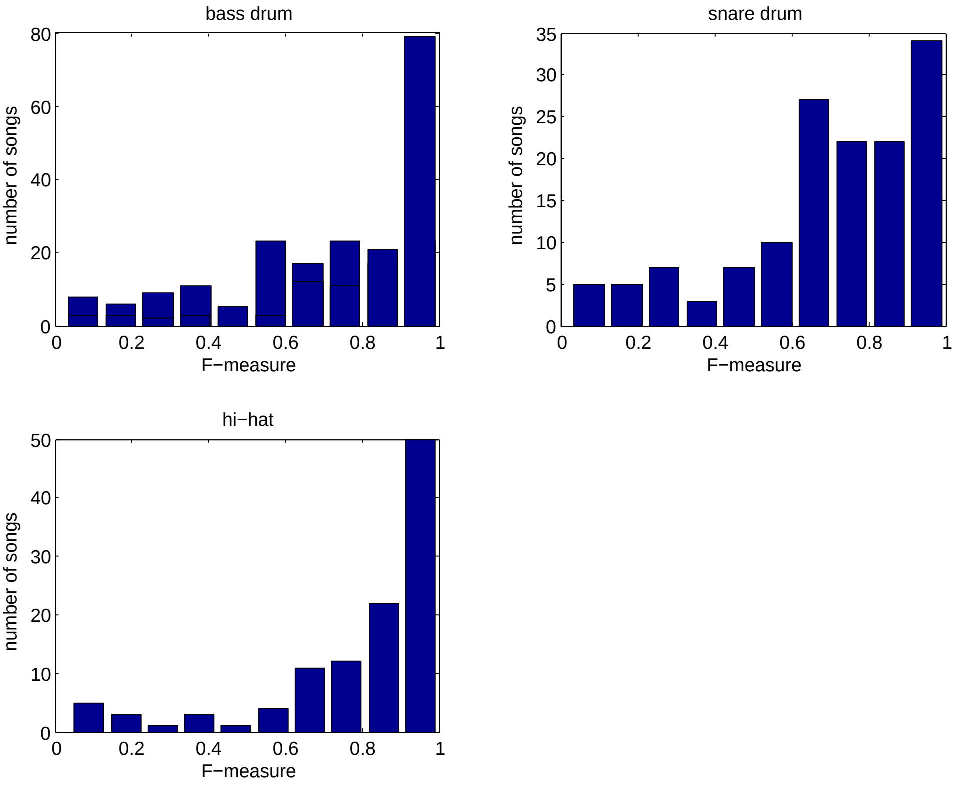


Figure 6.2: Relaxed transcription: F-measure distribution over all songs (142 - 30 sec. excerpts).

Evaluating database-level results we observe that ENST_wet and MAMI present similar measures and Sandvold's performance is about 10% lower. As mentioned above this could be explained by the labeling strategy of this database. The overall performance measure of this "relaxed" transcription system is 77.1 %.

6.3.2 Main Conclusions

In this chapter we evaluate the performance of a whole drum transcription system that uses the classification models learned in the previous chapter.

From the transcription results, compared with state-of-the-art systems, we derive that our system can be placed among the best existing ones, even though there is still a lot of room for improvement. It is interesting to notice that these good results are achieved by a relative simple algorithm using the “classic” pattern-recognition approach. We relate these results to the presence of better descriptors, specially the temporal ones, and a good training database.

Since our final goal is to derive song level percussion descriptors we evaluate the performance of our system at this level obtaining an overall result of 77.1 % when evaluating all databases (142 songs). These results encourage us to investigate if useful percussion descriptors could be computed, at a song-wise level, from the transcription output of the system. This evaluation is done in the next chapter.

Chapter 7

Mid-level percussion descriptors

In this chapter we present and compute several percussion-related descriptors extracted from the output of the transcription system presented in the previous chapter. Then we explore the usefulness of these new descriptors for some music information retrieval (MIR) tasks such as genre classification, danceability estimation and Western / non-Western classification.

7.1 Background

Provided that we can detect the number of occurrences of bass drum, snare and hi-hat events in polyphonic music with an average precision and recall of more than 70% (see subsection 6.3.1) in this chapter we will try to use this information to build mid-level percussion descriptors.

In [28] and [51] two percussion-related descriptors were presented and evaluated with promising results namely:

- **Percussion Index:** a ratio between the number of detected percussion events and the number of detected onsets.
- **Percussion profile:** the relative amount of bass drum, snare, cymbals, and non-percussion events (normalized by the total number of onsets).

Following this idea of percussion related descriptors we decide to compute and evaluate the following mid-level percussion descriptors starting from the classification models derived in the previous chapters. Some of this descriptors appear as suggested future work in [51] but up to our knowledge they have not been implemented nor evaluated on previous works yet.

Computed mid-level percussion descriptors:

- **Percussion profile** (as explained before): for bass drum, snare, hi-hat and drum[1] .
- **Intra instrument ratio:** the ratio between percussive instruments namely: bass drum/snare, bass drum/hi-hat and snare/hi-hat
- **Instrument per minute:** the number of detected events per minute for bass drum, snare, hi-hat and drum.
- **Intra instrument interval (iii):** we first construct a histogram with the differences between successive events of the same instrument. Then the first and second peak values are taken to describe the most frequent intervals between each instrument. Thus we compute: first and second-iii-peak for bass drum, snare and hi- hat. This is the only group of descriptors that strongly depends on the correct temporal estimation of drum events, hence the less reliable (see section 6.3 for a comparison between transcription and relaxed transcription results).

At the end of this process we get 17 mid-level percussion descriptors for each song. To evaluate the usefulness of these new descriptor within the MIR field we have focused on those applications where percussion related information could add relevant information. We believe that genre and tempo-related processes could take advantage of these new descriptors. It seems clear that, for some cases, genre definitions are mainly based on rhythm (e.g. waltz, blues, tango. etc.) or at least that drum-related events are one of the key aspects on genre identification by humans. We evaluate the mid-level percussion descriptors for genre and sub-genre classification.

[1] In this context drum means the number of detected onsets that are labeled as bass drum, snare or hi-hat

We choose for the sub-genre evaluation “electronic” sub-genres because within this group percussion seems to be one key aspect for class discrimination.

In a yet unpublished paper Gómez and Herrera [19] present an algorithm for automatic discrimination between Western and non-Western music traditions. This could also be an interesting task to evaluate percussion descriptors, since percussion types, and tonal features, seem to be the most relevant differences between both music traditions. Within tempo-related application we will evaluate our descriptors to estimate the danceability of musical tracks (i.e. the easiness with which one can dance to it [59]).

7.1.1 Units of study

- For genre classification we use an in-house database of 30 seconds excerpts extracted from 350 songs equally distributed in 7 genres: classic, dance, hip-hop, jazz, pop, rhythm'n blues and rock.

- For electronic genre classification we perform our experiments on an in-house database of 270 songs (30 seconds length each) equally distributed among the following genres: ambient, drum'n bass, house, techno and trance.

- For danceability tests we use an in-house database of 374 song excerpts of 30 seconds equally distributed in three classes (non-danceable, mid-danceable and high-danceable).

- For Western / non-Western experiments we use an in-house database of 139 Western songs from 16 genres including classical, jazz, rock, pop, religious, hip-hop, etc. and 139 non-Western songs including songs from Africa, Arabian countries, central Asia, China, Japan and Java.

7.2 Genre classification

7.2.1 Methodology:

1. Compute, for every song on the database, the mean value of a set of "classic" descriptors to be used as baseline for the evaluation. The list of these descriptors is as follows: barkbands, barkbands kurtosis, barkbands skewness, barkbands spread, spectral centroid, spectral crest, spectral decrease, spectral dissonance, spectral energy, spectral energyband high, spectral energyband low, spectral energyband middle high, spectral energyband middle low, spectral flatness db, spectral flux, spectral hfc, spectral kurtosis, MFCC, spectral skewness, spectral spread and temporal zerocrossingrate (see section 2.3 for an explanation on these descriptors). We call this set of 60 descriptors as "timbral descriptors".

