POLITECNICO DI MILANO
Facoltà di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria e Design del suono
Dipartimento di Elettronica e Informazione

Automatic Drum Trascription based on joint Prior–Subspace and Feature–Based Analysis

Supervisor: Prof. Augusto Sarti
Assistant supervisor: Dr. Massimiliano Zanoni

Master graduation thesis by:
Dario Giubileo, ID 721604
Federico Maffei, ID 720012

Academic Year 2009-2010

POLITECNICO DI MILANO
Facoltà di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria e Design del suono
Dipartimento di Elettronica e Informazione

Automatic Drum Trascription based on joint Prior–Subspace and Feature–Based Analysis

Relatore: Prof. Augusto Sarti
Correlatore: Dr. Massimiliano Zanoni

Tesi di Laurea di:
Dario Giubileo, matricola 721604
Federico Maffei, matricola 720012

Anno Accademico 2009-2010

To our beloved
Music.

Abstract

Due to the large diffusion of internet and of multimedia digital formats, the way people deal with music has been radically changing over the last few years: what is becoming more and more important for many applications in this field is the effectiveness in describing, retrieving and classifying informations able to distinguish a specific musical content over a countless number of others.
In the last few years, research has found that much of that kind of information lies on the rhythmic characterization of a music excerpt and has come to great achievements towards the challenging goal of transcribing percussive elements from the complex mixture of different sounds.
Many possible approaches have been developed so far, often combining source separation methods with tempo extraction techniques and classification based on audio features and statistics.
The piece of work presented in this thesis consists of an automatic drum transcription algorithm, developed as an improvement and an enhancement of the technique known as Prior Subspace Analysis (PSA).
Our efforts aim at increasing the effectiveness of the transcription creating a consistent training set, then adding an error correction through the analysis of the features of each drum part, and finally extending the capabilities of the algorithm to the recognition of tom drums, while mainly kick and snare drum were analysed in past studies.
These implementation steps have been followed by a phase during which our system has been tested on a number of polyphonic music recordings, measuring its performance by the means of standard parameters like precision, recall and f–measure.
The obtained results are encouraging: transcription returned an f–measure up to almost 90%, showing an improvement of about 10% over the original PSA algorithm.

Sommario

A causa della grande diffusione di internet e della digitalizzazione dei formati multimediali, il modo in cui le persone interagiscono con la musica è cambiato radicalmente negli ultimi anni: l'obiettivo di maggior importanza per le nuove applicazioni in questo campo è divenuto l'efficacia nell'estrarre, descrivere e classificare le informazioni utili a distinguere, tra un enorme numero di possibili candidati, uno specifico contenuto musicale.
Negli ultimi anni la ricerca in questo ambito ha dimostrato come si possa estrarre una notevole quantità di informazioni su di un contenuto musicale in esame attraverso l'analisi delle sue caratteristiche ritmiche, e ha raggiunto ottimi risultati nella trascrizione automatica delle occorrenze di suoni percussivi partendo da complessi *mix* audio composti da suoni differenti.
Sono stati proposti molti possibili approcci alla soluzione di questo problema di trascrizione, facendo sovente uso di algoritmi di separazione delle fonti sonore, combinati con tecniche di estrazione della scansione temporale e con metodi di classificazione basati sull'estrazione di *audio features*.
Il risultato del lavoro di ricerca presentato in questa tesi è un metodo di trascrizione automatica della batteria, sviluppato come un'estensione ed un miglioramento della tecnica conosciuta come *Prior Subspace Analysis*.
Lo sforzo progettuale è stato mirato ad aumentare l'efficacia della trascrizione, attraverso la costruzione di un *dataset* adatto alla fase di *training*, l'aggiunta un sistema di correzione per gli errori di trascrizione tramite analisi delle *features* di ogni componente della batteria preso in esame, e inserendo inoltre un'estensione dell'algoritmo per il riconoscimento dei tom–tom, laddove fino ad ora il *testing* era stato limitato alla grancassa ed al rullante.
Allo sviluppo sopra descritto è seguita una fase di verifica, in cui il sistema è stato testato su una serie di brani di musica moderna, valutandone la *performance* tramite parametri standard come *precision*, *recall* e *f–measure*.
I risultati ottenuti sono stati incoraggianti: il testing della trascrizione è stato caratterizzato da valori di *f–measure* fino al 90%, mostrando un miglioramento di circa il 10% rispetto all'algoritmo *PSA* originale.

Acknowledgements

We would like to thank:
Professor Augusto Sarti for the opportunity to accomplish this work, for his precious suggestions and help.
Dott. Massimiliano Zanoni, for the invaluable supervision, kindness and knowledge. He has been a constant source of new ideas and support, sometimes more a friend than a supervisor.
All our colleagues from the Sound Design courses, in particular our mates Dave, Eros, James, and many more.
All our friends in Ancona, Milano, and everywhere: you know who you are.

Dario would like to thank:
Federico: you accepted me joining this thesis even if you already started it alone by yourself. A friend is the best colleague I can imagine.
My family: Carlo, Giovanna, Elisa and Margherita. Throughout these years you provided me a constant support living everyday at my side. Priceless!
P. Fabio Sem, for wisely convincing me in starting this master degree study track.
All the staff at Nuova Audio Musicmedia, in particular Giacomo Lampugnani: you are teaching me how to play bass guitar and how to make my life better at the same time.

Federico would like to thank:
Dario: nice job!
My family: Claudio, Stella, Alessandra and Andrea. You are always there when I am in need. Thanks.

Contents

Abstract **I**

Sommario **III**

Acknowledgements **V**

1 Introduction **5**

2 State of the art **9**

2.1 Event-based drum transcription 11

2.1.1 Template matching algorithms 11

2.1.2 Evolution to Instrument Model Adaptation 12

2.1.3 Classification with Feature Extraction 15

2.2 Source Separation Techniques 16

2.2.1 From ISA to PSA 17

2.2.2 Non-Negative Matrix Factorization 19

2.3 Musicological Modelling 19

2.3.1 High-Level Algorithms 20

2.4 Conclusions 22

3 Theoretical background **23**

3.1 Principal Component Analysis 23

3.2 Independent Component Analysis 25

3.2.1 FastICA algorithm 27

3.3 Independent Subspace Analysis 28

3.4 Prior Subspace Analysis 32

3.4.1 Comparison with ISA 33

3.4.2 K-means clustering 33

3.5 Audio features analysis 34

3.5.1 Spectral centroid 34

3.5.2 Spectral skewness 34

3.5.3 Spectral kurtosis . . . 34
3.5.4 Spectral rolloff . . . 34
3.5.5 Spectral roughness . . . 35
3.5.6 Spectral irregularity . . . 36
3.5.7 Spectral flux . . . 36
3.5.8 Noise likeness . . . 36
3.5.9 Inharmonicity ratio . . . 37
3.6 Gaussian Mixture Models . . . 37
3.6.1 Maximum likelihood . . . 38
3.6.2 The Figueredo-Jain algorithm . . . 39

4 Methodology applied to drum transcription 41
4.1 Dataset creation . . . 41
4.2 Training step . . . 44
4.2.1 Prior subspace calculation . . . 44
4.2.2 Audio feature analysis . . . 46
4.2.3 Considerations on the analysed set of features . . . 51
4.3 Testing step . . . 52
4.3.1 PSA stage . . . 53
4.3.2 Error correction layer . . . 55
4.3.3 Tatum extraction and quantization . . . 56

5 Evaluation of the results 59
5.1 Testing the system . . . 59
5.2 Results . . . 61
5.3 Evaluation and comments . . . 66
5.3.1 Transcription results: kick drums . . . 66
5.3.2 Transcription results: snare drums . . . 66
5.3.3 Transcription results: tom drums . . . 67

6 Conclusions and future developments 69

A Training dataset details 71
A.1 *BFD2* drum pieces listing . . . 71
A.1.1 Kick drums . . . 71
A.1.2 Snare drums . . . 72
A.1.3 Tom–tom drums . . . 73
A.2 *Addictive Drums* drum pieces listing . . . 74
A.2.1 Kick drums . . . 74
A.2.2 Snare drums . . . 74
A.2.3 Tom–tom drums . . . 75

List of Figures

2.1 A basic drum kit: kick drum (1), floor tom(2), snare (3), toms (4), hi–hat (5), crash and ride cymbals (6). 9
2.2 A snare drum template pattern used for matching, represented as the power distribution of its frequency components. [19] . . 12
2.3 Classification accuracy of localized models represented against the decreasing proportions of the general dataset used 17
2.4 Separation of a kick drum track and a snare drum track from a mixed sound signal 18
2.5 A diagram that represents the idea behind conventional and periodic N-grams. Each drum piece played is represented by a letter (B = kick drum, S = snare drum, H = hi-hat, T = tom and C = cymbal). The horizontal arrow indicates a normal trigram prediction, and the vertical arrow is a periodic trigram prediction. L is set to be the length of the musical measure (eight in this case). [15] 21

3.1 An example of spectral rolloff. 35
3.2 Sensory dissonance and spectral roughness (considering two tones only). 35
3.3 Original spectrum (solid) and model spectrum (dashed) of percussive component. 37
3.4 An example surface of a two-dimensional Gaussian mixture PDF with three components 38

4.1 A MIDI sequence used to generate the drum samples for the training phase. 42
4.2 The overall block schema for the training step. 43
4.3 A visual representation of the prior subspaces of kick drum (blue), tom drum (green) and snare drum (red). 44
4.4 The spectral skewness computed on samples of the training set. 45
4.5 The spectral kurtosis computed on samples of the training set. 46

4.6 The spectral flux computed on samples of the training set. . . 47
4.7 The attack and release spectral envelopes computed on segmented samples of the training set. 48
4.8 Global, attack and release spectral centroids computed on segmented samples of the training set. 49
4.9 The spectral roll–off computed on samples of the training set. 50
4.10 The spectral irregularity computed on samples of the training set. 51
4.11 The spectral roughness computed on samples of the training set. 52
4.12 The inharmonicity ratio computed on samples of the training set. 53
4.13 The noise–likeness computed on samples of the training set. . 54
4.14 The overall schema used in our drum transcription algorithm. 55
4.15 Tatum grid vs. beat times in musical piece; the tatum grid represents the set of all possible note onsets. 57

List of Tables

5.1 Precision, recall and f–measure obtained by PSA without Error Correction layer 62
5.2 Precision, recall and f–measure obtained by PSA with Error Correction layer. 64
5.3 Precision, recall and f–measure obtained on tom–toms with rule–based and GMM–based Error Correction layer. 65
5.4 Overall results of the testing phase for each drum type. 66

Chapter 1

Introduction

In every moment of daily life, our auditory and neurological system performs some kind of sound analysis and classification, even if it is done at a subconscious level. Humans are able to focus on the source of major interest in a sonic mixture, that is classified according to mechanisms that are still almost unknown. The reproduction of this human capability is part of the objects of research in the music information retrieval (MIR) field.

MIR is the interdisciplinary science of retrieving information from music. It involves signal processing techniques, computational methods for clustering and modelling, machine learning, psychological and perceptual aspects, music analysis and its representation.

One of the main reasons why a further development in this topics is needed is that lately the volume of musical production is highly increased, since the technical tools required for it are getting cheaper and more affordable. We are witnessing a progressive change from the concept of "consumer" opposed to "producer" of contents to the merged one of "prosumer", as a greater number of enthusiasts can now both contribute and access to contents without a sharp distinction.

Thus, the ability to collect ever–increasing number of audio elements and organize them efficiently in a database is becoming more and more crucial in the digital era: software applications and websites, like *iTunes* and *Last.fm* just to mention two famous ones, are countless.

Algorithms for analysis and classification of a music excerpt are becoming vital, in a world where the production and the commercialization of audio material is so widespread, but instruments to create categories and metadata able to describe the content have not been developed with the same speed and reliability.

To date, this classification task has been carried out mainly in a context–

based manner, with tags subjectively associated to contents even by users themselves. Conversely, greater dimensions and stricter reliability require the usage of content–based automated criteria, whose definition is the actual frontier of research. The interest of the scientific community in this area has consequently grown a lot.

More in detail, systems developed in this sense can lead to correct pitch detection, beat tracking, chord change detection and so on. The retrieval of data related to the melodic and rhythmic content of a song are two fields that are often studied separately, since the first involves the need to transcribe pitched instruments, while the second is mainly about recognizing and distinguishing the appearance of un-pitched sounds, that very often belong to the drum sounds family.

A transcription can be defined as a representation of a given musical performance, given from several perspectives, but mainly regarding the instruments involved and the notes played by each one (tracking their pitch and amplitude).

The analysis of rhythmic events, in particular of drums, is our task. Building a transcription algorithm that focuses on the percussive parts means to put the retrieval effort both on detecting onset times of individual rhythmic events, and on detecting which drum instrument is played at the detected onset time. The result is a time–line of the events, that is retained for each drum instrument analysed.

A correct drum transcription, executed on a dataset of songs, can thus be useful for searching engines based on "query–by–tapping" (search for a song that has a specific rhythmic line). Classification of music content can also be performed, distinguishing song genres by beats per minute, drumming style and so on. Moreover, the automatic creation of music sheets and diagrams based on the transcription of a drum could be used as an aid for music students, that would have a powerful interface to check their accuracy and improve their skills.

The goal of this thesis is to provide an improvement of the existing drum transcription methodology known as Prior Subspace Analysis (PSA). This algorithm is based on the creation of prior subspaces, which are a description in the frequency domain of a class of sounds, in our case kick drums, snare drums and tom–toms. It is noteworthy that the latter drum instrument has rarely been took into account by other works in literature.

In order to create the above mentioned subspaces, we built a dataset of drum sounds from scratch. Then we combined an implementation of the PSA algorithm with a research on audio features on the same dataset, defining an error correction layer that helps the system to discard a number of false–

positive detections and consistently enhances the overall performance of the system.
The achieved results have been evaluated by means of standard parameters like precision, recall and f–measure, showing that the error correction of our algorithm increases the performances of the original PSA by about 10%, achieving a f–measure up to almost 90%.

Overview of the thesis The thesis is organised as follows.
In chapter 2, a list of drum transcription related works is presented, offering a broad view on the state of the art in this field of research.
In chapter 3 we will provide the necessary theoretical background of PSA and audio feature extraction, an overview that is useful to fully understand how our system works.
Then, chapter 4 explains our methodology in details, including observations and remarks on the analysed audio features, while a detailed description of the results achieved is contained in chapter 5.
Finally, the last chapter (6) is devoted to conclusions, also providing ideas for future improvements and evolutions.
Appendix A contains further details on the created dataset.

Chapter 2

State of the art

The drum has always been (and perhaps will always be) a fundamental component in the vast world of music instruments: the majority of scientific research to date has focused on the unpitched percussion instruments found in Western popular music, with a particular focus on the drums found in the standard rock/pop drum kit, namely snare drum, kick drum (also known as a bass drum), tom-toms, hi-hats, and cymbals [15].

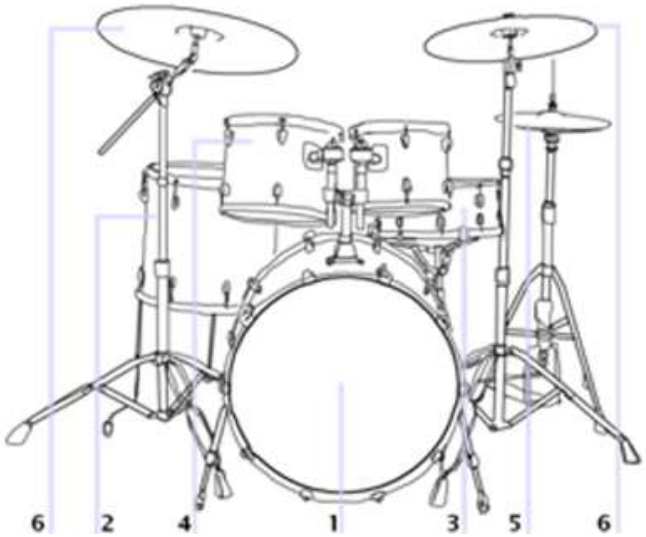

Figure 2.1: A basic drum kit: kick drum (1), floor tom(2), snare (3), toms (4), hi–hat (5), crash and ride cymbals (6).

According to the Hornbostel-Sachs classification [34] the drum parts mentioned in this chapter can be divided into two main types:

- membranophones: include drums such as snare drums, kick drums and

tom drums, that typically consist of a membrane or skin stretched across a frame;

- idiophones: including instruments such as hi-hats and cymbals, are typically rigid bodies, such as a metal plate.

In modern Western music, deeply characterized by pop and rock influences, the sound of drums has always been associated to the idea of rhythm and time division in a song.
In this scenario, the ability to determine the presence or absence of drum instruments plays an important role for music classification [21], providing means of analysis at an higher level than signal processing, and being crucial for further steps like metrical analysis, rhythm recognition and in general any task related to categorization of mixed audio tracks [23].
The models discussed in this chapter are those which have been tested over polyphonic music excerpts, where "polyphonic" may be related to a signal composed by a mixture of drums instruments only, or by a mixture of drums and pitched instruments: this last one is the frontier of research and the main focus of this thesis too, however it will be made clear for every system described to help disambiguation.
The first piece of work in the field of polyphonic music transcription can be traced back to 1975, when J.A. Moorer built a system for transcribing duets for his PhD thesis [36].
In the years, research on the polyphonic music transcription field has progressively followed split paths between transcription techniques aimed at the analysis of the parts played by pitched instruments (like piano, guitar, etc.) and the ones referring to percussive ones (in main instance, drums).
The last decade has seen a significant improvement in the field of drum transcription, trying more often to get a robust transcription of small sets of drums rather than having large drum sets transcribed with uncertain results [12].
It is of vital importance, to achieve understanding of the mechanism behind drum transcription, to define the meaning of low-level and high-level transcription techniques.
In general, the approaches that are limited to audio signal processing, without a further model of improvement of the results, are called low-level, while all the algorithms making use of an error correction layer, that has an action over the results produced by a transcription, are called high-level. The algorithms belonging to the first of these two macro-areas can be divided between event–based and source separation–based systems [15]; definition and examples of these methods are presented respectively in 2.1 and 2.2.