2. Compute the mid-level percussion descriptors on the same database (17 descriptors)

3. Determine the "best" classification values by performing a grid search on the parameters of the SVM classification algorithm for "timbral", "percussion" and "timbral+percussion" descriptors [2]

4. Evaluate the relevance of the mid-level percussion descriptors in the classification process.

5. Complement the analysis by computing two feature selection algorithms for the "timbral+percussion" database: CFS and reliefF. Evaluate the list of selected descriptors to document the presence, or absence, of the percussion descriptors within the selected features.

6. Finally, select the best percussion descriptors (by CFS in 10-fold). Apply an

[2] This database is obtained by combining the descriptors of the two previous databases resulting in a set of 77 descriptors.

Expectation Maximization (EM)[3] clustering algorithm to automatically group the songs into clusters. The main idea behind this experiment is to evaluate the usefulness of the percussion descriptors to form homogeneous and meaningful groups of songs. If this happens it could illustrate the capability of these new descriptors to be used to browse and recommend songs.

7.2.2 Results and Discussion

The genre classification results can be seen in table 7.1 and figure 7.1. From these results it is interesting to notice that by using the percussion descriptors only, good discrimination rates can be achieved for classic (81.8 %), rock, dance and hip-hop (about 60%). The overall classification for the percussion-only data set is about 12 % below "timbral" descriptors. When combining "timbral" and "percussion" descriptors a small improvement in the overall result is observed (+2.1%). It is worth notice that big improvements are produced in dance (+12.7 %) and pop (+15 %) results, whereas results for rock and rhythm'n blues decrease 5.6 % and 4.4 % each.

The best accuracy results for the last MIREX 2007 genre contest [4] was 68.29 % for 10 genre classification task. While they are not exactly the same, the accuracy measure from this contest can be compared with the precision measure in our study. In this context we can say that our results, informally speaking, are comparable with the ones obtained in MIREX 2007. Our overall precision result is 2.8 % above the best MIREX result but we have 3 genres less to classify.

[3] The EM algorithm is an iterative method for computing maximum-likelihood estimates on incomplete data. Every iterative step alternates between an expectation step, computing an expectation of the likelihood by including the unseen variables as if they were known, and a maximization step, computing the maximum likelihood estimates of the parameters by maximizing the expected likelihood found on the previous step (see [1] for further information).

[4] see MIREX url: `http://www.music-ir.org/mirex/2007/index.php/Audio_Genre_Classification_Results`

.	**Timbral**			**Percussion**			**Timbral + Percussion**		
Genre	**prec.**	**recall**	**F**	**prec.**	**recall**	**F**	**prec.**	**recall**	**F**
classic	0.818	0.900	**0.857**	0.750	0.900	**0.818**	0.863	0.880	**0.871**
dance	0.623	0.760	**0.685**	0.611	0.660	**0.635**	0.804	0.820	**0.812**
hip-hop	0.755	0.740	**0.747**	0.571	0.640	**0.604**	0.709	0.780	**0.743**
jazz	0.805	0.674	**0.733**	0.542	0.531	**0.536**	0.667	0.735	**0.699**
pop	0.489	0.460	**0.474**	0.439	0.500	**0.467**	0.576	0.680	**0.624**
rhythm'n blues	0.449	0.440	**0.444**	0.357	0.200	**0.256**	0.450	0.360	**0.400**
rock	0.915	0.860	**0.887**	0.674	0.620	**0.646**	0.949	0.740	**0.831**
Average	**0.694**	**0.691**	**0.690**	**0.563**	**0.579**	**0.566**	**0.717**	**0.714**	**0.711**

Table 7.1: Genre classification results for every set of descriptors (LibSVM grid search).

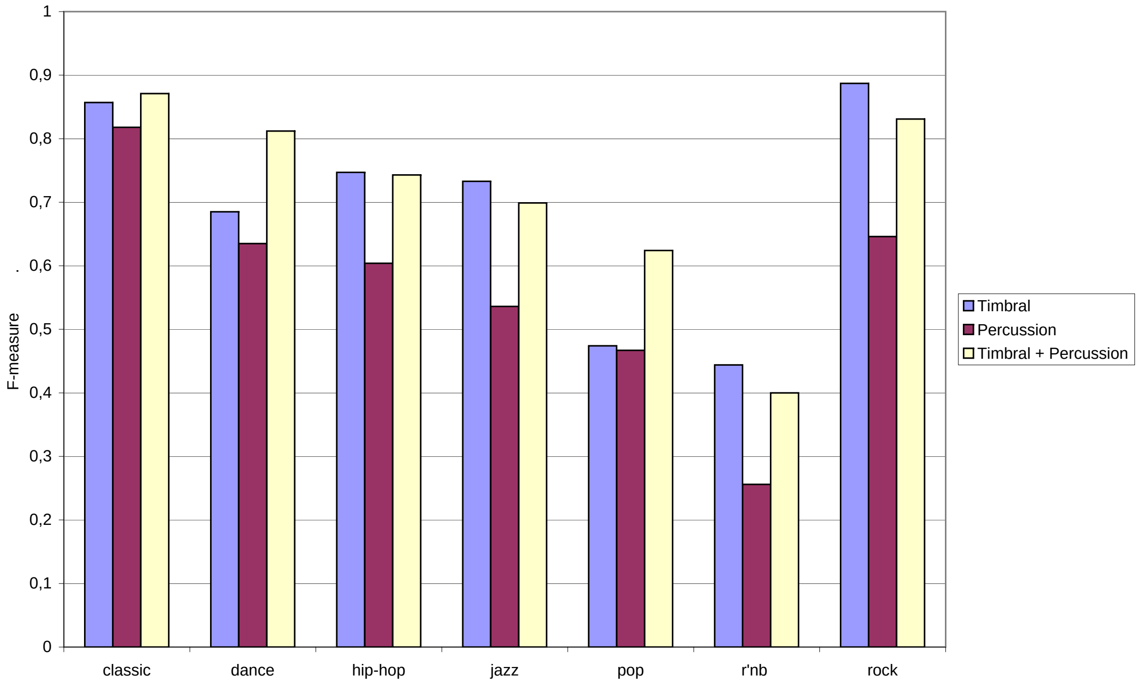


Figure 7.1: Genre classification results per genre and descriptor set.

In table 7.2 the output of two features selection algorithms can be seen. The first column corresponds to features selected more than 5 times by the 10-fold CFS algorithm. The second column reflects how many times the feature have been selected for the CFS algorithm. The maximum number of folds is 10, that means that the feature was choose every trial. The third column corresponds to the ranking of the best 20 features selected by the reliefF algorithm. We observe that among the 17 computed percussion descriptors 6 are selected by the CFS algorithm as informative features and 8 are selected by the ReliefF algorithm. It is worth noticing that for this latter algorithm the 3 best features are percussion features.

CFS			**ReliefF**
Descriptor	**# folds**		**Descriptor**
bass_drum/min	10		**bass_drum/total**
barkbands_0	10		**bass_drum/min**
barkbands_16	10		**drum/total**
barkbands_18	10		spectral_flux
barkbands_24	10		**drum/min**
mfcc_0	10		spectral_dissonance
mfcc_2	10		mfcc_0
spectral_dissonance	10		**snare/total**
spectral_flux	10		**hihat/total**
first_peak_iii_snare	9		**hihat/min**
barkbands_1	9		barkbands_spread
barkbands_5	9		barkbands_20
mfcc_3	9		**snare/min**
snare/min	7		spectral_energyband_high
barkbands_15	7		mfcc_2
bass_drum/total	6		spectral_hfc
bass_drum/snare	5		barkbands_1
hihat/min	5		barkbands_21
barkbands_kurtosis	5		spectral_energyband_low
mfcc_1	5		mfcc_1
mfcc_5	5		
spectral_crest	5		

Table 7.2: Selected features for genre classification.

Clustering

As mentioned in the methodology section we perform a clustering experiment on the genre database using only percussion descriptors. The EM algorithm produces 6 clusters. The genre distribution among each cluster can be seen in table 7.3 and figure 7.2.

Genre	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Total
classic	0	1	0	15	**34**	0	**50**
dance	2	8	**23**	3	0	14	**50**
hip-hop	9	6	2	2	0	**31**	**50**
jazz	6	13	0	**21**	9	1	**50**
pop	2	**25**	8	3	0	12	**50**
r'nb	6	14	5	7	0	18	**50**
rock	**13**	12	4	2	0	19	**50**
Total	**38**	**79**	**42**	**53**	**43**	**95**	350

Table 7.3: Clustering genre database with percussion descriptors (EM - CFS60).

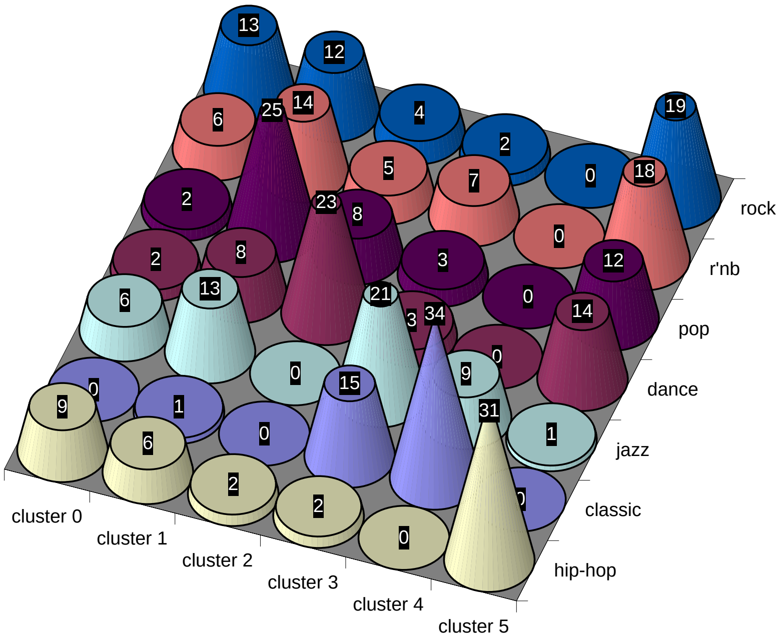


Figure 7.2: Clustering plot: 5 clusters produced by the EM algorithm.

We observe two clusters that are formed mainly by one musical genre namely cluster 2, majority of dance and cluster 4 with majority of classic. Cluster number 3 has the rest of classic songs together with jazz songs. By listening cluster 3 and 4 we can say that are pretty much equals formed by mainly un-percussive songs. The majority of jazz songs within these clusters are played with "brushes" and the two hip-hop songs in cluster 3 are fragments with voice only sounds. Also by listening, we can refer to cluster 0 as having drum sounds with more snares and hi-hats and cluster 5 as having more presence of bass drum events. Cluster 1 is hard to categorize since no characteristic sound nor genre can be detected except for being the cluster with majority of pop songs.

7.3 Electronic sub-genre classification:

7.3.1 Methodology

1. Compute the "timbral" set of descriptors (see subsection 7.2.1).

2. Compute the "percussion" descriptors (see subsection 7.2.1).

3. Use a decision tree algorithm with the same parameters (MinNumObj = 5 %) to evaluate the classification performance for the "timbral", "percussion" and "timbral+percussion" sets of descriptors.

4. To complement the analysis compute two feature selection algorithms for the "timbral+percussion" database: CFS and reliefF. The list of selected descriptors is then evaluated to document the presence, or absence, of the percussion descriptors within the selected features.

7.3.2 Results and Discussion

Results for "electronic" sub-genre classification are depicted in table 7.4. In this experiment we observe that classification results obtained by percussion descriptors only outperform "timbral" descriptors by 7.5 %[5]. The combination of "timbral" and "percussion" descriptors has no significant difference with results from percussion-only descriptors in the overall classification result. But this combination seems to output more balanced results among categories. The number of selected descriptors for every decision tree is as follow: 10 for "timbral", 12 for "percussion" and 10 for "timbral+percussion".

[5] Refer to appendix A.4 to see the resulted decision tree.

C4.5	**Timbral**			**Percussion**			**(Timbral + Percussion)**		
Genre	**prec.**	**recall**	**F**	**prec.**	**recall**	**F**	**prec.**	**recall**	**F**
ambient	0.508	0.556	**0.531**	0.394	0.481	**0.433**	0.625	0.556	**0.588**
drum'n bass	0.438	0.519	**0.475**	0.593	0.648	**0.619**	0.531	0.630	**0.576**
house	0.196	0.204	**0.200**	0.483	0.519	**0.500**	0.397	0.463	**0.427**
techno	0.388	0.352	**0.369**	0.393	0.204	**0.269**	0.413	0.352	**0.380**
trance	0.500	0.389	**0.438**	0.542	0.593	**0.566**	0.449	0.407	**0.427**
Average	**0.406**	**0.404**	**0.403**	**0.481**	**0.489**	**0.478**	**0.483**	**0.482**	**0.480**

Table 7.4: Electronic sub-genre classification.

For this experiment we also perform the feature selection test done in the "genre" evaluation. For the CFS algorithm only 8 features were selected more than 5 times. Within this 8 features 3 come from the percussion set namely: first_peak_iii_hihat, second_peak_iii_bass_drum and first_peak_iii_bass_drum. For the reliefF algorithm 7 percussion features remain within the 20 best being bass_drum/min ranked in the second place.

By personal communication Sesmero provide us with yet unpublished results from electronic sub-genre classification. The author evaluates 250 songs, splitted in the same five sub-genres as in the previous experiment, with several state-of-the-art descriptors (including rhythmic ones) arriving an overall F-measure of 0.907 (Sesmero J. pers. comm). We take 170 songs from Sesmero's database and add our percussion-related descriptors to the state-of-the-art descriptor-set selected by the author. After performing a grid search for SVM parameters we obtain almost the same results (an overall F-measure of 0.894), detecting an improvement on performance for "ambient", "drum'n bass" and "trance" but a decreasing performance for "techno" and "house".

7.4 Danceability estimation:

7.4.1 Methodology

1. Add to the "timbral" set of descriptors an estimation on the beats per minute (bpm) of the song. We compute bpm_value and bpm_estimates (mean, variance, minimum and maximum) using an in-house implementation of the beat

tracking algorithm described in [7]. We call this database "timbral+bpm"

2. Compute the mid-level percussion descriptors.

3. Determine the "best" classification values by performing a grid search on the parameters of the SVM classification algorithm for "timbral+bpm", "percussion" and "timbral+bpm+percussion" descriptors.

4. Evaluate the relevance of the mid-level percussion descriptors in the classification process.

7.4.2 Results and Discussion

Results for Danceability tests are shown in table 7.5 and figure 7.3. From these results we can derive that "percussion" descriptors perform better than "timbral+bpm" and even than "timbral+bpm+percussion". Percussion-only descriptors outperform by 8.9 % and 7.4 % "timbral+bpm" and "timbral+bpm+percussion" respectively, obtaining better results in all three categories. It is interesting to notice that percussion descriptors also outperform Streich and Herrera [59] results which achieve an accuracy of 61.78% in classifing 225 songs into the same three categories (i.e Non-danceable, Mid-danceable and High-danceable) by using Detrended fluctuation analysis (DFA)-derived features.