The last section of the chapter (2.3) takes into account high–level methods, making use of a further layer of error correction and performance enhancement: some details on algorithms are then provided, concluding with the results achieved by each one of them.

2.1 Event-based drum transcription

Event-based drum transcription systems segment the input signal into chunks containing meaningful events and try to evaluate the contents of each segment through pattern recognition methods [15]. in general, the steps of an algorithm that performs this kind of transcription can be summarized as:

1. segment the input signal into chunks by either looking for supposed sound event onsets (here, is considered as the beginning of a sound) or using regular temporal grid (whose spacing is determined by the fastest rhythmic pulse present in the signal). This technique has been developed, among the others, by Bilmes [8] and Gouyon et al. [20].

2. extract a set of features from each segment (*e.g.* the spectral centroid, but this is specific to methodology applied) [21], [41];

3. classify the contents of each segment based on the extracted features [25], [43].;

4. collect the information retrieved to yield the overall transcription.

In the following subsections we present a brief analysis of transcription methods belonging to this category.

2.1.1 Template matching algorithms

The first template matching method to achieve polyphonic drums–only transcription has been introduced in 1994 by Goto et al. [19], that attempts to detect the onset times of each drum sound in a pop music drum performance, which is a sound mixture of nine kinds of drum instruments: the bass drum, snare drum, low tom, middle tom, high tom, closed hi-hat, open hi-hat, ride cymbal, and crash cymbal.
The templates are obtained from the spectrogram of samples from each drum type (an example is shown in 2.2), and the system estimates the presence of each drum computing a distance measure between templates and spectrum of the (framed) signal.

Later on [18], this algorithm has been improved, although limiting its scope to kick and snare drums and for beat–tracking task instead of drum transcription, by first identifying potential onsets and then creating a histogram of spectral peaks at these onset locations: the matching was then done comparing the onset's peak frequency in the signal with two specific frequency bands found to be most descriptive for those drums.

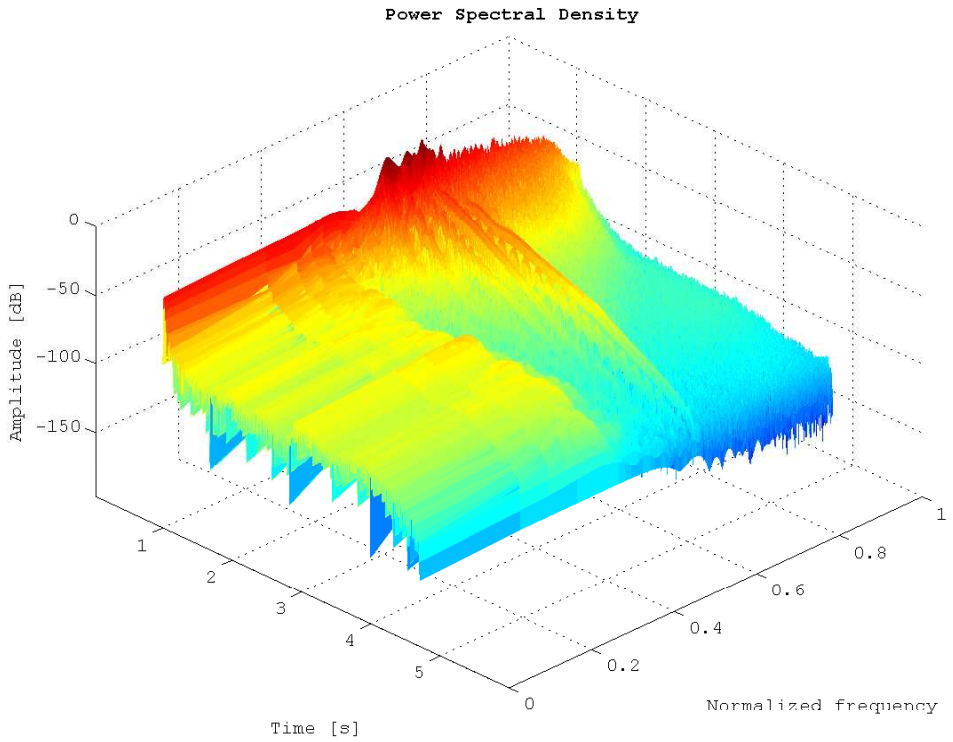

Figure 2.2: A snare drum template pattern used for matching, represented as the power distribution of its frequency components. [19]

Template matching algorithms evolved making the templates adaptive: the next section will give a brief overview on two of the most important among them.

2.1.2 Evolution to Instrument Model Adaptation

An adaptive template matching algorithm was proposed by Zils et al. [58]: the system aims to adapt a rough synthetic template to the sounds actually present in the input signal. This work considers a testing signal composed by a mixture of drums and pitched instruments, looking for kick and snare drums only.

The starting template consist of a simple percussive sound $I(t)$ (of length N_I), resulting from a filtered impulse (low-passed for kick drums and high-passed for snares). A correlation function $Cor(\delta)$ with the input signal $S(t)$ (of length N_S) is then computed, to find the occurrences of the template sound in the audio signal by means of amplitude peak detection:

$$Cor(\delta) = \sum_{t=1}^{N_I} S(t)I(t-\delta) \tag{2.1}$$

where $Cor(\delta)$ is defined for $\delta \in [1, N_S]$ with the introduction of a peak quality measure to filter out bad occurrences. In this first attempt, occurrences of peaks are often missed, creating a number of false negatives. The system is designed to adapt the templates, synthesizing a new percussive sound $newI(t)$ based on the results obtained by the first peak detection:

$$newI(t) = \frac{1}{2}\left[I(t) + \frac{1}{npeaks}\sum_{i=1}^{npeaks} S(peakposition(i)+t)\right] \tag{2.2}$$

with *npeaks* number of detected onset peaks. This process is repeated until convergence is reached, then a full drum track extraction is performed on the input signal, following two steps:

1. A first extraction based on the low-pitched sound provides the occurrences of the sound that is most similar to the bass drum template.

2. A second extraction based on the high-pitched sound provides the occurrences of the most important snare drum sound, that are not conflicting with the previous bass drum occurrences.

After the extraction, the zero-crossing rate (ZCR) is computed on the input signal. ZCR is an audio feature suggested by Gouyon [21] to discriminate kicks from snares (because ZCR is greater at higher frequencies, where typically snares are richer than drums): in this way a number of false positive detections given by previous correlation can be removed.
A testing phase has been carried on over a database of 100 songs, divided by transcription complexity in three levels: *Easily extractable*, with dominant and loud drum sounds, *Possibly extractable* with a balanced mix between drums and other instruments, and *Hardly extractable*, with low drum mixes or very unpredictable patterns. Some factors that lowered the overall performance of the algorithms were :

- A predictable bad performing in the *Hardly extractable* songs;

- A tendency of the snare to be confused with female vocals;
- The fact that two drum sounds overlapping always biases the detection response towards the lower pitched one;
- Particular cases in which the models is not fit to describe the actual drum sound that is played (*e.g.* when the snare drum sound is superimposed to a clap sound);
- Especially in the *Hardly extractable* cases, confusion with noisy components of the input signal, that would require further complex processing to get rid of.

The results were evaluated on a qualitative basis, achieving a correct transcription in more than 75 % of the test sample songs.
Another significant adaptive development of the template matching transcription can be found in the work of Yoshii et al. [56], [57] that proposed a template adaptation model, taking into account the fact that acoustic features of drum sounds vary with each musical piece and precise templates for them cannot be prepared in advance. In this case, the templates of drum sounds (hereby called "seed templates") are chosen as their spectrograms. This method is based on an iterative adaptation algorithm. At first, the onset candidates are roughly detected in the audio signal using the Rough Onset Detection, that takes as onsets the peaks of:

$$S(t) = \sum_{f=1}^{2048} F(f)Q(t,f) \tag{2.3}$$

where $F(f)$ is a filter with the frequency characteristics of bass or snare drum and $Q(t,f)$ is the time differential of the power spectrum of the input signal at frame t and frequency f, calculated by STFT with Hanning windows. Starting from each onset, a spectrum excerpt is extracted from the power spectrum. Then, by using all the spectrum excerpts and the seed template of each drum sound, the iterative algorithm goes through several stages to obtain the adapted template; according to the notation used in the referenced papers, it can be summarized as follows.

1. Calculate the Euclidean distances $D_{i,j}$ between a template T_i and the spectrogram excerpts P_j with:

$$D_{i,j} = \sqrt{\sum_{t=0}^{T-1}\sum_{k=0}^{K-1}\left[H_{LP}(k)(T_i(t,k) - P_j(t,k))\right]^2} \tag{2.4}$$

where T is the number of frames in the template and spectrogram excerpt, K is the number of frequency bins, and the response H_{LP} is used to attenuate high frequencies.

2. Order the spectrogram excerpts by their distances $D_{i,j}$ in ascending order.

3. Select the spectrum excerpts whose distance from the seed template (and in further steps to the adapted template) is smaller than a given threshold. The threshold is determined according to a priorly fixed ratio of the number of selected excerpts to the total number.

4. Calculate a new template spectrogram with

$$T_i(t,k) \leftarrow median_{j \in S}\{P_j(t,k)\} \tag{2.5}$$

5. Repeat until convergence is reached.

The use of the median operator is identified as a "Template Refinement" step, and represents an attempt to cancel the effect of the presence of pitched instruments on the drum templates, assuming that selected segments carry an almost invariant spectral information about drums while spurious components are different each time and can get reduced just by averaging.
The actual segment recognition phase is carried on using a template matching method [19]. The system was tested on a set of 10 songs, taken from the RWC database [35], at first enabling and then disabling the adaptive system described above. The adaptive part of the method granted an F–measure of 90% on kick drums , improving of a 23% over the "classic" method, and an overall f–measure of 88% on snares, with an improvement of 14%, showing that the use of adaptive techniques succeeds in significantly enhancing the template matching transcription results.

2.1.3 Classification with Feature Extraction

In the drum classification system implemented by Sandvold et al. in 2004 [43], the models used are based on a more complex description, that relies on a set of properly extracted features (*e.g.* percussivity, percussion profile and others), instead of relying on invariant time–frequency drum templates. This adaptive algorithm starts from the assumption that N occurrences of a drum sound are detected by an onset descriptor, proposing four subsequent steps:

1. The N obtained segments are classified with general instrument models (a set of previously computed feature values for every class);

2. The $M < N$ most reliable results are chosen;

3. Features are extracted from these M segments, proceeding with the creation of "localized" models;

4. The N segments are classified, this time using the ad-hoc models just developed.

For the general instrument models used in the first step, a total of 115 features is considered and analysed with a correlation-based algorithm, to perform dimensionality reduction to nearly 25 features per model. Classification is performed by means of decision trees. In the second step, selection of M best models is done parsing the set of models manually, by observing the best performances of classifiers chosen randomly between the candidates. The new locally-optimized classifiers are obtained performing a correlation based algorithm to retain the best features for the M sized set. In the last step, classification is performed again, this time with the "locally-optimized" models. The localized models reached an accuracy of respectively 95.06%, 93.1% and 89.17% for kicks, snares and cymbals, with an improvement of respectively 14.79%, 22.2% and 22.86% over the results of general model classification. Furthermore, the testing of the algorithm suggested that the accuracy of the localized features never drops under the accuracy of the general ones, even with a set of features reduced to 20% of the original. The relation between the two approaches is clearly shown in 2.3.

2.2 Source Separation Techniques

The source separation approach to polyphonic drum transcription is at the basis of most of the low-level state of the art algorithms developed to date. The basic concept behind source separation is that the polyphonic audio excerpt can be treated as a mixture of sound sources that can be separated to obtain basis functions that represent a single source, so that it is possible to perform onset detection separately on each one of them as in figure 2.4. Compared to the event-based transcription, this way of proceeding is a major advantage [15].
A number of techniques have been developed to achieve source separation. Among those: the independent subspace analysis (ISA) [10], the prior subspace analysis (PSA) [12], and the non-negative matrix factorization (NMF) [39]. Another noteworthy piece of work has been proposed by Virtanen, who described a source separation system based on Non-Negative Sparse Coding (NNSC) [52], using a power spectrogram as the input to the system. These

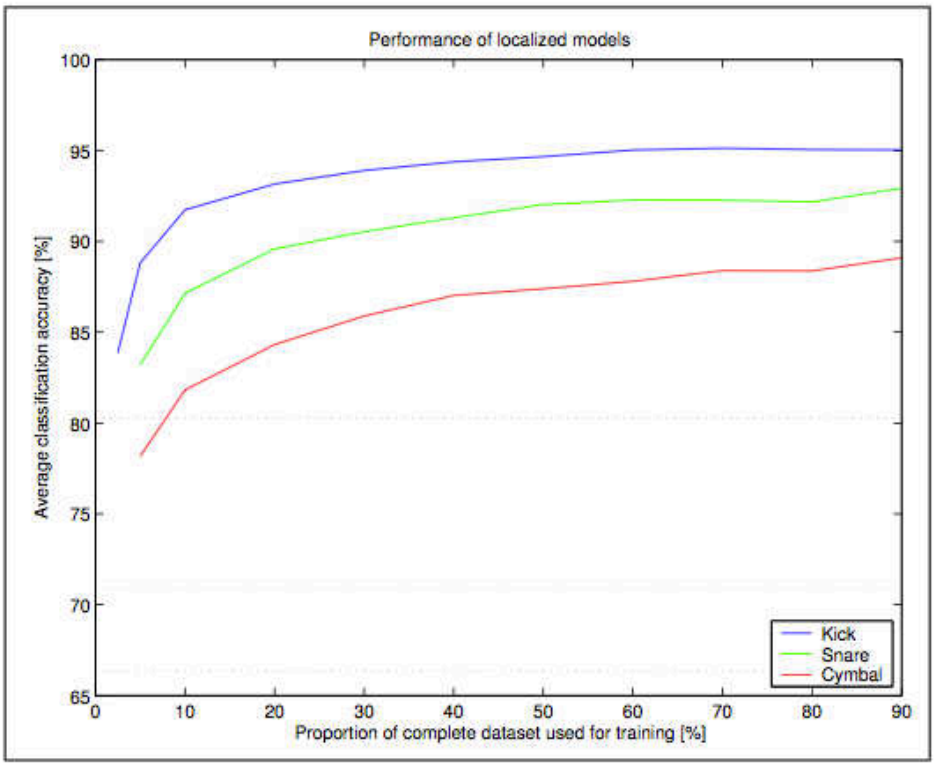

Figure 2.3: Classification accuracy of localized models represented against the decreasing proportions of the general dataset used

low-level techniques have been the starting point for a number of high-level transcription methods belonging to the state of the art of research. Further details about high-level systems will be given in 2.3.

2.2.1 From ISA to PSA

The first attempt to apply the Independent Subspace Analysis (ISA) to audio source separation was developed by Casey et al. [10]. This system obtains a blind source separation of an input mixture of sound sources, performing at first a Principal Component Analysis (PCA) on the spectrogram, reducing the dimensionality of the frequency lines, retaining the ones with the lower correlation, and then performing Independent Component Analysis (ICA) on them. The separated sources are then tracked by the similarity of the dynamic components after an analysis over small time steps of the input signal.

The ISA approach just introduced makes a fundamental assumption: the

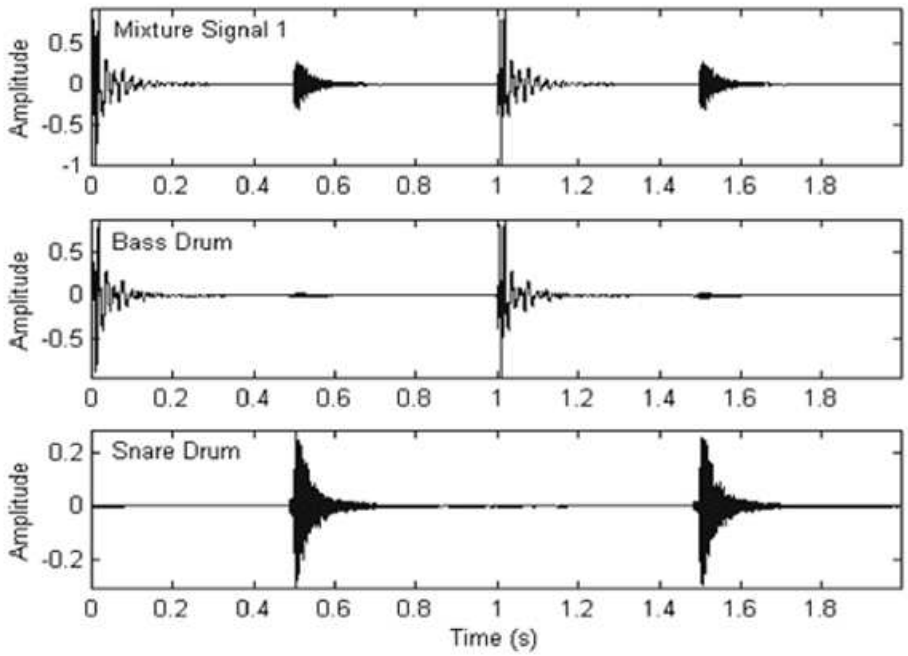

Figure 2.4: Separation of a kick drum track and a snare drum track from a mixed sound signal

use of invariant frequency basis functions means that no pitch changes are allowed over the course of individual spectrograms. This is valid for most drum sounds, where the pitch of the drum does not change from event to event, making this type of decomposition particularly suited for analysing percussive tracks in a polyphonic "drum-only" context.

The Prior Subspace Analysis system (PSA) [14], tries to overcome this limits and has been developed specifically for drum transcription in a polyphonic (drums and pitched instrument) context.

According to ISA, the number of components required for separation varies with the frequency and amplitude characteristics of the source sounds, and the ISA approach could not adequately predict the required number of components. To overcome this limitation, PSA assumes that there exist known prior frequency basis functions assumed to be a good approximation of the actual basis functions for the sources of interest.

Since this is the technique adopted in our work, more on this topic will be explained in next chapters.

2.2.2 Non-Negative Matrix Factorization

More recently, a different formulation of PSA has been proposed for the purposes of drum transcription based on using a Non-Negative Matrix Factorization (NMF) algorithm to calculate priorly fixed frequency basis functions [33]. The NMF algorithm estimates non-negative amplitude basis functions so that the reconstruction error of the model is minimized. This makes blind source separation more suitable to situations where prior knowledge of the sources present in the mixed input signal is unavailable. Virtanen et al. [39] implemented and tested NMF over studio-level drums-only tracks made of different kits and belonging to different genres, with a hit rate of 94%. No testing over mixed audio excerpts were performed.

An implementation of NMF, this time with a test over one jazz song, has also been proposed by Moreau et al. [37], with a precision of 89% for bass drums, 36%for snare drums, and 34% for hi hats.

The NMF algorithm has reached its best results when in 2009 Alves et al. [4] used multiple signals of the same drum source before mixdown, as usually available in recording studios, to tackle interferences between components and to get better accuracy in the onset detection. The testing phase has been carried on using the ENST database [16], reaching an f–measure of 94.7% with kick drums recognition, 77.6% with snares, and 83.0% with hats, showing an improvement over the single-channel NMF.

2.3 Musicological Modelling

The transcription systems presented in this overview until now concentrate only on low-level recognition, without any processing at a higher abstraction level. However, human beings for this task use a set of musicological concepts to classify and separate perceived musical sources, even at a subconscious level, mainly considering the relations between events (in particular from temporal perspective).

Here we use the term "musicological modelling" to refer to attempts to model the statistical dependencies of event sequences, adding an higher abstraction layer to enhance the results of any transcription algorithm. Usually, high-level processing consists of a sort of probabilistic modelling of the temporal relations between sound events. In practice, this requires that the low-level recognition results must also be presented as a probability or likelihood value.

2.3.1 High-Level Algorithms

One of the first attempts to improve the recognition ability by using higher-level musicological "language model" has been made by Paulus et al.[40], with a method based on conventional and periodic N-grams.

In the periodic N-gram, defining a word $w(n)$ as a set of symbols found in a sequence at time m, the units used to predict the probability of the presence of $w(n)$ are picked at multiples of interval L before $w(n)$, *i.e.* at time instances $n = [k - (N-1)L, \ldots, k - 2L, k - L]$.

The conventional N-gram is a special case of the previous where $L = 1$. In this system, drum sounds are classified into categories by defining a finite alphabet of symbols Σ which represents the drum categories.

In the work cited above, the alphabet was chosen to have size $S = 7$, with symbols representing bass drums, snare drums, hi-hats, cymbals, ride cymbals, tom toms, and other percussion instruments. Words are defined to be sequences of symbols, to represent a set of drum categories that are played simultaneously at a given instant of time. Silence is accounted as an empty word which does not contain any symbols. V is the vocabulary size, that is total number of possible words in the language: it can be calculated as $V = 2^S$.

The first step of the algorithm is quantizing musical time by finding the tatum of the incoming musical performance. The term tatum, or, time quantum, refers to the shortest timing value in a musical composition that is still more than incidentally encountered. After the tatum of a performance has been estimated, a grid of equidistant tatum pulses is aligned with the performance, and the position of each drum event is quantized accordingly. Then, a classification on each portion of signal contained between two grid points is performed through the use of GMM (Gaussian Mixture Models), with Mel–frequency cepstral coefficients (MFCCs) as features. After this step, a SVM (Support Vector Machine) is applied to distinguish silence from actual sound occurrences. At each segment, the low-level recognition produce likelihoods for different mixture events. The actual transcription is done by choosing the mixture having the highest likelihood at each grid point.

The results show an initially lacking transcription, with an error rate of 76%. The simplest N-gram, the baseline ($N = 1$), reached the 49%, while the periodic decagram ($N = 10$) reached a 41% and showed encouraging improvements.

A generalization of N-gram algorithm was provided in 2007 by Gillet et al. [17] that proposed a "supervised sequence modelling", by means of adapting dependencies at different scales, with a sequence model time support, or in

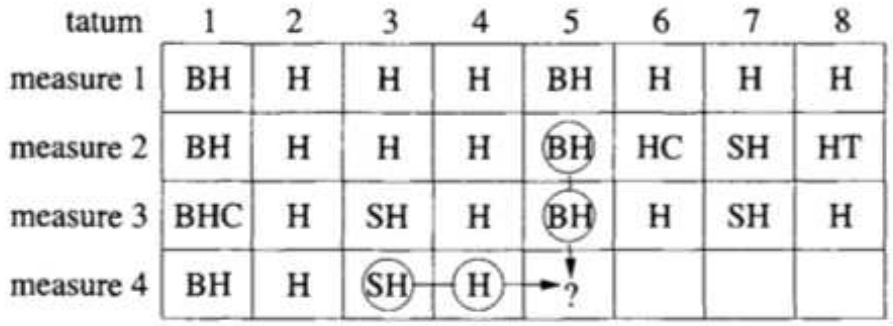

Figure 2.5: A diagram that represents the idea behind conventional and periodic N-grams. Each drum piece played is represented by a letter (B = kick drum, S = snare drum, H = hi-hat, T = tom and C = cymbal). The horizontal arrow indicates a normal trigram prediction, and the vertical arrow is a periodic trigram prediction. L is set to be the length of the musical measure (eight in this case). [15]

other words, the temporal levels at which the dependencies are considered. As such, their model allows the use of a time support T that achieves a trade-off between the drum pattern observation length and the number of probabilities to estimate. For instance, when the tatum corresponds to a sixteenth note, choosing T allows the model to learn dependencies between successive bars, beats, and tatum points, while keeping the overall complexity and the number of degrees of freedom of the model low. This generalized N-grams algorithm reached an accuracy of 82.6% with kick drums, 67.0% with snare drums and 82.9% with hi–hats.

Another high–level error correction framework has been created by Yoshii et al. [55]. This method has been built to work over the transcription results of the template matching algorithms described in 2.1.1. At first the framework performs a bottom–up processing, looking to extract patterns from detected onsets of drums, with a Template-matching algorithm, then subsequently correcting errors by tackling the periodicity of the drum patterns in a top–down processing phase. On the basis of the observation that the same drum patterns tend to be repeated, time points which deviate from the periodicity as error candidates are detected. Then each error candidate is examined to determine if it is an actual onset or not. The testing has been performed on 50 songs from RWC database, showing that this error correction method raises the f-measure from a 76.8% in absence of the framework to 81.1% with the framework enabled in the bass drum sound detection, and from 78.0% to 80.3% in the snare drum sound detection.

Finally, the work of Spich achieved a drum transcription result of about 90

% from a polyphonic (drums and pitched instruments) mixture in a pop–rock context, using PSA plus a musicological pattern recognition layer as described in [47].

2.4 Conclusions

This chapter provided an overview on the state of the art of the research in the field of automatic drum transcription from polyphonic music. All the methods presented have contributed to some extent to the development of the recognition and transcription of percussive instruments. But there are still open issues and a need for a further development in this matter. Part of our work is focused on this.

One main aspect to be improved is the lack of recognition of drum kit pieces other than kick drum, snare drum and hi-hats. In the high–level transcription field, the amount of past research is limited to a few attempts: they rely on a good performance of the lower levels, thus enhancing this one may result in an improvement of the whole system.

In the third chapter a theoretical background to one of the best performing low-level drum transcription methods will be given, mentioning all the signal processing algorithms and mathematical tools involved in the method that is the object of this dissertation.

Chapter 3

Theoretical background

As shown in the previous chapter, drum transcription in a polyphonic mixture with pitched instruments can be performed using many techniques: high level ones are very promising, but rely on efficient low level analysis approaches. Among those, source separation methods seems to be effective for this task: one of the most important is the Prior Subspace Analysis (PSA) [13].
The PSA algorithm considers each drum instrument as a source, trying to describe it through the creation of an invariant model in the frequency domain, called prior subspace, built before its usage for transcription. In this way, the separation in testing will no more be blind with respect to the sources, but pointed towards some specific kind of sounds, with a greater robustness against differences in amplitude. In the next chapter we will show our enhancement to the PSA methodology explained here.
This thesis is also focused on development of an error correction layer as a post–processing to PSA results in order to limit the number of false–positive detections: after an audio feature analysis research, we did that working on a rule–based system and we made an attempt with a Gaussian Mixture Model to be considered as a preliminary study for later development.
In this chapter, section 3.1 to 3.3 explains methods that yield to the derivation of PSA (section 3.4), followed by a brief explanation of the audio features used and the GMM theory.

3.1 Principal Component Analysis

Principal Component Analysis (PCA) transforms a set of correlated variables into a number of uncorrelated or orthogonal variables called principal components [31].

PCA has become a standard tool in statistics for measuring correlated data relationships between variables, and has also found applications in signal processing and pattern recognition: its use in audio research dates back to the 1950s [32], then Stautner worked with it on the analysis of a *tabla* performance (an Indian instrument) in 1983 [48], and later on it has also been applied to analysis and manipulation musical sounds [53].
The first principal component contains the largest amount of the total variance as possible, and each successive principal component contains as much of the total remaining variance as possible: as a result of this property, PCA is often used as a method of dimensional reduction in order to discard components of lower impact on the variance of the overall data.
To explain PCA, we need to recall the concept of orthogonality. Two random variables x_1 and x_2 are said to be uncorrelated or orthogonal if

$$E[x_1, x_2] - E[x_1]E[x_2] = 0 \tag{3.1}$$

where E[x] is the expectation of the variable x.
The most common method of carrying out PCA is singular value decomposition (SVD) [49], which decomposes the $m \times n$ matrix Y as

$$Y = USV^T \tag{3.2}$$

where U is an $n \times n$ orthogonal matrix, V is an $n \times m$ orthogonal matrix and S is an $n \times m$ diagonal matrix of singular values.
The SVD is calculated by finding the eigenvectors of YY^T and Y^TY, that stand for the columns of U and V respectively, while the singular values in S are obtained from the square roots of the eigenvalues of YY^T or Y^TY and are arranged in order of decreasing variance. Dimensional reduction can then be achieved by discarding singular vectors, starting from the lowest; also, if the required number of principal components is known *a priori*, the computational cost of PCA can also be reduced by calculating only the required number of eigenvalues and eigenvectors.
Let's consider the case when Y is a time-frequency representation as a spectrogram, with m frequency bins on rows and n time frames on columns: performing PCA on it, the columns of U contain the principal components of Y based on frequency, and the columns of V contain the principal components of Y based on time. Since the ordering of components is carried out on a variance criterion, PCA favours sources of high intensity over sources of low intensity, so the number of principal components to be retained when using PCA for dimensional reduction purposes has to be chosen carefully.
As shown in [12], a drawback of the PCA relies on its incapability of adequately characterising musical signals, as it assumes a Gaussian model which

clearly does not describe musical signals that are highly non-Gaussian, and so it does not take into account information contained in the higher order statistics of musical signals.
Despite the fact that it only characterises sources up to second order statistics, PCA is still a useful tool particularly for dimensional reduction, because it orders its components by decreasing variance, thereby allowing components of low variance to be discarded. Full statistical independence can only be achieved with reference to these higher order statistics, and achieving statistical independence for non-Gaussian sources has been studied extensively and has resulted in a set of techniques known as Independent Component Analysis [11].

3.2 Independent Component Analysis

Independent Component Analysis (ICA) attempts to separate a set of observed signals that are composed of linear mixtures of a number of independent non-Gaussian sources into a set of signals that contain the independent sources [26].
Early work in this area was carried out by Comon [11], and Bell et al. [7], but recent years have seen an explosion in interest in ICA, with an ever increasing number of algorithms being presented that perform independent component analysis, such as the FastICA algorithm [28] and the kernel approach taken by Bach [5].
The underlying mathematical model for ICA can be stated as follows: let's considering N independent sources, s_i, whose signals x_i are measured by M sensors and can be mapped to the sources using an unknown function f_i, resulting in

$$x_i = f_i(s_1, \ldots, s_N) \tag{3.3}$$

Assuming that each signal x_i is a linear combination of the N sources, the system is described in matrix form by

$$x = As \tag{3.4}$$

with $x^T = [x_1 \ldots x_N]$, $s^T = [s_1 \ldots s_N]$ and A is the so-called mixing matrix, with size $M \times N$; in many systems, the number of sensors equals the number of sources, resulting in a square mixing matrix, which is supposed to be invertible from now on.
ICA then attempts to estimate the matrix A or, equivalently, to find an unmixing matrix W such that

$$y = Wx = WAs \tag{3.5}$$

is an estimate of the original source signals where $y = [y1, \ldots, y_N]$, and W is of size $N \times N$.
The matrix y will have independent components y_i if and only if

$$p(y) = \prod_{i=1}^{N} p(y_i) \tag{3.6}$$

where $p(y_i)$ is the probability density function (PDF) of y_i, and $p(y)$ is the joint PDF of the matrix y.
Similarly to PCA, when x_i are random variables with a Gaussian distribution, then decorrelation and independence are equivalent, as the Gaussian distribution does not contain information in orders higher than second: this results in the requirement that the independent variables must be non-Gaussian for ICA to work, and musical signals fit this criteria, being non-Gaussian in nature as already mentioned.
ICA tries to find an unmixing matrix W such that the resulting matrix y has component PDFs that are factorisable in the manner shown in equation (3.6), and this is possible with two constraints:

1. it is impossible to recover the source signals in the order in which they came in: since both A and s are unknown, the order of the sources can freely permutated and any of the sources can be considered as the first independent component;

2. original amplitudes of input signal get lost: since both A and s are unknown, multiplying any of the sources s_i with a scalar could always be cancelled by dividing the corresponding column a_i of the matrix A.

This can be illustrated by substituting a permutation matrix and its inverse into equation (3.4), resulting in

$$x = APP^{-1}s \tag{3.7}$$

The elements of $P^{-1}s$ are just the original independent variables in another order, while AP is just another unknown unmixing matrix which can be estimated using an ICA algorithm.
As a result of the ordering and amplitude constraints of ICA, it cannot be used for dimensional reduction purposes in a manner similar to that of PCA as there is no way of knowing the amount of variance that the recovered sources contributed to the original data: therefore, PCA allows dimensional reduction, but at the expense of not characterising non-Gaussian sources properly, while ICA is capable of characterising non-Gaussian sources but at the loss of being able to carry out dimensional reduction.

While equations (3.1) and (3.6) give definitions of statistical independence, they are not in a suitable form for the creation of an iterative algorithm that will allow estimation of the unmixing matrix W: this led to several different criteria proposed as a basis for obtaining objective functions for ICA.
These include mutual information, negentropy, maximum likelihood estimation and information maximisation ("Infomax"): it has been demonstrated that all these criteria can be viewed as variations on the theme of minimising the mutual information between output components [9], [29], [54].
It is important to note that in many cases these various algorithms and approaches will converge more or less to the same solution; it has been observed that the source separation ability of the algorithms is not particularly sensitive to the approximations to the PDFs used.
In other words, the algorithms can tolerate a fair amount of mismatch between the assumed and the actual PDFs and still achieve good separation [9]: so, many of the algorithms achieve essentially the same results in most cases, while each one outperforms the others in a particular task.

3.2.1 FastICA algorithm

FastICA is an algorithm for carrying out ICA [28] based on the use of negentropy as a cost function.
Negentropy relies on the fact that a Gaussian variable has the largest entropy among all random variables of equal variance, resulting always non-negative and zero only for a Gaussian variable:

$$J(y) = H(y_{gauss}) - H(y) \tag{3.8}$$

where y_{gauss} is a Gaussian random variable with the same covariance matrix of y.
As shown in [11] and [12], minimising mutual information is equivalent to maximising the sum of the negentropy of the individual signals; however, this task is hard using the formulation above, requiring indeed an estimate of the PDF of the variable: thus, approximations of negentropy are normally used.
Hyvärinen [27] suggested this approximation

$$J(y_i \approx c[E\{G(y_i)\} - E\{G(v)\}]^2 \tag{3.9}$$

where G can be nearly any non-quadratic function, c is a constant and can be ignored, v is a Gaussian variable of zero mean and unit variance, and E is the expectation of the function; the random variable y_i is assumed to have zero mean and unit variance.

The choice of G was taken considering the robustness of the estimates of negentropy: functions that grow slowly were found to be better in that sense, resulting in

$$G_1(y) = \frac{1}{a_1}\log(\cosh(a_1 y) \quad G_2(y) = -\exp\frac{-a_2 y^2}{2} \tag{3.10}$$

where $a_1, a_2 \in [1, 2]$ are constants. G_1 and G_2 can also be considered as approximations to the log-density of an assumed PDF, *i.e.* $G(y) \approx -\log(f(y))$ where $f(y)$ is the PDF of y: if the PDF of sources is known, then the optimal G for separating the mixtures is the log-density.
Using these functions, an iterative algorithm was designed by Hyvärien for implementing ICA (for the derivation, [28]) as follows:

1. Whiten the mixture signals: signals get uncorrelated and their variance is normalized to 1 (this reduce the number of parameters to be estimated in ICA).
2. Chose an initial weight vector w.
3. Compute $w^+ = E\{xg(w^T x)\} - E\{g'(w^T x)\}$ where g is the derivative of the function G and g is the derivative of g.
4. Normalize w^+: $W = w^+/\|w^+\|$.
5. Decorrelate the output to prevent the vectors from converging to the same maxima: $w = (WW^T)^{-1/2}W$.
6. Go to step 3 until convergence is reached.

FastICA, as its name suggests, runs faster than other ICA algorithms when dealing with large batches of data without any compromises in the performance of the algorithm: indeed, in some circumstances it is more robust than the Infomax and kurtosis based approaches.
Even if ICA has been presented as a development from PCA, the properties of those two methodologies can be "summed" in a new algorithm that uses both: this is the key–idea of Indipendent Subspace Analysis.