LibSVM grid p.	**Timbral + bpm**			**Percussion**			**Timbral+bpm+Percus.**		
Class	**prec.**	**recall**	**F**	**prec.**	**recall**	**F**	**prec.**	**recall**	**F**
Non-danceable	0,609	0,602	**0,606**	0,747	0,705	**0,725**	0,654	0,580	**0,614**
Mid-danceable	0,452	0,477	**0,464**	0,529	0,614	**0,568**	0,439	0,534	**0,482**
High-danceable	0,607	0,580	**0,593**	0,671	0,602	**0,635**	0,646	0,580	**0,611**
Average	**0,556**	**0,553**	**0,554**	**0,649**	**0,640**	**0,643**	**0,580**	**0,564**	**0,569**

Table 7.5: Danceability classification.

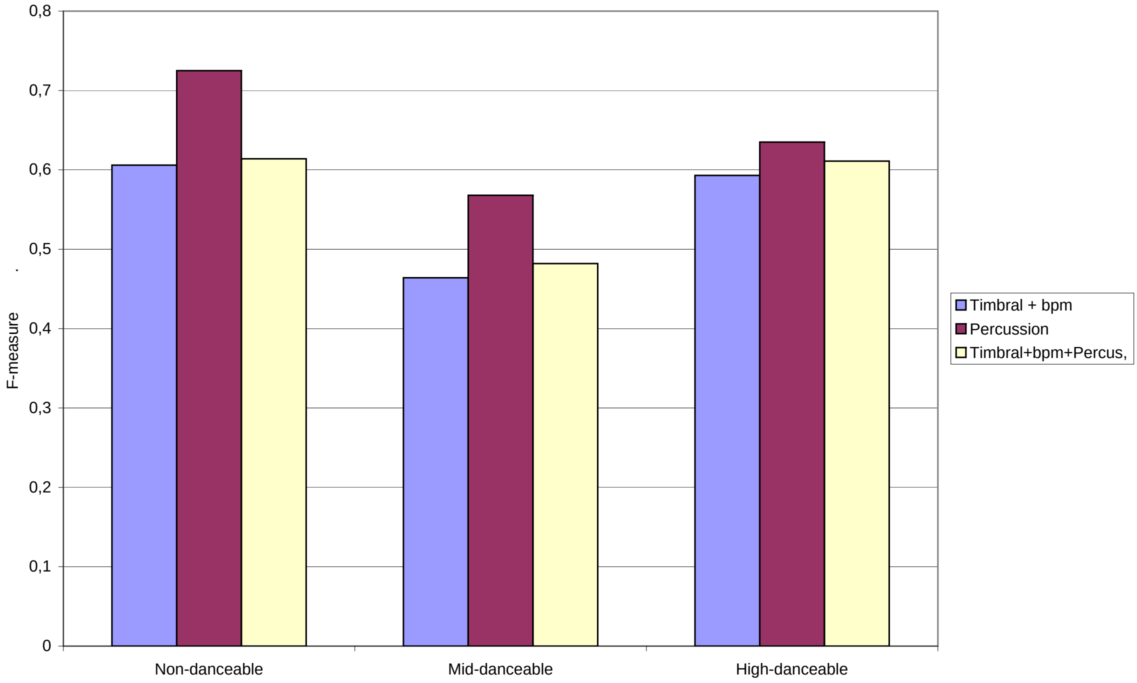


Figure 7.3: Danceability classification results.

7.5 Western and non-Western classification:

7.5.1 Methodology

1. Use the same experiment set-up as in genre classification (see subsection 7.2.1).

7.5.2 Results and Discussion

Finally we try to automatically classify Western vs. non-Western music. The results for these experiments are shown in table 7.6 and figure 7.4. Here we observe an almost linear increment in the classification rates starting by "timbral" descriptors with 78.2 % followed by "percussion" descriptors with 81.6 % (+3.4 %) and "timbral+percussion" with 84.4 % (+2.8 % from "percussion"). It seems clear that adding percussion descriptors help in the process of Western / non-Western song discrimination. Is is also interesting to notice that, as expected, classification results for non-Western music are much better when percussion descriptors are used (more than 8 % above "timbral"). Since this experiment is inspired on an "in press" paper by Gomez and Herrera [19] we also compare our results with those obtained by their work. They classify over 1,500 pieces by using tonal features (with SVM) achieving F-measure values of 86.9 % and 86.1 % for Western and non-Western respectively. As can be seen in table 7.6 we obtain, by using "timbral+percussion" descriptors, F-measure values of 85.6 and 83.3 for Western and non-Western respectively.

LibSVM grid p.	**Timbral**			**Percussion**			**Timbral + Percussion**		
Class	**prec.**	**recall**	**F**	**prec.**	**recall**	**F**	**prec.**	**recall**	**F**
Western	0,717	0,950	**0,817**	0,867	0,748	**0,803**	0,800	0,921	**0,856**
non-Western	0,926	0,626	**0,747**	0,779	0,885	**0,828**	0,907	0,770	**0,833**
Average	**0,821**	**0,788**	**0,782**	**0,823**	**0,817**	**0,816**	**0,853**	**0,845**	**0,844**

Table 7.6: Western / non-Western classification.

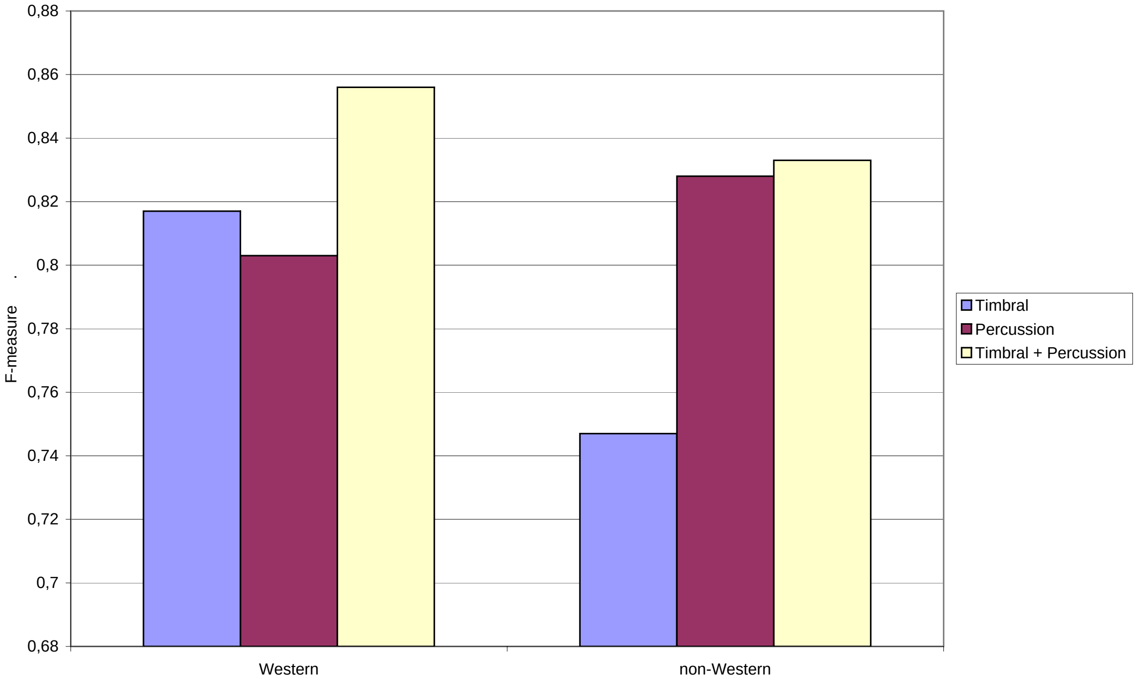


Figure 7.4: Western / non-Western classification results.

As in previous experiments we also evaluate the percussion descriptors by performing two feature selection algorithm namely CFS and reliefF. The list of selected features can be seen in table 7.7. From the output of the CFS algorithm we can observe that 3 percussion descriptors were selected in all 10 evaluation-rounds. From the 20 best features selected by the reliefF algorithm we can see that the 4 best features are all percussion features.

CFS			**ReliefF**
Descriptor	**# folds**		**Descriptor**
barkbands_22	10		**bass_drum/total**
barkbands_25	10		**bass_drum/min**
mfcc_0	10		**hihat/total**
mfcc_3	10		**hihat/total**
mfcc_4	10		spectral_flux
mfcc_11	10		mfcc_3
mfcc_12	10		**drum/min**
barkbands_skewness	10		mfcc_2
spectral_flux	10		**hihat/min**
spectral_skewness	10		mfcc_0
spectral_crest	10		mfcc_9
bass_drum/total	10		spectral_flatness_db
bass_drum/min	10		spectral_dissonance
drum/total	10		mfcc_8
barkbands_8	9		temporal_zerocrossingrate
barkbands_13	9		mfcc_4
mfcc_6	9		mfcc_1
barkbands_0	8		barkbands_22
barkbands_5	8		**snare/min**
mfcc_2	8		spectral_crest

Table 7.7: Selected features for genre classification.

7.6 Main Conclusions

In this chapter we explore the usefulness of a set of mid-level percussion descriptors. These preliminary tests suggest that these mid-level percussion descriptors can help in several MIR tasks such as genre classification, danceability estimation and Western / non-Western classification.

For genre estimation the percussion descriptors give the impression to help in the classification of some genres such as “dance” and “pop”.

The results extracted from the clustering experiment suggest that these percussion descriptors could be used to retrieve similar songs from the point of view of their rhythm or “type” of percussion, e.g. songs containing a lot of bass drum sounds, like the ones belonging to cluster 5 in the test.

The classification of sub-genres can also take advantage of these new descriptors. The “electronic” sub-genre classification tests suggest that again by using these descriptors we can achieve better results than using “timbral” descriptors. Notice the case of the “drum’n bass” sub-genre where almost 15 % of improvement is obtained by using percussion descriptors only.

The case of danceability estimation seems to be an area where these mid-level descriptors offer big help in the classification task, obtaining almost 9 % of improvement by using them only[6].

For Western / non-Western song discrimination we obtain an improvement of 8.6 % in the classification of non-Western music by combining “timbral” and “percussion” descriptors. Since Gomez and Herrera also achieve more than 80 % of accuracy on discriminating Western and non-Western music [19] by using tonal features, it would be interesting to evaluate the performance of our percussion descriptors combined with tonal features,.

These experiments, although exploratory, offer a good starting point for stimulating the use of mid-level percussion related descriptors. It seems clear that these descriptors offer useful information that complements the one provided by classic “spectral” and “timbre” descriptors.

[6] We also perform some exploratory experiments on beat detection, i.e. trying to determine the beat per minute (bpm) measure of a song, since *a priori* this is an area where these percussion-related descriptors could help to improve state of the art algorithms. Unfortunately we could not report any conclusion on this subject yet. More experimentation is needed, and probably new percussion descriptors to retrieve in a more consistent way time-related information for every percussion instrument.

Chapter 8

Conclusions and further work

8.1 Conclusions

Within the present thesis we have conducted percussion related experiments in order to detect and describe percussive events in polyphonic music. Due to their importance in several music styles we decide to focus on the problem of bass drum, snare drum and hi-hat detection and description.

Firstly we build, by combining publicly available databases, a large set of songs (more than 100 songs and more than 10 genres) and evaluate their representativeness by cross-database classification experiments and by comparing normalization parameters. At the end of these experiments we conclude that the evaluated database is a good option, in terms of variability and representativeness, to be used as training set.

Secondly we present and evaluate object-level temporal evolution descriptors to be computed on every frame-level descriptor. After several experiments we conclude that adding these descriptors to classic BOF descriptors leads to an improvement in the classification results of about 3.6 % on overall classification results. This is an interesting outcome provided that we are dealing with "near ceiling" performances.

The best classification results (about 80 %) are obtained by using support vector machines (SVM) with a relative small sub-set of descriptors. Within these selected descriptors more than a half come from the object-level temporal descriptors. We also observe that Bark band related descriptors are always more than 50 % of these selected features.

Thirdly we evaluate the performance of a whole automatic drum transcription system that uses the previously defined SVM models and a state of the art onset detector to transcribe percussion events in polyphonic music.

From the transcription results we derive that our relatively simple algorithm can be placed among the top ranked ones, even though there is still a lot of room for improvement. We relate these good results to the presence of well defined descriptors, specially the temporal ones, and a good training database.

After performing "relaxed" transcription experiments we realize that our system can detect the total number of percussion events within a song with an overall accuracy of 77.1 %. These results encourage us to investigate if useful percussion descriptors could be computed, at a song-wise level, from the transcription output of this system.

Finally we present 17 mid-level percussion descriptors to evaluate their usefulness among MIR tasks in where we suspect these descriptors could provide useful information. These preliminary results tests suggest that mid-level percussion descriptors can help in several MIR tasks such as genre classification, danceability estimation and Western / non-Western classification.

For genre estimation the presented descriptors seem to improve the classification results for some genres (such as "dance" and "pop") and some sub-genres (like "ambient" and "drum'n bass"). The clustering results suggest that these percussion descriptors could be used to retrieve similar songs from the point of view of their rhythm or percussion type.

The tests on "danceability" estimation reflect that this could be an area where the percussion descriptors offer big help. We obtain classification results of about 64 % for three-class experiments. This result, obtained from percussion descriptors only, represents almost 9 % of improvement than using general timbre plus bpm descriptors.

For Western / non-Western song discrimination we obtain an improvement of 8.6 % in the classification of non-Western music by combining timbre and percussion descriptors.

These experiments offer a good starting point for stimulating the use of mid-level percussion related descriptors into MIR tasks. It seems clear that these descriptors provide useful information that complements the one obtained by classic descriptors.

At the end of this work we strongly believe that to add object-level temporal descriptors to a traditional BOF set, and to take a "descriptionist" approach is an excellent path to derive useful information from polyphonic music. We humbly encourage other authors to dive into these waters.

8.2 Further work

Several things can be done in order to improve the present percussion description system like: to train more percussion instruments, to develop instrument-specific onset detectors, to derive new object-level temporal descriptors, to evaluate other machine learning algorithms or to use localized models like in [68] and [52]. One interesting path could be to add "knowledge" to the system by e.g. sequence modeling [16] or some other heuristic information.

It could be important to evaluate the influence of pre-processing steps like harmonic/noise decomposition, drum enhancement by source separation or by simple band-filtering of the input signal.

We believe that more object-level temporal descriptors must be evaluated beyond those described in this thesis e.g. non-linear time series related descriptors like the mean and the standard deviation of the distances and angles in the phase space [37].

Finally we see as an interesting area of development the use of techniques depicted in this thesis to detect and describe more general sound objects (i.e. sounds that share some characteristic but are not necessarily produced by the same type of instrument) going beyond instrument-level sounds to focus on perceptually related sound events.

Bibliography

[1] N. M. Laird A. P. Dempster and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[2] J. J. Aucouturier, B. Defreville, and F. Pachet. The bag-of-frame approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. *Journal of the Acoustical Society of America*, 122(2):881–891, 2007.

[3] J. J. Aucouturier and F. Pachet. Improving timbre similarity: How high is the sky? *Journal of Negative Results in Speech and Audio Sciences*, 1 (In press)(1), 2004.