3.3 Independent Subspace Analysis

Independent Subspace Analysis (ISA) [10] is a totally blind source separation method, which contains no information about the sources to be separated; it is based on the concept of reducing redundancy in spectrogram representations of signals, and represents sound sources as low dimensional subspaces in the time–frequency domain.

ISA takes advantage of the dimensional reduction properties of PCA and the statistical independence achievable with ICA using the former to reduce a spectrogram to its most important components, and then the latter to make these components independent [12].
ISA assumes that the single channel sound mixture signal can be considered as a sum of p unknown independent sources:

$$s(t) = \sum_{q=1}^{p} s_q(t) \tag{3.11}$$

Carrying out the Short-Time Fourier Transform of $s(t)$ and retaining only the magnitude of the coefficients calculated (phase is not relevant for the purpose [12]), the result is the spectrogram Y, *i.e.* a $m \times n$ matrix where m and n are the number of frequency bins and time frames respectively.
Then, this spectrogram is supposed to be the linear superposition of l independent spectrograms Y_j:

$$Y = \sum_{j=1}^{l} Y_j \tag{3.12}$$

It is then assumed that each of the Y_j can be uniquely represented by the outer product of an invariant frequency basis function f_j, and a corresponding invariant amplitude envelope or weighting function t_j which describes the variations in amplitude of the frequency basis function over time: this yields

$$Y = \sum_{j=1}^{l} f_j t_j^T = f t^T \tag{3.13}$$

In practice, the assumption that the frequency basis functions are invariant stands for the fact that the pitch can't change along time frames: although this can be very limiting when trying to describe many instruments, indeed it is valid for most drum sounds, where the pitch of the drum does not change from event to event, making ISA particularly suited for analysing drum loops.
Each source is composed by a number of these independent basis functions, forming a low-dimensional subspace: once identified, the independent sources can then be re-synthesised if required. As Casey has noted, "the utility of this method is greatest when the independent basis functions correspond to individual sources in a mixture" [10]: in other words, ISA works best when each individual component corresponds to a single source.
To decompose the spectrogram in the manner described above, PCA is performed on the spectrogram Y: this is carried out using the SVD method and

yields:

$$Y = USV^T \tag{3.14}$$

as described in section 3.1. As the sound sources are assumed to be low-dimensional subspaces in the time-frequency plane, dimensional reduction is carried out by discarding components of low variance: if the first l principal components are retained then the last equation can be rewritten as

$$Y = \sum_{j=1}^{l} u_j s_j v_j^T \tag{3.15}$$

Taking $h_j = u_j s_j$ and $z_j = v_j$, it can be seen that the spectrogram has been decomposed into a sum of outer products as described in equation (3.13), in matrix form:

$$Y = hz^T \tag{3.16}$$

However, PCA does not return a set of statistically independent basis functions but only uncorrelated basis functions, so the spectrograms obtained from the PCA basis functions will not be independent.
To achieve the necessary statistical independence, ICA is carried out on the l components retained from the PCA step; this independence can be obtained on a frequency basis or a time basis: for this derivation, it will be assumed that independence in frequency is required (independence in time of the spectrograms can be derived in a similar manner).
Carrying out ICA on h to obtain basis functions independent in frequency yields

$$f = Wh \tag{3.17}$$

where matrix f contains the independent frequency basis functions and W is the unmixing matrix.
The associated amplitude basis functions can be obtained by multiplying the spectrogram Y by the pseudo inverse f_p of the frequency basis functions f, resulting in

$$t = f_p Y \tag{3.18}$$

Once independent basis functions have been obtained, a spectrogram of an independent subspace can be obtained from

$$Y_j = f_j t_j^T \tag{3.19}$$

Since ISA works on the magnitudes of the STFT coefficients, there is no phase information available to allow re-synthesis: this problem has been dealt in [12], [22] and [46].

Finally, it must be estimated how much information should be retained from the dimensional reduction step: this is of vital importance in obtaining the optimal sound source separation, keeping too few components may result in the incorrect separation of the sources, while too much can result in features which cannot be identified as belonging to a given source.
The amount of information contained in a given number of basis functions can be estimated from the normalised cumulative sum of the singular values obtained when carrying out the SVD; a threshold can then be set for the amount of information to be retained, and the following inequality can be used to solve for the number of basis functions required:

$$\frac{\sum_{i=1}^{\rho} \sigma_i}{\sum_{i=1}^{n} \sigma_i} \geqslant \phi \tag{3.20}$$

where σ_i is the singular value of the i^{th} basis function, ϕ is the threshold, ρ is the required number of basis functions and n is the number of variates.
There is a trade-off between the amount of information to retain and the recognisability of the resulting features: setting $\phi = 1$ results in a set of basis functions which support a small region in the frequency range, instead when $\phi \gg 1$, the basis functions are recognisable spectral features with support across the entire frequency range.

Limitations of ISA Though ISA does provide an effective tool of separating sound mixtures, it should be noted that there are still several problems with the method.
While the combination of PCA and ICA to perform ISA makes use of the properties of each method to perform a separation not achievable using either method *per se*, ISA retain some of the problems associated with each method, such as ICA's indeterminacy with regards to source ordering and scaling and the variance-based nature of PCA that inherently biases the analysis towards sources of high amplitude.
It should also be noted that the separation achieved, while good, is not perfect; in particular, when dealing with drum sounds which are, as already noted, broadband noise-based instruments, there will be regions of overlap between the sounds, and as a result sometimes other drums show up as small peaks in the amplitude envelopes of the separated drums.
That, combined with the higher threshold on number of components required to describe low amplitude sources, implies that each source would require an indeterminate *a priori* number of components, suggesting that PCA may not be the optimal way to achieve dimensionality reduction.

3.4 Prior Subspace Analysis

As previously mentioned, the Prior Subspace Analysis (PSA) is a technique developed by FitzGerald [12] that tries to overcome the drawbacks of ISA. The key–idea here consists in creating a model of each class with an algorithm such as the k-means (for details about this, see 3.4.2) in order to reduce the number of invariant components needed to describe that class, and retaining only the frequency ones: in this way, the separation in testing will no more be blind with respect to the sources, but pointed towards some specific kind of sounds with a greater robustness against differences in amplitude.
The derivation of this algorithm is identical to ISA's until equation (3.13), the difference lies on the way PSA decomposes the spectrogram into a set of independent invariant basis functions: ISA uses PCA followed by ICA, PSA instead assumes that exists known prior frequency subspaces or basis functions f_{pr} that are good initial approximations to the actual subspaces, *i.e.*:

$$Y \approx \sum_{j=1}^{l} f_{pr} t_i^T = f_{pr} t^T \tag{3.21}$$

Multiplying the overall spectrogram by the pseudo-inverse of the prior frequency subspaces yield estimates of the amplitude basis functions, $\hat{t}$:

$$\hat{t} = (f_{pp} Y) \tag{3.22}$$

where f_{pp} is the pseudo-inverse of f_{pr}.
Then, ICA is performed on $\hat{t}$ to calculate independent basis functions:

$$t = (W \hat{t}^T)^T \tag{3.23}$$

where W is the unmixing matrix obtained from ICA and t contains the independent amplitude basis functions.
Improved estimates of the frequency basis functions can then be obtained from

$$f = Y t_p^T \tag{3.24}$$

where t_p is the pseudo-inverse of t.
Finally, the independent spectrograms can then be individually computed as

$$Y_j = f_j t_j^T \tag{3.25}$$

Re-synthesis can be performed as cited in section 3.2.

3.4.1 Comparison with ISA

As can be argued from the name, the main distinction between PSA and ISA is that PSA requires prior knowledge of what sources are present, and prior information about the invariants associated with these sources: it only looks for as many sources as are known to be present.
The use of prior subspaces also overcomes the problem of sources at lower relative amplitudes to other sources in the mixture which was a problem when using PCA for the initial decomposition; however, PSA still suffers from the limitation that the source signals cannot be recovered in the order in which they came in, requiring a mandatory identification of the recovered subspaces with the original prior subspaces by some means such as their frequency characteristics or their amplitude envelopes.
The removal of the PCA step in PSA makes it faster than ISA and relaxes the assumption that no pitch changes are allowed: PSA only assumes that the searched sources don't change their pitch in the overall spectrogram (since this is not decomposed in a linear manner [12]), while undesired "noisy" sources are free to do that. This makes PSA suitable for attempting to transcribe drums in the presence of pitched instruments.

3.4.2 K-means clustering

As previously described, PSA involve a very famous tool often used in statistics and machine learning tasks: the k-means. Here it turns out to be useful in subspace creation, since we need a clustering tool to aggregate the frequency invariant descriptor found for each sample (for every class).
It is a heuristic method of cluster analysis whose purpose is to partition n observations into k clusters S_k (with $1 < k < n$), in which each observation belongs to the cluster with the nearest mean, in order to minimize the within-cluster sum of squares (WCSS):

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x_j \in S_i} |x_j - \mu_i|^2 \tag{3.26}$$

where μ_i is the mean of points in S_i.
The basic implementation of the algorithm is the following:

1. select k points as initial centroids;
2. form k clusters by assigning all points to the closest centroid;
3. recompute the centroid for each cluster;
4. go back to step 2 until convergence is reached (centroids don't change).

3.5 Audio features analysis

As mentioned before, our system aims to create an error–correction layer over the PSA algorithm that is based on feature analysis. Below we described the set of features used.

3.5.1 Spectral centroid

The spectral centroid is the barycentre of the spectrum; it is calculated as the weighted mean of the frequencies present in the signal, with their magnitude as the weights:

$$sc = \frac{\sum_{n=0}^{N-1} f(n)x(n)}{\sum_{n=0}^{N-1} x(n)} \tag{3.27}$$

where $x(n)$ represents the weighted frequency value of bin n, and $f(n)$ is the central frequency of that bin.

3.5.2 Spectral skewness

The skewness coefficient gives a measure of the asymmetry of the spectrum around its mean value; it is taken from the third central moment:

$$sskew = \frac{\sum_{n=0}^{N-1} x(n)(f(n) - sc)^3}{\sigma^3} \tag{3.28}$$

Value of skewness equal to 0 indicates a perfectly symmetric distribution of the spectrum, positive values indicate more energy on the left, negative values indicate more energy on the right.

3.5.3 Spectral kurtosis

The (eccess) kurtosis gives a measure of the flatness of the spectrum around its mean value; it is computed from the fourth order of the moment:

$$skurt = \frac{\sum_{n=0}^{N-1} x(n)(f(n) - sc)^4}{\sigma^4} - 3 \tag{3.29}$$

The "minus 3" at the end of this formula is often explained as a correction to make the kurtosis of the normal distribution equal to zero; negative values indicate a flatter distribution of the spectrum, positive indicate a peaker one.

3.5.4 Spectral rolloff

The spectral rolloff frequency is defined as the frequency below which 85% of the accumulated spectrum magnitude is concentrated (*e.g.* in figure 3.1),

according to [50]:

$$\sum_{k=0}^{K_{roll}} |S(k)| = 0.85 \sum_{k=0}^{N_{Ft}/2} |S(k)| \tag{3.30}$$

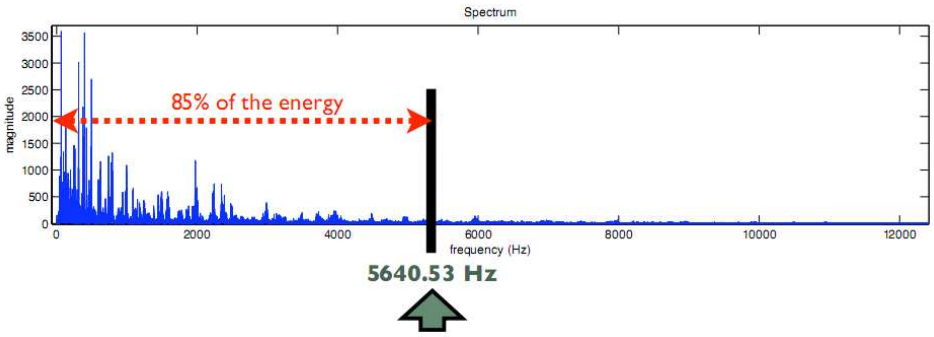

Figure 3.1: An example of spectral rolloff.

3.5.5 Spectral roughness

The roughness r of a signal is an estimation of the sensory dissonance proposed in [42]: it is related to the beating phenomena between the peaks of its spectrum.

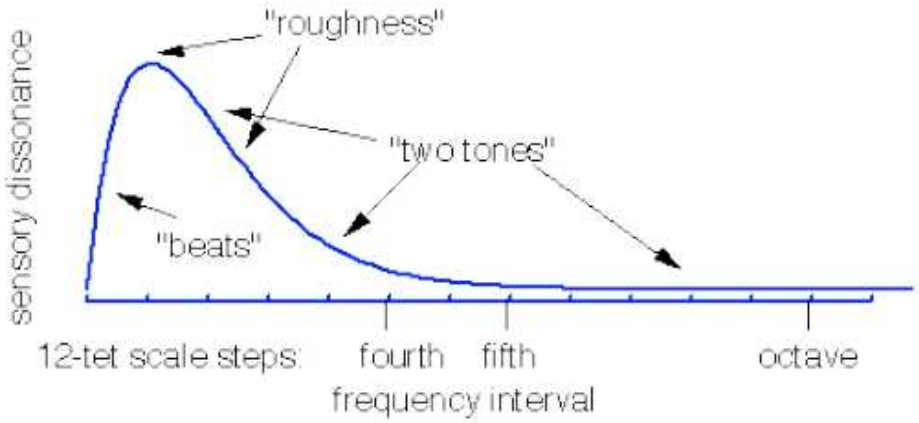

Figure 3.2: Sensory dissonance and spectral roughness (considering two tones only).

The method used refers to [45]: the roughness of a spectrum having two sinusoidal components with frequencies f_1, f_2 and amplitudes A_1, A_2, where $f_{min} = min(f_1, f_2)$, $f_{max} = max(f_1, f_2)$, $A_{min} = min(A_1, A_2)$, $A_{max} = max(A_1, A_2)$, is:

$$r = X^{0.1} * 0.5(Y^{3.11}) * Z \tag{3.31}$$

$$X = A_{min} * A_{max} \tag{3.32}$$

$$Y = \frac{2A_{min}}{(A_{min} + A_{max})} \tag{3.33}$$

$$Z = e^{-b_1 s(f_{max} - f_{min})} - e^{-b_2 s(f_{max} - f_{min})} \tag{3.34}$$

where $b_1 = 3.5$, $b_2 = 5.75$, $s = \frac{0.24}{(s_1 f_{min} + s_2)}$, $s_1 = 0.0207$ and $s_2 = 18.96$. An example of the spectral roughness with the method involved is shown in figure 3.2.
An estimation of the total roughness can be obtained by computing the peaks of the spectrum, and taking the average of all the dissonance between all possible pairs of peaks.

3.5.6 Spectral irregularity

The irregularity of a spectrum is the degree of variation of the successive peaks of the spectrum; according to [30], it is computed as the sum of the square of the difference in amplitude between adjoining partials:

$$irr = \frac{\sum_{k=1}^{N}(a_k - a_{k+1})^2}{\sum_{k=1}^{N} a_k^2} \tag{3.35}$$

3.5.7 Spectral flux

Spectral flux is a measure of how quickly the power spectrum of a signal is changing, calculated by comparing the power spectrum for one frame against the power spectrum from the previous frame. More precisely, it is usually calculated as the 2-norm (also known as the Euclidean distance) between the two normalised spectra:

$$sf = |S_n - S_{n-1} = \sqrt{(S_n - S_{n-1})^2} \tag{3.36}$$

3.5.8 Noise likeness

As proposed in [51], noise likeness (NLN) is a measure extracted to quantify the degree of noisiness of a signal, which is higher in general for percussive sounds.
For each spectral frame, NLN is computed with this algorithm:

1. find the local maxima (peak–picking according to a threshold);
2. convolve the local maxima with a gaussian impulse of zero mean and σ variance (equal to the variance of the signal in the current frame):

$$g(x) = exp(-\frac{x^2}{2\sigma^2}) \tag{3.37}$$

3. NLN is estimated as the correlation coefficient between original and resulting spectral vector.

An example of the spectra computed for NLN calculation is shown in figure 3.3.

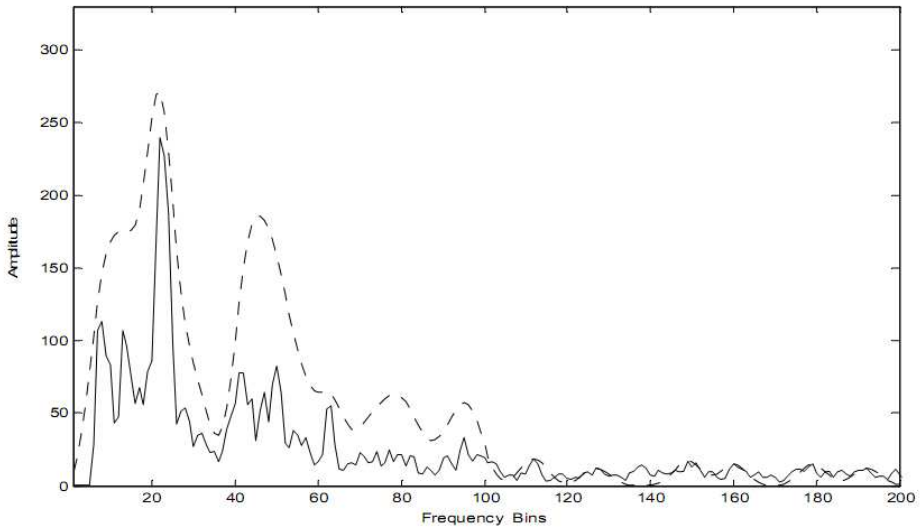

Figure 3.3: Original spectrum (solid) and model spectrum (dashed) of percussive component.