[4] P. Brossier. *Automatic annotation of musical audio for interactive systems*. PhD thesis, Centre for Digital music, Queen Mary University of London, 2006.

[5] M. A. Casey and A. Westner. Separation of mixed audio sources by independent subspace analysis. In *proceedings of International Computer Music Conference*, 2000.

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

[7] S. Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30:39–58, 2001.

[8] M. R. Every. Discriminating between pitched sources in music audio. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):267–277, 2008. ID: 5.

[9] D. FitzGerald. *Automatic Drum Transcription and Source Separation.* PhD thesis, Conservatory of Music and Drama, Dublin Institute of Technology., 2004.

[10] D. FitzGerald, B. Lawlor, and E. Coyle. Prior subspace analysis for drum transcription. In *AES 114th Convention*, 2003.

[11] D. Fitzgerald, R. Lawlor, and E. Coyle. Drum transcription using automatic grouping of events and prior subspace analysis. In E. Izquierdo, editor, *4th European Workshop on Image Analysis for Multimedia Interactive Services*, page 309, 2003. PT: B; CT: ; CY: APR 09-11, 2003; CL: Queen Mary,.

[12] D. Fitzgerald and J. Paulus. *Unpitched Percussion Transcription*, pages 131–162. In Klapuri and Davy [33], 1 edition, 2006.

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

[14] O. Gillet and G. Richard. Drum track transcription of polyphonic music using noise subspace projection. In *Proceedings of ISMIR*, pages 92–99, 2005.

[15] O. Gillet and G. Richard. Enst-drums: an extensive audio-visual database for drum. In *ISMIR*, pages 156–159, 2006.

[16] O. Gillet and G. Richard. Supervised and unsupervised sequence modelling for drum transcription. In *Proceedings of the 8th International Conference on Music Information Retrieval,ISMIR*, pages 219–224, 2007.

[17] O. Gillet and G. Richard. Transcription and separation of drum signals from polyphonic music. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(3):529–540, 2008.

[18] E. Gómez. *Tonal Description of Music Audio Signals*. PhD thesis, UPF, 2006.

[19] E. Gómez and P. Herrera. Comparative analysis of music recordings from western and non-western traditions by automatic tonal feature extraction. *Empirical Musicology Review*, In press, 2008.

[20] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. Rwc music database: Popular, classical and jazz music databases. In *ISMIR*, pages 287–288, 2002.

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

[22] M. A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proc. 17th International Conf. on Machine Learning*, pages 359–366. Morgan Kaufmann, San Francisco, CA, 2000.

[23] M. Helen and T. Virtanen. Separation of drums from polyphonic music using nonnegative matrix factorization and support vector machine. In *Proc.European Signal Processing Conf.(EUSIPCO)*, 2005.

[24] P. Herrera, X. Amatriain, E. Batlle, and X. Serra. Towards instrument segmentation for music content description:a critical review of instrument classification techniques. In *ISMIR*, page 9p., 2000.

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

[26] P. Herrera, A. Klapuri, and M. Davy. *Automatic Classification of Pitched Musical Instrument Sounds*, pages 163–200. In Klapuri and Davy [33], 1 edition, 2006.

[27] P. Herrera, G. Peeters, and S. Dubnov. Automatic classification of musical instrument sounds. *Journal of New Music Research*, 32(1):3–21, 2003.

[28] P. Herrera, V. Sandvold, and F. Gouyon. Percussion-related semantic descriptors of music audio files. In *Proceedings of 25th International AES Conference*, London, UK, 2004.

[29] P. Herrera, A. Yeterian, and F. Gouyon. Automatic classification of drum sounds: a comparison of feature selection methods and classification techniques. In *Proceedings of Second International Conference on Music and Artificial Intelligence*, Edinburgh, Scotland, 2002.

[30] P. O. Hoyer. Non-negative sparse coding. In *Proceedings of the 2002 12th IEEE Workshop on*, page 557, 2002.

[31] E. Aylòn i Pla. Automatic detection and classification of drum kit sounds. Master's thesis, 2006.

[32] A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *Acoustics, Speech, and Signal Processing, 1999. ICASSP '99. Proceedings., 1999 IEEE International Conference on*, volume 6, pages 3089–3092 o.6, 1999.

[33] A. Klapuri and M. Davy, editors. *Signal Processing Methods for Music Transcription.* Springer, New York, 1 edition, 2006.

[34] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13,*, pages 556–562, 2000.

[35] P. Leveau. Instrument-specific harmonic atoms for mid-level music representation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(1):116–128, 2008. ID: 1.

[36] D. Levitin. *This Is Your Brain On Music: The Science of a Human Obsession.* Dutton/Penguin, 2006.

[37] F. Moerchen, I. Mierswa, and A. Ultsch. Understandable models of music collections based on exhaustive feature generation with temporal statistics. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 882–891, New York, NY, USA, 2006. ACM.

[38] F. Morchen, A. Ultsch.and M. Thies, and I. Lohken. Modeling timbre distance with temporal statistics from polyphonic music. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(1):81–90, Jan. 2006.

[39] A. Camurri G. De Poli M. Leman D. Rocchesso X. Serra V. Välimäki G. Widmer N. Bernardini, R. Bresin. *A Roadmap for Sound and Music Computing.*

[40] J. P. Bello P. Brossier and M. D. Plumbley. Real-time temporal segmentation of note objects in music signals. In *Proceedings of the International Computer Music Conference (ICMC 2004)*, 2004.

[41] F. Pachet and P. Roy. Exploring billions of audio features. In *Proceedings of CBMI 07*, 2007,.

[42] J. Paulus and A. Klapuri. Combining temporal and spectral features in hmm-based drum transcription. In *Proc. of the 8th International Conference on Music Information Retrieval*, pages 225–228, Vienna, September 2007.

[43] J. Paulus and T. Virtanen. Drum transcription with non-negative spectrogram factorisation. In *Proceedings of 13. European Signal Processing Conference, EUSIPCO*, page 4 p., Antalya, Turkey, 4-8 September 2005 2005.

[44] G. Peeters. A large set of audio features for sound description (similarity and classification) in the cuidado project. Technical report, CUIDADO I.S.T. project at Ircam, 2004.

[45] R. Plomp and W. J. M. Levelt. Tonal consonance and critical bandwidth. *Journal Of The Acoustical Society Of America*, 38(4):548–&, 1965.

[46] R. J. Quinlan. C4.5: Programs for machine learning (morgan kaufmann series in machine learning). January 1993.

[47] J. Ricard. *Towards computational morphological description of sound.* PhD thesis, 2004.

[48] J. Ricard and P. Herrera. Morphological sound description computational model and usability evaluation. In *AES 116th*, 2004.

[49] T. D. Rossing. *Science of Percussion Instruments.* World Scientific, 2000.

[50] P. Roy, F. Pachet, and S. Krakowski. Improving the classification of percussive sounds with analytical features: a case study. In *Proceedings of Ismir 07*, pages 229–232, 2007.

[51] V. Sandvold. Percussion descriptors. a semantic approach to music information retrieval., cand. scient. thesis. Master's thesis, November 2004.

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

[53] E. D. Scheirer. *Music-listening systems.* PhD thesis, 2000. note: Supervisor-Barry L. Vercoe.

[54] W. A. Schloss. *On the Automatic Transcription of Percussive Music: From Acoustic Signal to High Level Analysis.* PhD thesis, Stanford University, May 1985.

[55] M. Slaney. Introduction to the special issue on music information retrieval. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):253–254, 2008. ID: 1.

[56] P. Smaragdis and J. Brown. Non-negative matrix factorization for polyphonic music transcription. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 177–180, 2003.

[57] D. Van Steelant, K. Tanghe, S. Degroeve, B. De Baets, M. Leman, and J. P. Martens. Support vector machines for bass and snare drum recognition. In *Proceedings of the annual machine learning conference of Belgium and The Netherlands*, 2004.