3.5.9 Inharmonicity ratio

Inharmonicity estimates the amount of partials that are not multiples of the fundamental frequency; it is computed by considering the ideal locations of harmonics vs the actual harmonics in a spectrum where f_0 is the fundamental and f_n refers to subsequent harmonics:

$$h_n = \frac{|f_n - nf_0|}{nf_0} \tag{3.38}$$

3.6 Gaussian Mixture Models

Gaussian Mixture Models (GMM) have been widely used in many classification applications: the reason for their success is the capability to model arbitrary probability densities of general features extracted from datasets; for a concise yet complete coverage of the topic, see [38].

The GMM approach assumes that the density of an observed process can be modelled as a weighted sum of component densities given by:

$$p(x|\lambda) = \sum_{m=1}^{M} c_m b_m(x) \tag{3.39}$$

where x is a d-dimensional random vector, M is a number of mixture components,c_m is the weight associated to the component and $b_m(x)$ is a Gaussian density function, parametrized by a mean vector μ_m and the covariance matrix S_m. In particular, we can define $b_m(x)$ as:

$$b_m(x) = \frac{1}{2\pi^{D/2}|S^{1/2}|} \exp\left[-\frac{1}{2}(x-\mu)'S^{-1}(x-\mu)\right] \tag{3.40}$$

The Gaussian distribution is usually a quite good approximation for a class model shape in a suitably selected feature space and extends easily to multiple dimensions (for a 2–d example, see figure 3.4).

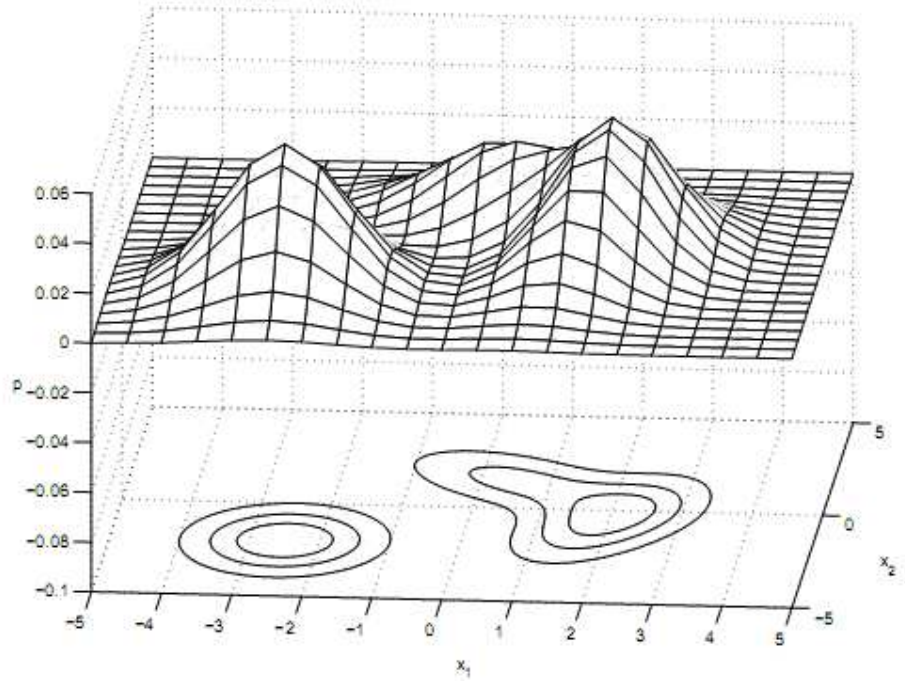

Figure 3.4: An example surface of a two-dimensional Gaussian mixture PDF with three components

3.6.1 Maximum likelihood

Assume that there is a set of independent samples $X = \{x_1, \ldots, x_N\}$ drawn from a single distribution described by a probability density function $p(x|\lambda)$.

The following function stands for the likelihood of the data x given the distribution or, more specifically, given the distribution parameters λ:

$$\mathcal{L}(x|\lambda) = \prod_{n=1}^{N} p(x_n|\lambda) \tag{3.41}$$

The goal is to find $\hat{\lambda}$ that maximizes the likelihood, *i.e.*:

$$\hat{\lambda} = \arg\max_{\lambda} \mathcal{L}(x|\lambda) \tag{3.42}$$

Usually this function is not directly maximized, instead the logarithm is considered:

$$L(x|\lambda) = \ln \mathcal{L}(x|\lambda) = \sum_{n=1}^{N} \ln p(x_n|\lambda) \tag{3.43}$$

This is the so–called log-likelihood function, being it analytically easier to handle: thanks to the monotonicity of the logarithm function, the solution to equation (3.42) is the same using $\mathcal{L}(x|\lambda)$ or $L(x|\lambda)$.

3.6.2 The Figueredo-Jain algorithm

A useful algorithm for estimating the parameters of a GMM that maximize the likelihood of a set of n data vector is the Expectation Maximization (EM) algorithm, that works by iteratively updating the parameters according (in the case of diagonal covariance matrices) to the following equations:

$$\mu_m^{new} = \frac{\sum_{i=1}^{n} p(m|x_i, \lambda) x_i}{\sum_{i=1}^{n} p(m|x_i, \lambda)} \tag{3.44}$$

$$\Sigma_M^{new} = \frac{\sum_{i=1}^{n} p(m|x_i, \lambda)(x_i - \mu_m)'(x_i - \mu_m)}{\sum_{i=1}^{n} p(m|x_i, \lambda)} \tag{3.45}$$

$$c_m^{new} = \frac{1}{n} \sum_{i=1}^{n} p(m|x_i, \lambda) \tag{3.46}$$

where the value $p(m|x_i, \lambda)$ can be computed as:

$$p(m|x_i, \lambda) = \frac{c_m b_m(x_i)}{\sum_{j=1}^{M} c_j g_j(x_i)} \tag{3.47}$$

The drawbacks of this algorithm are:

1. the outcome depends from an initial guess of the parameters to be maximized;

2. the number of components M must be known *a priori* (something like the number of clusters in the k-means algorithm, which is a direct derivation of the EM method).

The Figueredo-Jain algorithm tries to overcome these issues by adjusting the number of components during estimation: components that are not supported by the data are annihilated; FJ also allows to start with an arbitrarily large number of components: the initial guesses for component means can be distributed into the whole space occupied by training samples.
A further development consists in taking one component for every single training sample, the so-called component-wise EM (CEM): CEM updates the components one by one, computing the E-step after each component update (while the basic EM updates all components "simultaneously"), and when a component is annihilated its probability mass is immediately redistributed strengthening the remaining components.

Chapter 4

Methodology applied to drum transcription

The aim of this thesis is to improve the performance of the PSA algorithm, combining the low–level source separation with a high–level error correction algorithm based on audio feature extraction and Gaussian Mixture Models. The purpose is to reduce the number of PSA false–positive in the detection of drum events in a mixture of drums and pitched instruments.
In our work we considered kick drums, snare drums and tom–toms as the only percussive sound sources of interest.
The first section describes the creation of the dataset and the algorithm used to build prior subspaces, then the focus is moved to the research profile held in the feature analysis. Finally the overall PSA testing methodology is explained, in particular concerning the applied error correction layer.

4.1 Dataset creation

As explained in section 3.4, PSA requires known prior subspaces that are good approximations to the searched source sounds: this implies that the first task is to find the invariants in the frequency domain as treated in section 3.4), that will be used as a reference to find occurrences of a given type of sound, in our case a drum instrument, in the tested mixture signal. Therefore, the analysis of a large number of drum samples to build a model of the drum should allow us to find good average descriptors inferring from a set to the whole class.
The creation of a consistent database of drum sound is the first phase of our work. In the past some datasets have been created, to have a reference library for testing the algorithms produced by the research community. A

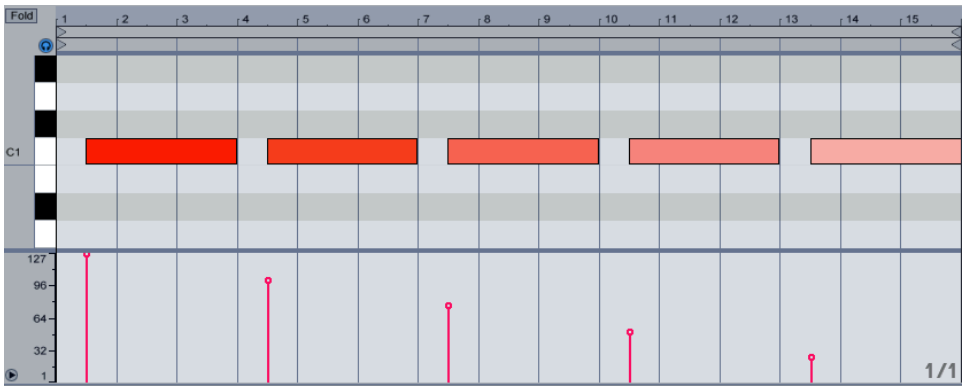

Figure 4.1: A MIDI sequence used to generate the drum samples for the training phase.

notable effort in this field has been done with the RWC database in 2002 [35], and the ENST database in 2006 [16].

RWC database, despite being very extensive and containing a large number of royalty–free songs, for our purpose is limited since it has an insufficient number of single drum hits (*e.g.* for kick drums, about 20 samples with three dynamic levels each). The ENST database, even if extremely suitable for standard testing (it also contains drum phrases, solos, and entire songs, with transcription), lacks in assortment of single drum hits and different drum kits available, since only three drum kits produced all the sounds contained in it.

The problem is that, according to the description of prior subspaces just made, it would be more reasonable for finding invariants to have a large number of different drum samples, extracted from different drum kits, with different characteristics in frequency, in order to get a more general description of each drum piece. This is why we decided to build our dataset from scratches.

The chosen path to overcome this first issue has been employing a Digital Audio Workstation software, *Ableton Live* [1] to render drum samples made available by two award–winning plug–ins, *Addictive Drums* [2] and *BFD Drums* [3].

A brief definition: virtual instruments are applications which, driven from the user via MIDI sequences, allow the playback of audio samples, with a certain level of realism in re–synthesis. We chose those two because they contain two of the most complete and high–quality recorded set of drum samples available in the world–wide market.

What we did then was to program a MIDI track to play every sample (of every drum kit selected in the plug–ins used) five times, with velocities $v =$

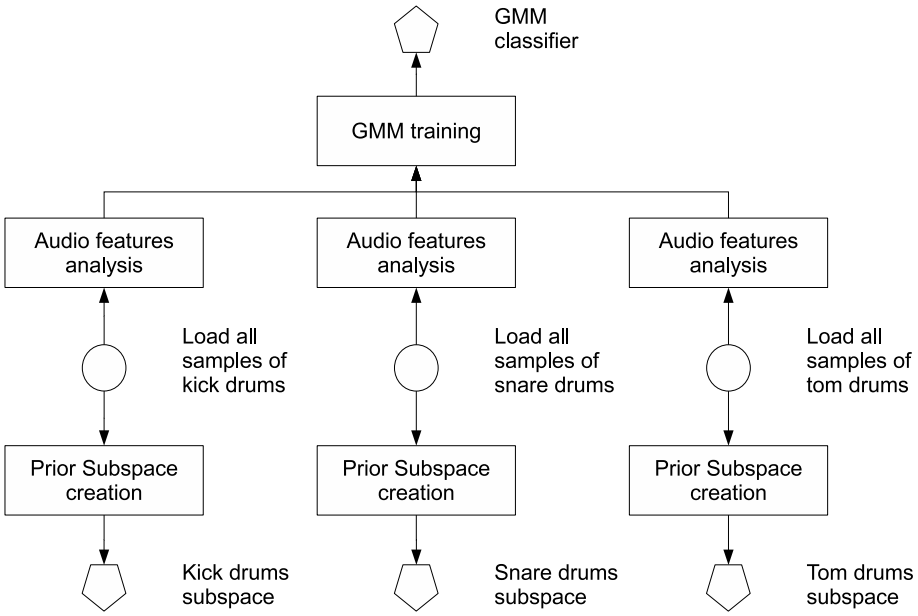

Figure 4.2: The overall block schema for the training step.

$[25, 50, 75, 100, 125]$ according to MIDI standard (which defines 128 levels of possible dynamics for a musical note), as shown in figure 4.1. Each sample results five seconds long: all the samples are aligned in time having all of their onset times after exactly one second (however, details about this information are hidden inside the plug–in and not accessible) and a silence before (zero–padding), and then a 4 seconds long decay, that is enough for this kind of percussive instruments.

The DAW adopted allows a resampling of the signal when exporting. We chose to "render" our samples to the format of 44.1 kHz sampling frequency with a word length of 16 bits, which is the standard for audio CDs.

Thus a dataset of kick drum hits, snare drum hits and tom–tom hits has been obtained, starting from the libraries of the virtual instruments, summing up to 110 samples of kick drums, 115 samples of snare drums and 215 samples of tom–toms. For more details about this, see appendix A.

Now, in order to perform transcription, the next tasks to be done are the creation of the subspaces and the analysis of audio features, that can be considered as a sort of training.

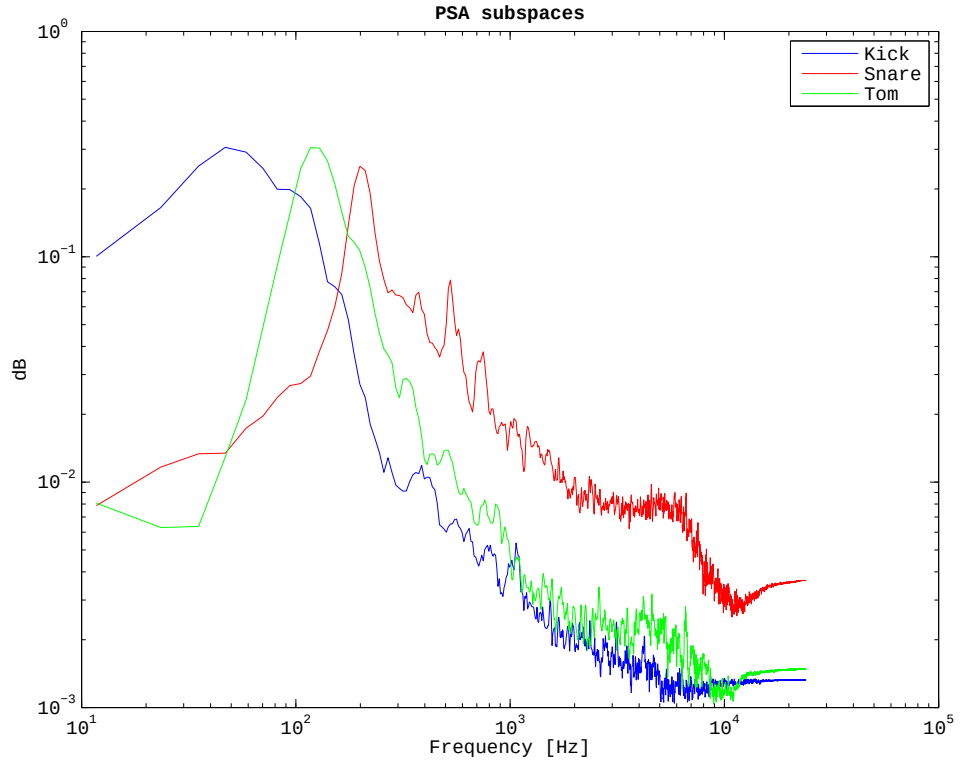

Figure 4.3: A visual representation of the prior subspaces of kick drum (blue), tom drum (green) and snare drum (red).

4.2 Training step

Here we are describing what we used to train our drum transcription algorithm, *i.e.* the creation of the prior subspaces required by PSA and the analysis of a set of audio features brought to the development of the error correction layer. The general schema used is shown in figure 4.2.

4.2.1 Prior subspace calculation

The algorithm performed on each drum type dataset, according to the original PSA formulation [12], is composed by these steps:

1. for each drum sample, the STFT of the signal is computed in order to obtain its spectrogram, using a Hamming window 2048 samples long and hop–size of 256 samples;

2. Principal Component Analysis (PCA) is performed on the frequency bins of each spectrogram;

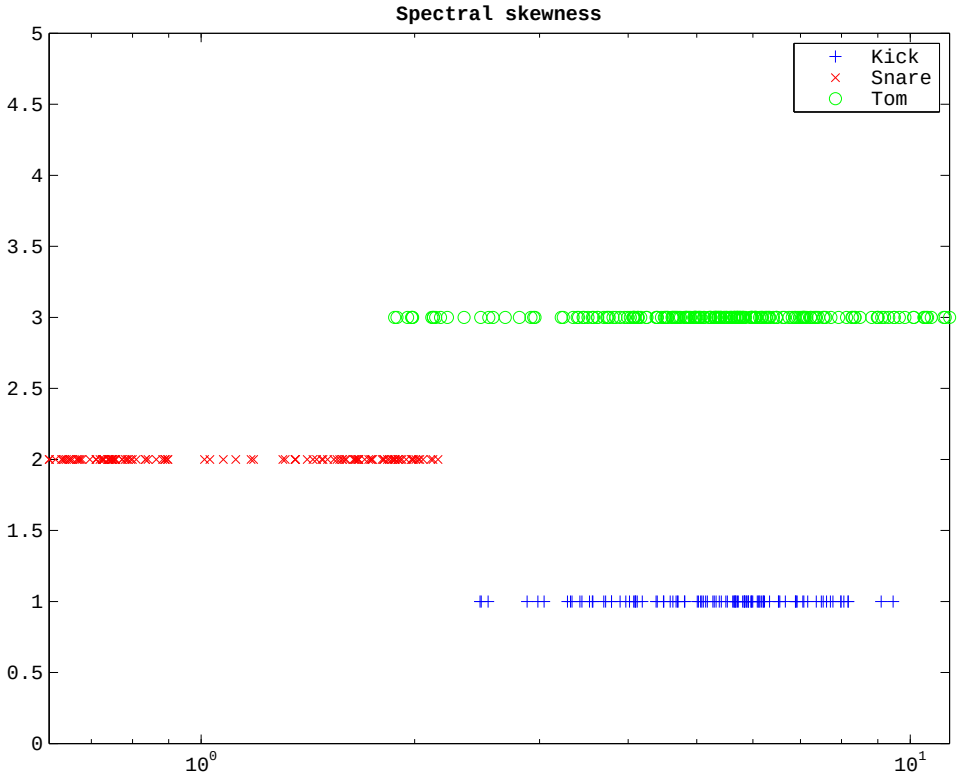

Figure 4.4: The spectral skewness computed on samples of the training set.

3. for each drum sample, Independent Component Analysis (ICA) is performed on the spectral components retained by PCA, keeping the one with the largest projected variance;

4. K-means algorithm with $k = 1$ is performed on the whole set of results (one each drum sample), to keep one descriptor, that in this case corresponds to the mean of the spectra returned by previous step.

An example of what a subspace is can be seen in figure 4.3. As can be argued, the transcription of toms using only PSA may be compromised since their spectrum of toms is somehow similar to kicks' one for frequencies greater than about 200 Hz. To overcome this issue, we decided to add the mentioned layer of error correction, whose development required an audio feature analysis step described in the next section.

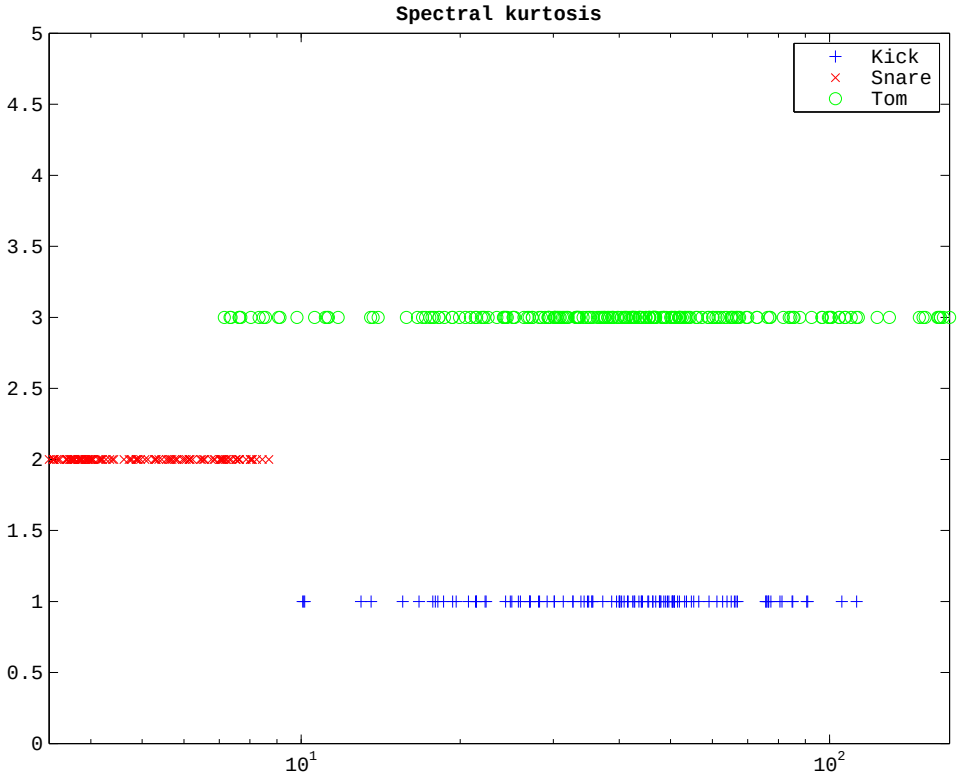

Figure 4.5: The spectral kurtosis computed on samples of the training set.

4.2.2 Audio feature analysis

As seen in the previous section, the problem of adding some descriptors able to help in distinguishing the classes searched by PSA has come to hand. Some personal experiences in recording studios and dealing with drums (and drummers), associated to many perceptual concepts, turned out to be a great resource.

For the calculation of the audio features described in section 3.5 we used the Matlab toolbox *MIRtoolbox 1.3.1*, developed by Olivier Lartillot, Petri Toiviainen and Tuomas Eerola at the Department of Music of the University of Jyväskylä, Finland. For this section we used the same dataset of subspaces creation.

Instead of performing an extensive research in the vast field of audio feature analysis, we concentrated our attention only on a limited set of features, taking as a reference the work by Helén and Virtanen [24] where they list some of the most performing ones for drums analysis (*e.g.* spectral roll–off and noise–likeness), and adding others we thought could be worth of it.

For each feature extracted, we computed its mean value on the overall sample

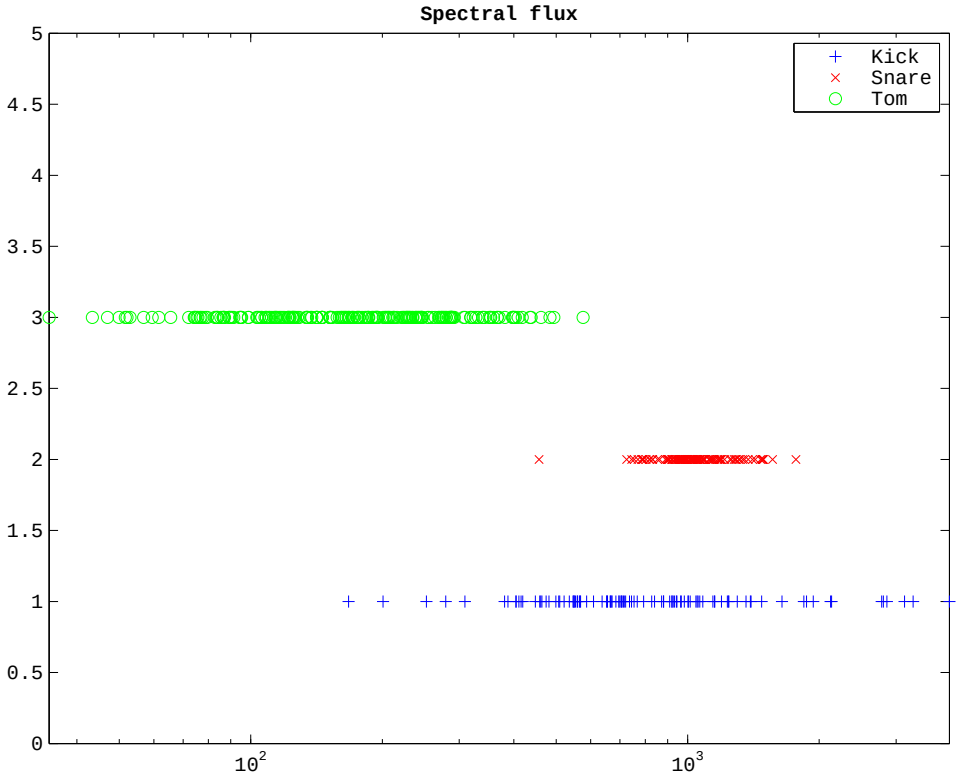

Figure 4.6: The spectral flux computed on samples of the training set.

set, taking this as reference for the setup of thresholds as it will be cleared in section 4.3.2. Our work in this sense can be subdivided in two parts, relating to the ideas involved as explained below: a statistical approach and an "envelope" approach.

Statistical approach

We focused on a set of statistical properties of the spectrum, trying to exploit the advantages of PSA compared to ISA: indeed, since PCA (sensitive to statistics of first and second order only) is actually involved in the creation of the subspaces but not in their usage on the testing (transcription) phase, we thought to "compensate" this using spectral skewness and kurtosis features that are practically, third and fourth order moments (for more on this, see chapter 3).

In particular, we expected a more symmetrical spectrum for the snare drum, since its spectrum is more relevant in high frequency compared to others. Results are shown in 4.4 and 4.5.

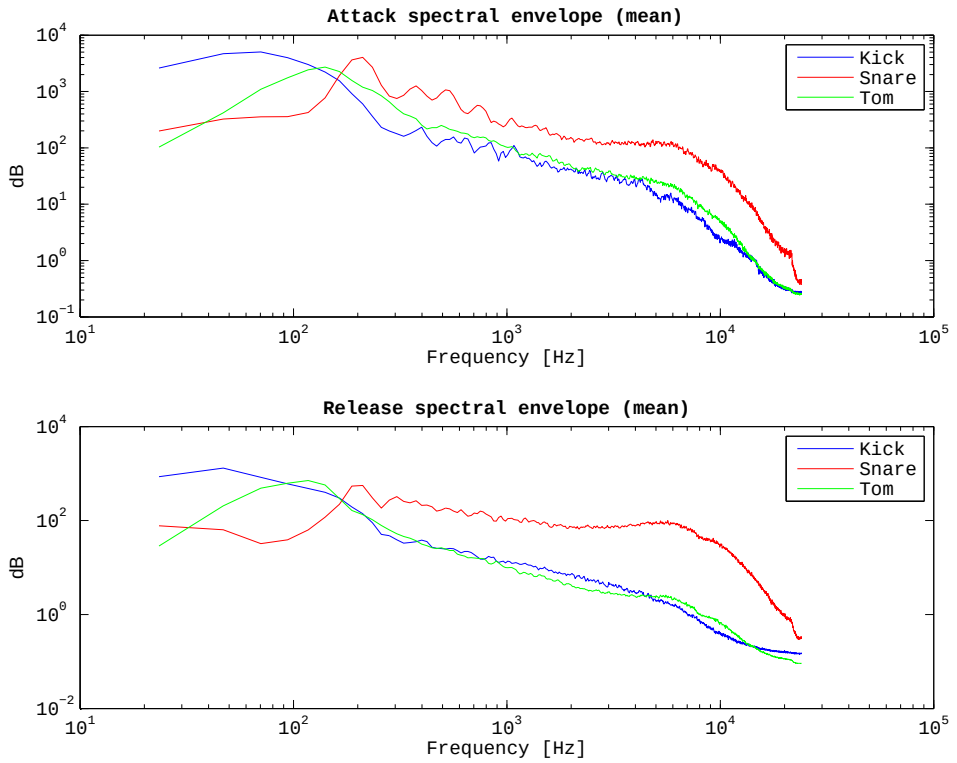

Figure 4.7: The attack and release spectral envelopes computed on segmented samples of the training set.

"Envelope" approach

Since PSA subspaces are practically a sort of average between samples spectra, it can be useful to analyse features meant to describe the shape of the spectrum and their evolution in time, resembling both concepts under the name "envelope".
As a first step, we computed a simpler feature, the spectral flux: the reason relies on the longer decay of tom–tom sounds compared to the other drums, so we looked for a feature aimed to describe the evolution of energy in time. Since the samples in the training sets are inherently time–aligned given the creation methodology, a comparison of their mean spectral fluxes is possible and coherent. As shown in figure 4.6, the results confirm what expected.
Then, an interesting "basic" question came out: strictly speaking about how the sound is generated from a physical point of view, what are the main differences between the three types of drums?

- The excitation to the tom and snare drum membrane (drumhead) is transmitted by drumsticks (usually made of wood), while for kick

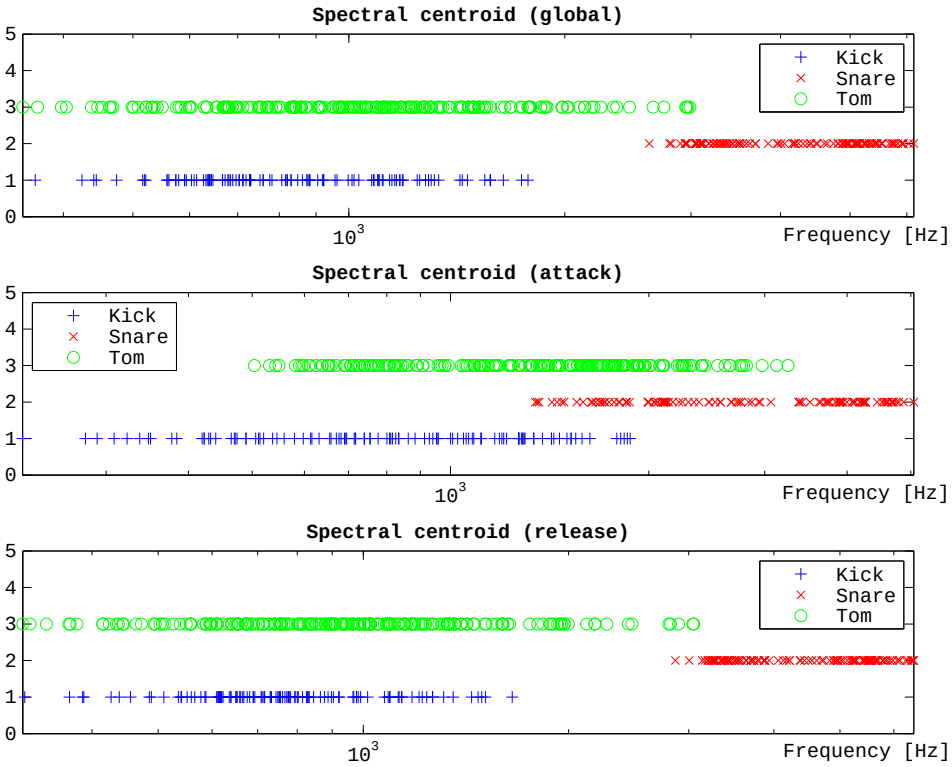

Figure 4.8: Global, attack and release spectral centroids computed on segmented samples of the training set.

drums we have a pedal beater (most of the times, felt with a metal hardware), so we expect a difference in the attack spectrum.

- The sound of the snare drum, as the name suggests, is mainly due to the presence of an hardware system (a sort of bull plate and an adjustable strainer) to hold the snare wires to the drumhead, which is unique (not present in toms and kick drums) and characterizes the release phase.

The average spectrum computed by PSA in training basically merges the informations of the attack and release phases, so a segmented analysis seemed to be a valuable solution. To evaluate this, we used the functions provided by *MIRtoolbox* to determine the onset detection on each drum sample, which is by definition the end of the attack phase, then we analysed the spectral envelopes of the segmented signals and their centroids (figures 4.7 and 4.8). Looking at these results, in particular considering the release spectrum, we decided to concentrate our efforts to discriminate the snare drum; also, in the presence of other instruments the onset detection and the following seg-

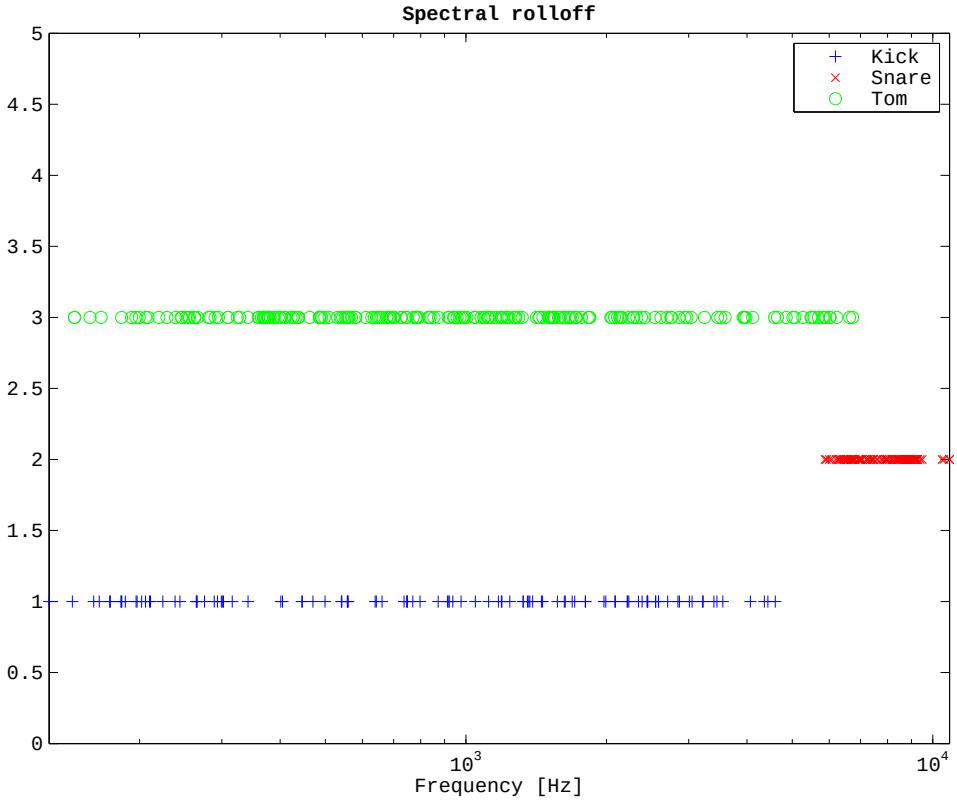

Figure 4.9: The spectral roll–off computed on samples of the training set.

mentation may be compromised by the non–drums onsets. Thus we tried to characterize this from a spectral point of view, according to the "philosophy" followed by PSA in describing sources from a frequency perspective.

Spectral roll–off frequency appeared to be a good descriptor to characterize "brightness" of the snare drum *vs* the sound of kick drums pronounced in lower bands, thinking the tom could place its values in between; results are shown in 4.9.

That result is due to the effect of the snares that vibrate on the resonant drumhead when the (so–called) snare drum is hit, producing higher frequency noise components. Thus, on the contrary, a set of features tailored to describe how much a signal resembles a noise is expected to quantify the perceived more sinusoidal release phase of tom–toms compared to snares and kicks.

This is why we decided to compute spectral irregularity, spectral roughness, inharmonicity ratio and finally noise–likeness. Results are shown in figures 4.10, 4.11, 4.12 and 4.13 respectively.