[58] S. Streich. *Music Complexity a multi-faceted description of audio content.* PhD thesis, UPF, 2007.

[59] S. Streich. and P. Herrera. Detrended fluctuation analysis of music signals danceability estimation and further semantic characterization. In *AES 118th*, 2005.

[60] Y. Sun and J. Li. Iterative relief for feature weighting. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 913–920, New York, NY, USA, 2006. ACM.

[61] K. Tanghe, S. Degroeve, and B. De Baets. An algorithm for detecting and labeling drum events in polyphonic music. In *Proc.of First Annual Music Information Retrieval Evaluation eXchange, London, UK, Sept*, 2005.

[62] K. Tanghe, M. Lesaffre, S. Degroeve, M. Leman, B. De Baets, and J. Martens. Collecting ground truth annotations for drum detection in polyphonic music. In *ISMIR 2005*, pages 50–57, 2005.

[63] A. Tindale, A. Kapur, and I. Fujinaga. Towards timbre recognition of percussive sounds. In *ICMC 2004: International Computer Music Conference*, pages 592–595. University of Michigan, Ann Arbor: International Computer Music Conference, 2004.

[64] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.

[65] E. M. von Hornbostel and C. Sachs. Classification of musical instruments: Translated from the original german by anthony baines and klaus p. wachsmann. *The Galpin Society Journal*, 14:3–29, 1961.

[66] A. Webb. *Statistical Pattern Recognition.* Number ISBN 0470845139. 2nd edition, 2002.

[67] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowl. Inf. Syst.*, 14(1):1–37, 2007.

[68] K. Yoshii, M. Goto, and H. G. Okuno. Adamast: A drum sound recognizer based on adaptation and matching of spectrogram. In *Proceedings of the 2nd*

Music Information Retrieval Evaluation eXchange (MIREX 2005), September 2005.

[69] 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. *Audio, Speech and Language Processing, IEEE Transactions on*, 15(1):333–345, January 2007.

Appendix A

Appendices per chapter

A.1 Chapter 3 appendix

Table A.1: Common label dictionary for database compatibilization.

Original labels				
MAMI	**ENST_wet**	**Sandvold's**	**Adopted Label**	**Description**
BD	bd	Kick, Kick+Cymbal	bd	Bass drum
SD	sd	Snare, Snare+Cymbal	sd	Snare drum
OH, CH	chh, ohh	—-	hh	Hi-hat
OH, CH, RC, CC, SC	chh, ohh, rc, ch, cr, c, spl	Cymbal, Kick+Cymbal, Snare+Cymbal	cy	Cymbal

Table A.1: Original and adopted labels for database compatibility.

A.2 Chapter 4 appendix:

Instances (after equalization) from MIX (ENST_wet+MAMI) database can be seen in table A.2. Figure A.1 shows the genre distribution within this new database

Instrument	**Instances**
bass_drum	4,011
snare	2,857
hi-hat	5,854
cymbals	5,254

Table A.2: MIX database instances (after equalization).

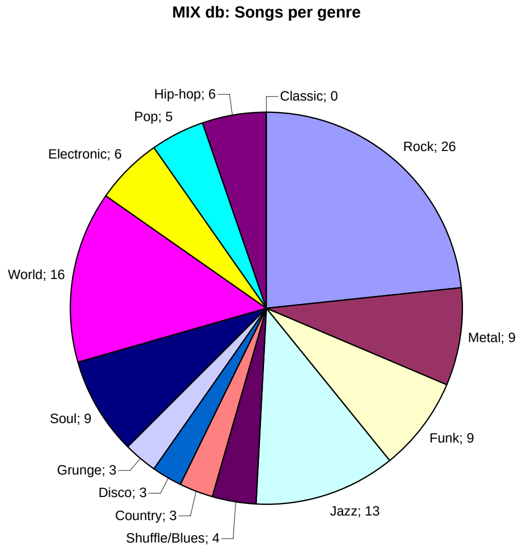


Figure A.1: MIX database: genre distribution.

A.3 Chapter 5 appendix

Selected features from 90%MIX_BOF+TDESC

- **Bass drums:**

spectral-pitch-salience-centroid barkbands-slope-1 barkbands-slope-2 barkbands-skewness-2 barkbands-skewness-3 barkbands-decay-2 barkbands-max-0 barkbands-max-2 barkbands-max-5 barkbands-norm-decay-1 barkbands-norm-decay-25 barkbands-t-skewness-1 barkbands-t-skewness-2 barkbands-norm-attack-0 barkbands-norm-attack-1 barkbands-norm-attack-2 barkbands-centroid-26 barkbands-var-0 barkbands-var-1 barkbands-var-2 barkbands-t-kurtosis-0 barkbands-t-kurtosis-1 barkbands-kurtosis-2 spectral-energy-slope-0 spectral-flux-slope-0 spectral-flux-max-norm-pos-0 spectral-flux-min-0 spectral-flux-centroid-0 spectral-flux-t-skewness-0 spectral-flux-mean-0 spectral-flatness-db-decay-0 spectral-energyband-low-norm-attack-0 spectral-energyband-low-t-skewness-0 barkbands-ratio-slope-2 barkbands-ratio-slope-19 barkbands-ratio-slope-21 barkbands-ratio-attack-0 barkbands-ratio-decay-1 barkbands-ratio-decay-21 barkbands-ratio-var-0 barkbands-ratio-var-1 barkbands-ratio-var-2 barkbands-ratio-max-0 barkbands-ratio-max-4 barkbands-ratio-norm-attack-0 barkbands-ratio-norm-attack-1 barkbands-ratio-t-skewness-1 MFCC-max-4 MFCC-max-7 MFCC-max-8 MFCC-norm-attack-11 spectral-rms-kurtosis-0 hpcp-var-21 hpcp-var-29 hpcp-var-30 hpcp-centroid-0.

- **Snare drums:**

temporal-lpc-var-8 temporal-lpc-var-9 temporal-lpc-max-9 temporal-lpc-skewness-1 temporal-lpc-min-6 barkbands-slope-4 barkbands-slope-5 barkbands-slope-7 barkbands-slope-9 barkbands-slope-14 barkbands-slope-19 barkbands-skewness-4 barkbands-skewness-11 barkbands-skewness-22 barkbands-min-22 barkbands-min-23 barkbands-min-25 barkbands-max-4 barkbands-max-5 barkbands-decay-4 barkbands-norm-attack-4 barkbands-t-skewness-4 barkbands-attack-4 barkbands-centroid-4 barkbands-var-4 barkbands-var-5 barkbands-t-kurtosis-4 barkbands-t-kurtosis-20 barkbands-t-kurtosis-22 barkbands-kurtosis-4 barkbands-kurtosis-5 spectral-pitch-slope-0 spectral-flatness-db-min-0 barkbands-ratio-slope-4 barkbands-ratio-max-norm-pos-4 barkbands-ratio-min-4 barkbands-ratio-max-4 barkbands-ratio-max-5 barkbands-ratio-max-12 barkbands-ratio-decay-4 barkbands-ratio-norm-attack-4 barkbands-ratio-norm-attack-21 barkbands-ratio-min-norm-pos-1 barkbands-ratio-min-norm-pos-4 barkbands-ratio-attack-4 barkbands-ratio-attack-22 barkbands-ratio-centroid-2 barkbands-ratio-centroid-4 barkbands-ratio-t-skewness-4 barkbands-ratio-var-3 barkbands-ratio-var-4 barkbands-ratio-var-5 barkbands-ratio-mean-4 barkbands-ratio-mean-21 barkbands-ratio-mean-22 MFCC-slope-0 MFCC-min-6 MFCC-min-8 MFCC-var-2 MFCC-max-4 MFCC-max-6 MFCC-max-8 MFCC-min-norm-pos-0 MFCC-mean-6 spectral-energyband-middle-low-slope-0 spectral-energyband-middle-low-skewness-0 spectral-energyband-middle-low-centroid-0 spectral-energyband-middle-low-t-skewness-0 spectral-hfc-t-kurtosis-0 MFCC-ratio-max-4 MFCC-ratio-attack-1 MFCC-ratio-t-kurtosis-1 hpcp-slope-3 hpcp-t-kurtosis-24 hpcp-min-0 hpcp-min-13 hpcp-mean-22.