Spectral irregularity is pretty useless as not characterizing the three types of

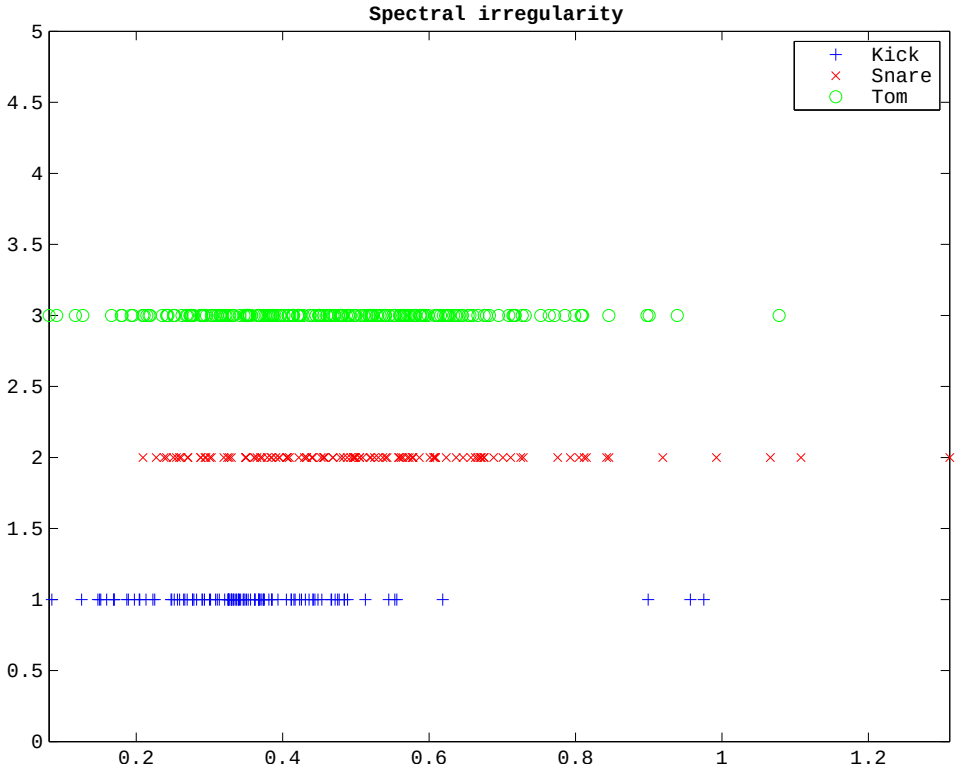

Figure 4.10: The spectral irregularity computed on samples of the training set.

drum, while noise–likeness on the opposite seems to be very promising.

4.2.3 Considerations on the analysed set of features

In general, the analysed feature set shows promising results for snare drums as in some cases (*e.g.* the spectral rolloff) the computed values are able in particular to distinguish toms from other drums. It's impossible to find a perfect plane of separation (*i.e.* an optimal clustering) for any feature by itself.

Also, kick drums and tom drums gives similar values for practically every feature considered, except for noise–likeness, inharmonicity ratio and spectral flux. The last one however is very sensitive to the presence of pitched instruments, because their sound is in general harmonically sustained and so the value of spectral flux between adjacent windows is significantly decreased, thus we decided to discard it. Unfortunately, these features are not discriminant for kicks vs snare drums, so we don't have means of distinction expressively for kick drums.

Summarizing, the set of features composed by spectral rolloff, noise–likeness

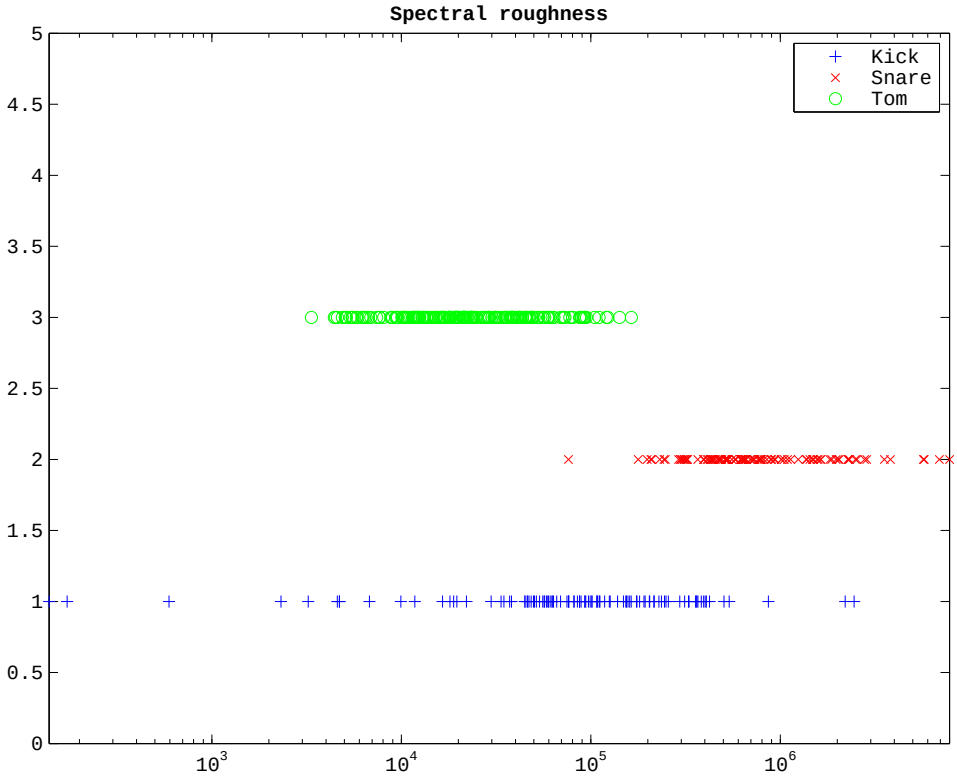

Figure 4.11: The spectral roughness computed on samples of the training set.

and inharmonicity ratio proved to be the most descriptive for our purposes, and this is the reason why we chose to implement the error correction layer as described in section 4.3.2.

4.3 Testing step

Once the training has been done, the algorithm is ready to perform drum transcription. We present here the overall schema of our methodology, as in figure 4.14.
Briefly, the algorithm works as follows:

1. once loaded the previously created prior subspaces, PSA is computed over the mixture musical signal in input, retrieve the onset time instants of each hit;

2. for each onset, the chosen feature set is computed from the corresponding segment of the input signal and a decision is taken (the event considered is retained or discarded);

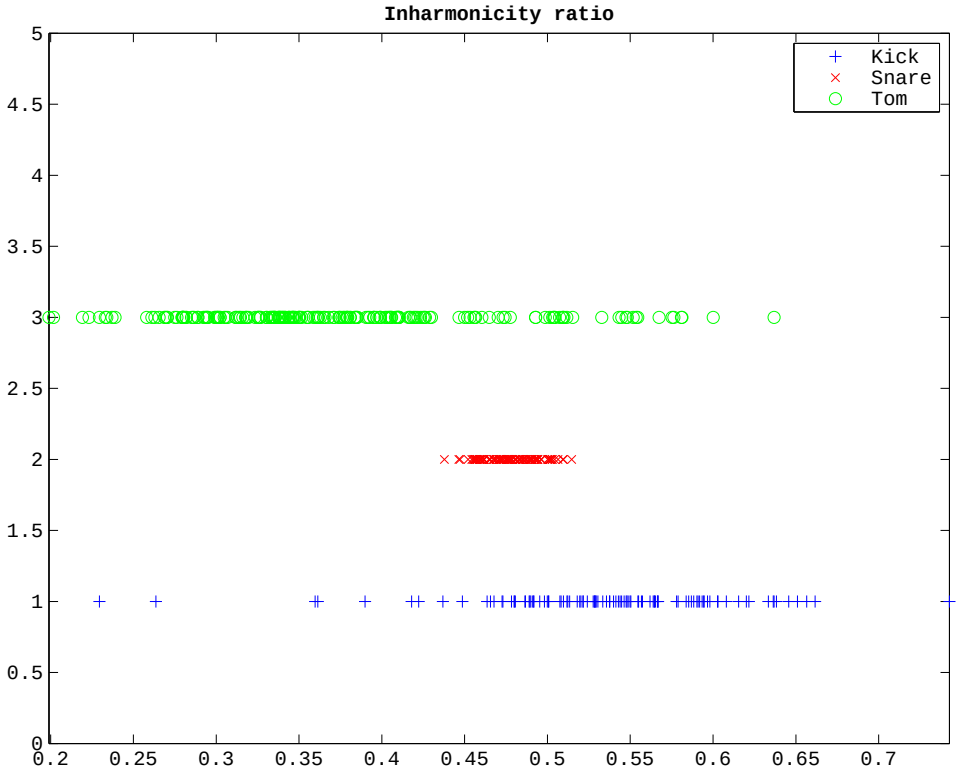

Figure 4.12: The inharmonicity ratio computed on samples of the training set.

3. considering the whole set of detected events, the time quantum (tatum) is extracted;

4. onset time indexes are quantized to a grid defined by the computed tatum line.

4.3.1 PSA stage

Our methodology relies on the events detected by PSA, according to the theory explained in chapter 3 and taking as a reference the work treated in [47], except that we added the tom–tom subspace detection to kick and snare drums just as an extension.
The input signal can be a single channel or multichannel mixture of drums and pitched instruments: in the second case, given L channels, the algorithm is carried independently over each channel producing L results for every drum instrument. This implies that next stages will require same independent process until results are merged to achieve one global transcription.
Considering a single–channel case, an improvement on the original PSA by

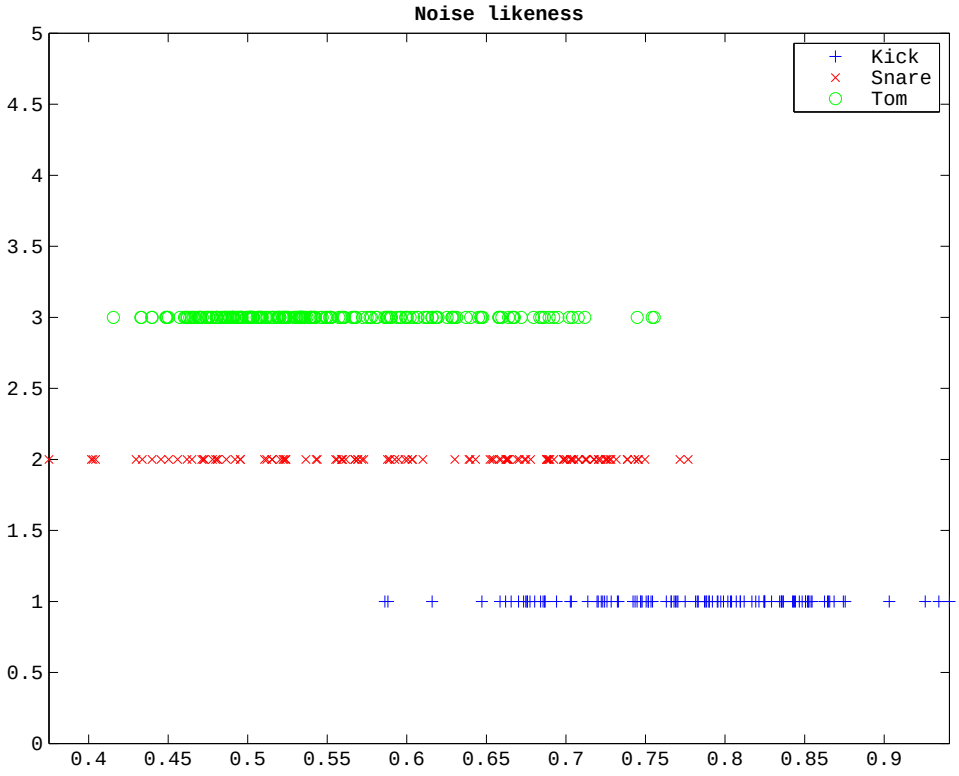

Figure 4.13: The noise–likeness computed on samples of the training set.

FitzGerald was proposed in [47] and consists in the use of a spectral modulation technique [6]. The idea behind it is to exploit the percussive character of drums as opposed to most pitched instruments: in the former, usually there can be seen fast energy transitions over a wide frequency range (*i.e* a great spectral flux), as opposed to the latter where spectrum is richer in energy near the fundamental frequency and its harmonics.

The input signal $x(n)$ is segmented with non–overlapping windows of 2 seconds, and for each block:

1. a time–frequency representation X is obtained as the STFT of x, resulting in a $m \times k$ matrix where m is the number of time slices and k in the number of frequency bins;

2. the log–difference of the spectrogram is calculated as:

$$X'_m(k) = 20 \log_{10} \frac{X_{m-1}(k)}{X_m(k)} = [X_{m-1}(k)]_{dB} - [X_m(k)]_{dB} \qquad (4.1)$$

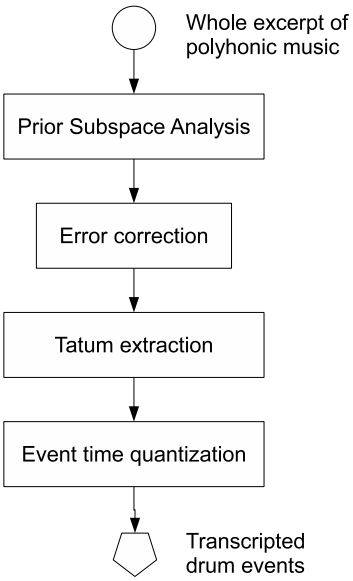

Figure 4.14: The overall schema used in our drum transcription algorithm.

3. the percussivity index is computed as:

$$P_e = \sum_k P_m(k) \tag{4.2}$$

where

$$P_m(k) = \begin{cases} 1 & \text{if } X_m(k) > T \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

with T a properly set threshold.

In practice P_e is a counter on the number of frequency bins that exceeds the fixed threshold T.

4.3.2 Error correction layer

The error correction layer proposed in this work tries to limit the number of PSA false–positive detections. Since we didn't find a distinctive feature set to describe kick drums and transcription performances for this drum instrument are already good as shown in chapter 5, we decided to concentrate our efforts on the development of an error correction for snare and tom drums only.

For each tom and snare event previously detected, we consider a portion of the signal starting from the onset time and extending to the next two windows, whose length is 2048 samples with an hop–size of 256 samples as in the analysis (training) step explained in section 4.2.
The chosen set of features is composed by spectral rolloff, inharmonicity ratio and noise–likeness, which resulted to be the most descriptive for snare and tom drums. Then, this set of features is extracted from the signal segment discussed above and, being two windows, the mean result of the two is computed.
For the snare drum, we used a rule–based approach: given the spectral rolloff mean value, we simply apply a threshold on it; if the value on the tested signal is higher, then the event considered is retained, otherwise it is discarded. The reference value used for threshold was taken as the mean obtained on snare drums samples on the training dataset: we found good results using a threshold at about 7 kHz.
This approach however didn't brought the desired results on tom–toms too, with neither of the features considered, as explained in chapter 5. Thus, just as an exploratory attempt, we build a Figuereido–Jain Gaussian Mixture Model using the three mentioned descriptors and taking a decision depending on the likelihood in output. The system can be thus summarized as:

- for snares: if the spectral rolloff computed for the given event is above the threshold, then it is retained, otherwise it is discarded;
- for toms: if log–likelihood of the GMM output is maximum for the tom–tom class, then the event is retained, otherwise it is discarded.

However, the sly global problem of feature analysis is that in a polyphonic musical context, especially when dealing with harmonically–sustained pitched instruments, it can't be assured *a priori* that the values of an audio feature will reflect what obtained on the training dataset made of only drum sounds. The signals of other instruments in this context can be considered as a "noise" but it can't be modelled and inserted in the feature analysis or GMM training phase without a lack of generality, and this leaves room for further development of our system.
The error correction layer built this way improved the results as it is shown in the next chapter.

4.3.3 Tatum extraction and quantization

According to the algorithm exposed in [47], in order to quantize the onset temporal positions, it is necessary to estimate a time quantum (tatum),
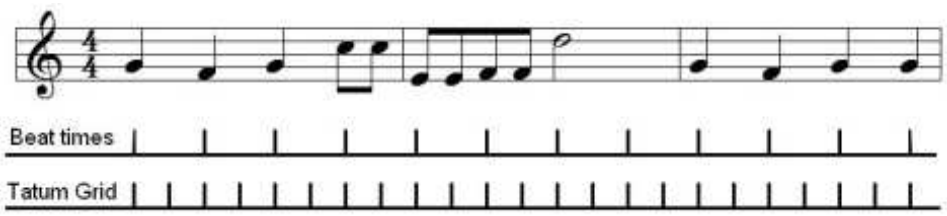
Figure 4.15: Tatum grid vs. beat times in musical piece; the tatum grid represents the set of all possible note onsets.

defined as the smallest temporal quantum in the musical piece.
This operation is not trivial as it seems because involves the identification of the beats and their pace (tempo); we assumes that all drum events occur at integer multiples of the tatum grid and that there is no tempo change for the whole duration of the excerpt.
Defining the Inter–Onset Interval (IOI) as the time that elapses between consecutive onsets, the tatum is computed iteratively as follows:

1. from the whole set of drum onsets (*i.e* considering all drum instruments together), tatum candidates are selected constructing an histogram of IOIs, as described in [44], and reatining only peaks whose amplitudes are higher than the 90% compared to the highest one;

2. a tatum candidate grid is estimated for each tatum candidate;

3. with respect to the detected actual onset times, inter–onset events that fall out of the candidate tatum line are removed;

4. gaps left by previous step are filled according to actual tatum grid;

5. the candidate tatum grid is aligned and evaluated comparing with low–level drum onsets;

6. steps 2 to 5 are repeated for each tatum candidate: the one with the highest matching rate is selected as the final tatum grid.