- **Hi-hats:**

temporal-lpc-slope-2 temporal-lpc-skewness-3 temporal-lpc-centroid-4 temporal-lpc-centroid-6 temporal-lpc-min-4 temporal-lpc-min-6 temporal-lpc-max-7 temporal-lpc-t-kurtosis-3 temporal-lpc-t-kurtosis-4 temporal-lpc-mean-9 barkbands-slope-25 barkbands-slope-26 barkbands-max-norm-pos-26 barkbands-decay-26 barkbands-norm-decay-26 barkbands-norm-attack-23 barkbands-min-norm-pos-26 barkbands-attack-25 barkbands-centroid-26 barkbands-t-skewness-26 barkbands-var-26 spectral-skewness-mean-0 spectral-flux-t-skewness-0 spectral-flatness-db-min-0 barkbands-ratio-slope-26 barkbands-ratio-max-norm-pos-26 barkbands-ratio-min-19 barkbands-ratio-max-0 barkbands-ratio-max-2 barkbands-ratio-max-9 barkbands-ratio-min-norm-pos-0 barkbands-ratio-min-norm-pos-26 barkbands-ratio-centroid-26 barkbands-ratio-t-skewness-26 barkbands-ratio-t-kurtosis-26 MFCC-slope-3 MFCC-min-3 MFCC-ratio-norm-attack-1.

- **Cymbals:**

temporal-lpc-slope-3 temporal-lpc-slope-4 temporal-lpc-slope-7 temporal-lpc-skewness-1 temporal-lpc-norm-attack-7 temporal-lpc-t-kurtosis-3 temporal-lpc-t-kurtosis-4 temporal-lpc-t-skewness-2 temporal-lpc-centroid-3 temporal-lpc-centroid-4 temporal-lpc-centroid-5 temporal-lpc-centroid-6 temporal-lpc-mean-6 temporal-lpc-mean-8 MFCC-slope-0 MFCC-min-3 MFCC-mean-8 barkbands-slope-25 barkbands-slope-26 barkbands-max-25 barkbands-min-norm-pos-25 barkbands-min-norm-pos-26 barkbands-max-norm-pos-26 barkbands-decay-25 barkbands-decay-26 barkbands-norm-decay-25 barkbands-t-skewness-11 barkbands-t-skewness-23 barkbands-t-skewness-24 barkbands-t-skewness-25 barkbands-t-skewness-26 barkbands-centroid-26 barkbands-var-24 barkbands-var-25 barkbands-var-26 barkbands-mean-25 spectral-pitch-salience-max-0 spectral-rms-norm-attack-0 spectral-pitch-instantaneous-confidence-centroid-0 spectral-flux-slope-0 spectral-flux-t-skewness-0 barkbands-ratio-slope-26 barkbands-ratio-min-norm-pos-0 barkbands-ratio-min-norm-pos-26 barkbands-ratio-max-norm-pos-26 barkbands-ratio-min-19 barkbands-ratio-attack-26 barkbands-ratio-centroid-26 spectral-dissonance-norm-decay-0 spectral-crest-mean-0 hpcp-centroid-34.

A.4 Chapter 7 appendix:

A.4.1 Genre classification

Electronic:

Decision tree for “electronic” sub-genre classification (figure A.2)

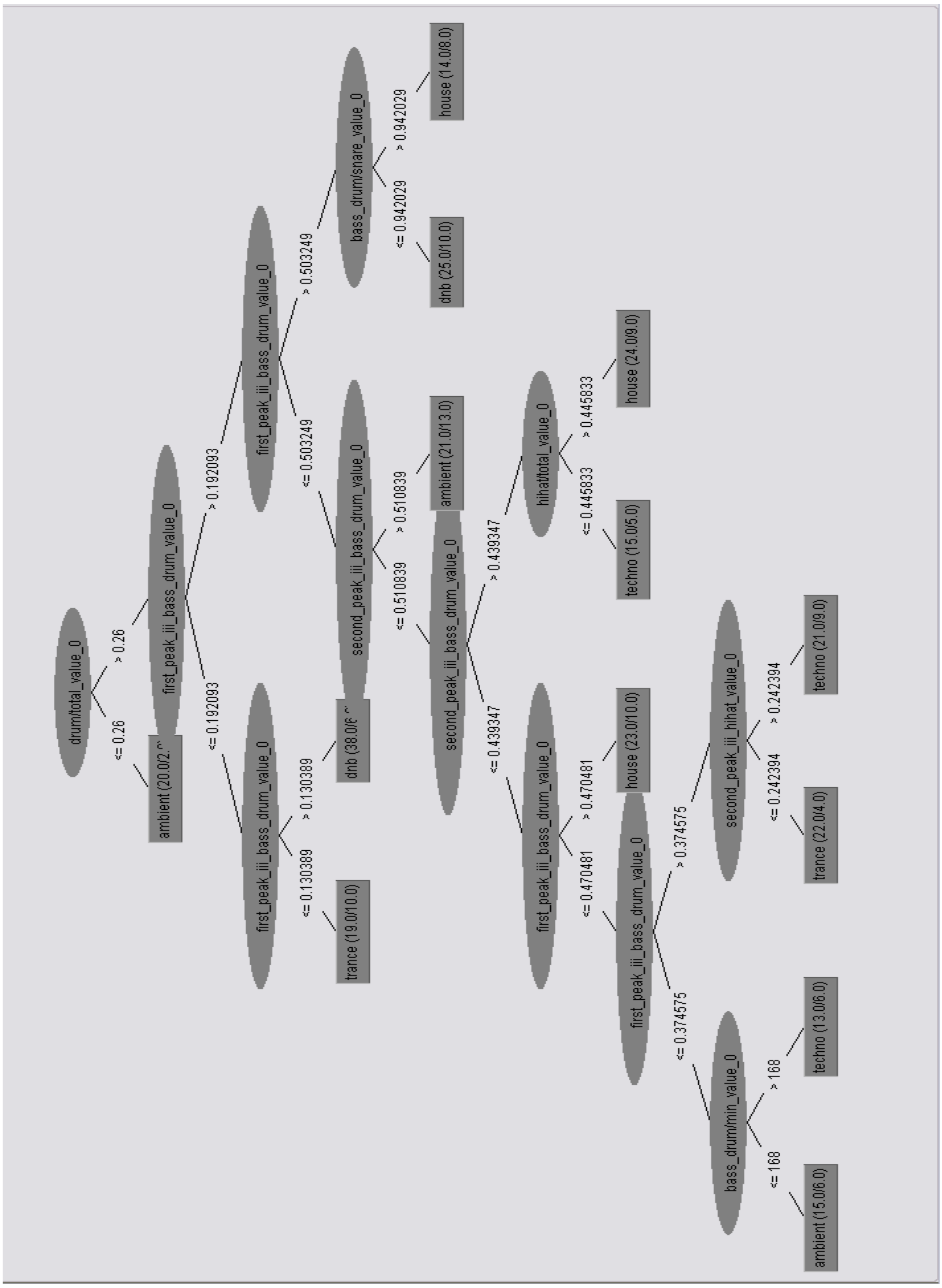


Figure A.2: Decision tree for "electronic" sub-genre classification using percussion descriptors only.