Once these operations are done, we are just quantizing every event to the nearest tatum grid position: if that "bin" is already containing an event, the one with the highest amplitude (likelihood of presence, coming from PSA) is retained, otherwise it is discarded.
Moreover, in the case of multichannel input signal, the tatum extraction and quantization are performed after merging the transcription results of each channel: in particular, to avoid increasing the number of false positives, we

decided to merge the results of all channels taking into account the slight differences in onset times that are to be expected. Here indeed, we assume that each channel is very much correlated to the others, as it is generally true for the majority of recorded musical performances.
So, for each drum instruments, all the events are merged in unique time line, and:

- events with the same onset time are condensed to one, retaining only the one with the maximum "amplitude";
- two adjacent events with different onset times are both retained if the difference in time is higher than the hop–size adopted when windowing the signal for the PSA step (because that is the maximum time resolution allowed by the algorithm), otherwise the second one is discarded.

Chapter 5

Evaluation of the results

This chapter contains the description of the testing phase of the proposed drum transcription method, providing details on how the results have been obtained. Section 5.1 gives an overview on the dataset used for the testing phase, on the metrics and on the parameters chosen to perform this task. Section 5.2 shows a table representation of the measures related to the performance of the testing dataset, while section 5.3 contains the necessary observations, deductions and evaluations on the performance of the developed system.

5.1 Testing the system

The proposed drum transcription system has been designed to work over whole excerpts of polyphonic music, with the presence of pitched instruments. For this reason we chose to build a dataset of 15 songs of western music, belonging to the pop-rock genre, without specific restrictions on time signature, and with different tempo values. The need to evaluate the system correctly resulted in a need to consider songs spanning a period starting 40 years ago, to have a different drum/pitched instrument mixture level.
To have an accurate description of the performance on tom drums transcription, we chose to include both songs that contain occurrences of this drum piece, and songs that do not. The transcription algorithm has been performed on 90 seconds audio excerpts, in audio CD format: 2 channels (stereo), 16 bits depth, sampling frequency equal to 44.1 kHz. The segmentation of the input signal fed to the algorithm has been performed with a Hamming window 2048 samples long and a hop-size of 256 samples. As a reference for the other parameters related to the original PSA system, we considered the work described in [47].

As described in chapter 4, an error correction layer for snare and tom drums has been developed. It relies on the extraction of a set of audio features tailored for this task: spectral rolloff, noise–likeness and inharmonicity ratio. Their usage depends on the drum instrument considered:

- for snare drums, the spectral rolloff value is compared to a threshold (defined on the mean obtained with training set): if higher, then the event considered is retained, otherwise it is discarded;
- for tom–tom drums, spectral rolloff, noise–likeness and inharmonicity ratio are the input of the GMM classifier built in the training process: if the event considered is classified correctly, then it is retained, otherwise it is discarded.

We also provide the results obtained using a rule–based approach for tom–toms: in that case we considered the median between noise–likeness and inharmonicity ratio for each event, since those (normalized) features returned the lowest values for this class of sounds with respect to kick drums and snare drums, as shown in chapter 4. The threshold used here was coherently the median between the noise–likeness mean value and the inharmonicity ratio mean value observed in the training dataset. A decision is then taken: if the mean value corresponding to the event considered is below the threshold, than it is retained, otherwise it is discarded.
The automatic transcription for each one of the drum pieces has been analysed and compared with the ground truth transcription, accounting for three types of events, that are necessary to extract performance measures: the true positive events (Tp), that are the drum events that are correctly detected and classified, the false positive events (Fp), that are those which are detected but do not correspond to an actual drum hit, and the false negative drum events (Fp), that are present in some points on the song, but not detected and classified.
The measures used in each song to evaluate the transcription performance are Precision (P), Recall (R) and F–measure (FM) (which is the weighted harmonic mean of Precision and Recall), that are represented by the following equations:

$$P = \frac{Tp}{Tp + Fp} \tag{5.1}$$

$$R = \frac{Tp}{Tp + Fn} \tag{5.2}$$

$$FM = 2 \cdot \frac{P \cdot R}{P + R} \tag{5.3}$$

5.2 Results

The algorithm has been tested twice over each song in the dataset. Firstly, the error correction layer has been disabled, achieving the results shown in table 5.1, then it was enabled leaving every other chosen parameter unmodified (table 5.2). The performance of the Error Correction layer on tom–toms was evaluated with both a rule–based approach and a GMM approach (table 5.3). For a quick comparison, see table 5.4.
For songs where a given drum instrument is not present, we replace the measures with "–" except for P where we provide the Fp number counted. "0" means the absence of true–positive detections (even if actually present).

Results of the testing phase of PSA without Error Correction layer

Song Name	Measure	Kick	Snare	Tom
01. Michael Jackson - Billie Jean	P	0.895	0.925	[90]
	R	0.959	0.989	–
	FM	0.925	0.956	–
02. The Commodores - Machine Gun	P	0.947	0.505	[156]
	R	0.959	0.895	–
	FM	0.956	0.647	–
03. Snow Patrol - Run	P	0.860	0.436	[202]
	R	0.974	1	–
	FM	0.914	0.608	–
04. Coldplay - Shiver	P	0.721	0.669	[165]
	R	0.936	0.929	–
	FM	0.815	0.778	–
05. Coldplay - Clocks	P	0.819	0.591	[329]
	R	0.965	1	–
	FM	0.886	0.743	–
06. Blondie - Maria	P	0.722	0.61702	[284]
	R	0.983	0.99	–
	FM	0.83	0.761	–
07. Toad The Wet Sprocket - All I Want	P	0.919	0.616	0.013
	R	1	1	0
	FM	0.958	0.763	0.027
08. Dire Straits - Sultans Of Swing	P	0.812	0.915	0.022
	R	0.891	0.976	1
	FM	0.849	0.944	0.0425

Continued on next page

Results of the testing phase of PSA without Error Correction layer				
Song Name	Measure	Kick	Snare	Tom
09. Roy Orbison - Pretty Woman	P	0.852	0.922	0.031
	R	0.986	0.987	0.375
	FM	0.914	0.954	0.058
10. Jet - Are You Gonna Be My Girl	P	0.675	0.614	0.063
	R	0.939	0.754	0.692
	FM	0.785	0.677	0.115
11. Pearl Jam - Last Kiss	P	0.886	0.805	0.019
	R	0.989	0.992	0.667
	FM	0.935	0.889	0.037
12. Green Day - She	P	0.908	0.589	0.0486
	R	0.990	0.977	0.429
	FM	0.947	0.735	0.0873
13. Guns'N'Roses - You Could Be Mine	P	0.839	0.547	0.491
	R	0.966	0.981	0.708
	FM	0.898	0.702	0.580
14. Green Day - Holiday	P	0.556	0.530	0.084
	R	0.983	0.885	0.73
	FM	0.710	0.663	0.151
15. Europe - Cherokee	P	0.736	0.543	0.139
	R	0.991	0.957	0.453
	FM	0.845	0.693	0.213

Table 5.1: Precision, recall and f–measure obtained by PSA without Error Correction layer

Results of the testing phase of PSA with Error Correction layer				
Song Name	Measure	Kick	Snare	Tom
01. Michael Jackson - Billie Jean	P	0.896	0.967	[5]
	R	0.980	0.989	–
	FM	0.935	0.978	–
02. The Commodores - Machine Gun	P	0.950	0.884	[34]
	R	0.987	0.934	–
	FM	0.968	0.908	–
03. Snow Patrol - Run	P	0.872	0.818	[8]
	R	0.974	0.937	–
	FM	0.920	0.874	–
04. Coldplay - Shiver	P	0.748	0.802	[47]
	R	0.931	0.802	–
	FM	0.830	0.802	–
05. Coldplay - Clocks	P	0.831	0.919	[108]
	R	0.961	0.934	–
	FM	0.891	0.926	–
06. Blondie - Maria	P	0.812	0.784	[27]
	R	0.909	0.940	–
	FM	0.858	0.855	–
07. Toad The Wet Sprocket - All I Want	P	0.940	0.616	0
	R	0.987	0.890	0
	FM	0.963	0.904	0
08. Dire Straits - Sultans Of Swing	P	0.794	0.975	0.154
	R	0.953	0.951	0.4
	FM	0.867	0.963	0.222
09. Roy Orbison - Pretty Woman	P	0.904	1	0
	R	0.946	0.943	0
	FM	0.925	0.971	0
10. Jet - Are You Gonna Be My Girl	P	0.684	0.662	0.105
	R	0.947	0.803	0.182
	FM	0.794	0.726	0.133
11. Pearl Jam - Last Kiss	P	0.886	0.895	0
	R	0.979	0.888	0
	FM	0.93	0.892	0
12. Green Day - She	P	0.912	0.724	0.082
	R	0.995	0.796	0.261
	FM	0.947	0.735	0.0873
Continued on next page				

Results of the testing phase of PSA with Error Correction layer				
Song Name	Measure	Kick	Snare	Tom
13. Guns'N'Roses - You Could Be Mine	P	0.838	0.864	0.917
	R	0.985	0.956	0.484
	FM	0.905	0.908	0.634
14. Green Day - Holiday	P	0.559	0.725	0.475
	R	0.983	0.829	0.413
	FM	0.712	0.773	0.442
15. Europe - Cherokee	P	0.766	0.790	0.724
	R	0.991	0.771	0.328
	FM	0.864	0.780	0.452

Table 5.2: Precision, recall and f–measure obtained by PSA with Error Correction layer.

We compare the results obtained on tom–toms using the rule–based error correction (RB–EC) and with the GMM (GMM–EC) in table 5.3.

Comparison between rule–based and GMM–based Error Correction on toms			
Song Name	Measure	RB–EC	GMM–EC
07. Toad The Wet Sprocket - All I Want	P	0	0
	R	0	0
	FM	0	0
08. Dire Straits - Sultans Of Swing	P	0	0.154
	R	0	0.4
	FM	0	0.222
09. Roy Orbison - Pretty Woman	P	0	0
	R	0	0
	FM	0	0
10. Jet - Are You Gonna Be My Girl	P	0	0.105
	R	0	0.182
	FM	0	0.133
11. Pearl Jam - Last Kiss	P	0	0
	R	0	0
	FM	0	0
12. Green Day - She	P	0.04	0.082
	R	0.17	0.261
	FM	0.064	0.0873
13. Guns'N'Roses - You Could Be Mine	P	0.792	0.917
	R	0.323	0.484
	FM	0.459	0.634
14. Green Day - Holiday	P	0.155	0.475
	R	0.552	0.413
	FM	0.242	0.442
15. Europe - Cherokee	P	0.462	0.724
	R	0.188	0.328
	FM	0.267	0.452
Overall	P	0.161	0.273
	R	0.137	0.230
	FM	0.148	0.223

Table 5.3: Precision, recall and f–measure obtained on tom–toms with rule–based and GMM–based Error Correction layer.

Considering the GMM–based error correction layer for toms, averaging the results provided results in table 5.4.

Overall results			
Drum piece	Measure	Standalone PSA	PSA with Error Correction
Kick Drum	P	0.810	0.826
	R	0.968	0.967
	FM	0.878	0.888
Snare Drum	P	0.655	0.849
	R	0.954	0.891
	FM	0.767	0.868
Tom Drum	P	0.101	0.273
	R	0.562	0.230
	FM	0.146	0.223

Table 5.4: Overall results of the testing phase for each drum type.

5.3 Evaluation and comments

An evaluation of the results is proposed, analysing the performance separately for kick, snare and tom drums.

5.3.1 Transcription results: kick drums

Considering the transcription results on kick drums, it can be noticed that a good performance is achieved from the PSA algorithm, with a slight improvement when enabling the error correction layer. This fact may seem contradictory, because, as shown in chapter 4 a suitable separation feature has not been found on this specific drum instrument.
However, we can deduce that, since tatum is extracted after the error correction step, the improved transcription obtained from the other drum pieces (namely snare and tom drums) enhances the tatum grid estimation, explaining the subtle enhancement just pointed out. In particular this translates into a raising of the overall precision value.

5.3.2 Transcription results: snare drums

The improvement over snare drums with the introduction of an error correction layer has been substantial. The detection of snare drum events took

great advantage from the feature analysis, which granted an increase in performance of 10%, according to the overall FM. As expected, the overall precision rate steps from 65.5% to 84.6%, proving that a feature discrimination based on spectral rolloff is able to discard a significant number of false positives.
A very slight decrease of the recall rate can be justified as a consequence of the removal of snare events due to our rule–based algorithm, that generates a trade–off effect: while a stricter threshold tends to remove true positive detections along with the false ones, a looser one is biased towards the opposite outcome.

5.3.3 Transcription results: tom drums

Tom drums transcription results are not as high as those reached in kick drum and snare drum cases. This is mainly due to two factors:

- The prior subspace of the tom drum presents evident similarities with both kick drum and snare drum ones, as discussed in section 4.2.1. The source separation carried on starting from this assumption is a cause for the presence of a large number of false positive detections.
- Tom drums, if compared to the other two drum pieces in exam, have a spectrum that resembles other components of the mixed audio signal: this is proved by the high number of false positive detections returned by the analysis of songs in which this drum piece is absent at all, while kick and snare drum have a better characterization from this point of view.

Besides from these reasons, as a matter of fact, tom drum signal is often not emphasized in the overall mix of the excerpt as a common artistic choice: this happens because the main rhythmic patterns played by drummers almost always rely on the use of kick and snare drum. This makes transcription a harder task for an algorithm that is based on amplitude onset detection like the PSA.
However, our error correction layer introduces an encouraging 8% improvement, compared to the PSA results, with a noticeable increase in the overall precision, at the cost of a loss in recall. In songs where tom–toms are absent, we counted with PSA 1226 false–positive tom hits: our layer lowered that number to 229, meaning a reduction of about 5 times.
In particular, we can see that the rule–based approach practically doesn't enhance the PSA transcription, while the GMM–based error correction outperforms the rule–based error correction by about 8% in FM: as stated

previously, the presence of other pitched instruments injects a "noise" in the features extraction that heavily modifies the values obtained. A rule–based approach is more affected by this kind of problem, and this leads us to believe that the GMM approach for the error correction layer suggested in our work is effective and suitable for transcription purposes.

Chapter 6

Conclusions and future developments

The transcription of a drum performance from a polyphonic musical excerpt represents a frontier for current research. The work proposed here achieved interesting results by making use of a low–level signal processing technique called Prior Subspace Analysis and by adding an error correction layer based on an audio feature analysis to improve its performances.

To date, the transcription accuracy proved to be very positive on kick and snare drums reaching an F–measure of almost 90%, compared to 22.3% of tom–tom, that is an important final result, considering that this drum piece has seldom been transcribed by other state of the art algorithms.

The use of spectral rolloff as a discriminant feature brought a significant overall improvement in the transcription of snare drum. On the other side, as explained in Chapter 4, noise–likeness and inharmonicity ratio, that are the features chosen to detect the occurrences of tom drums, were sensitive to noise (*e.g.* the presence of other instruments playing at the same time), hence not performing in a stable manner. To have an improvement on the transcription of tom drum, a GMM classifier has been introduced with encouraging results, as shown in Chapter 5. In must be said that it is just an exploratory attempt that requires a greater research to be tweaked optimally for this task. Thus our work can be used as a promising starting point for an improvement of this drum instrument transcription.

In the same way tom drums were included, the methodology developed allows the insertion of other percussive instruments to be analysed; for example the first ones to be added to the detection could be the hi–hat and other cymbals, since they are part of the standard drum kit together with the others analysed in our work.

Moreover, the optimization of threshold parameters would be an interesting extension: in this sense, but with a larger perspective, some room for development lies in the application of musicological concepts. Indeed, our neurological system already does a sort of adaptation to what it is listening to, in an unknown automated fashion, with a different level of accuracy depending on the experience and musical proficiency of the listener involved. For example, we can assume that, in a musical excerpt, signals produced by (pitched or unpitched) instruments have a highly correlated content. This holds in particular for bass guitar, whose performance is almost always tightly linked to drumming. This means that the combined use of several algorithms for bass guitar and other instruments transcription may turn out to be a great resource, allowing mutual comparison, and possibly further error corrections. This analysis of the relations between instrument's performances is the basis of the recognition of a musical genre, which is a process that is carried out almost everyday by our hearing system, considering the instrument present in the mixture, the notes they play, their timbre...
This type of human knowledge about content of music creates an expectation mechanism, that may actually be useful if reproduced in a transcription algorithm: *e.g.* dealing with jazz we expect a different set of percussive dynamics and timbres comparing to hard rock music. This is the reason why the detection of the musical genre of a song would be a major enhancement to an automatic music transcription method, allowing a more accurate selection of the prior subspaces. Furthermore, the features used for the detection could be more precise, if adapted to the kind of drums used in a specific type of music production, instead of the standard descriptors that are used in a the general case.
Another useful addition in an automatic drum transcription algorithm could be the modelling of post–processing effects that are usually applied to a track mixed in a production studio. This could be an important improvement, because the invariants used for separating the sound sources could be treated with the same kind of processing, subsequently increasing their similarity with the components of the mixed song we want to automatically transcribe.

Appendix A

Training dataset details

A.1 *BFD2* drum pieces listing

BFD2 is a virtual drum instrument developed by *FXpansion*. The sample set used for the construction of our dataset has been recorded by Joe Barresi, a world famous mixing and recording engineer.
A list of the drum kit pieces used is provided below, adding informations about the constructor, the materials, the size and the drumheads of every single drum component involved. For more details, see [3].

A.1.1 Kick drums

- Pork Pie 22x18" Tobacco Satin, DW plastic beater, drumheads: REMO Power Stroke 3 (beater), REMO Clear Ambassador (front w/hole)
- Ludwig 70's Clear Vistalite 22x14" Kick, plastic beater, drumheads: REMO Clear Black Dot (front), REMO Clear Ambassador (beater)
- Gretsch 24x18" Grape Stain, DW plastic beater, drumheads: REMO Coated Ambassador(beater), REMO Coated Ambassador (front w/hole)
- Gretsch 22x16" Natural Blonde Maple, wood beater, drumheads: REMO Power Stroke 3 (beater), REMO Clear Ambassador (front w/hole)
- Pork Pie 24x18" Glamouflage, plastic beater, drumheads: REMO Power Stroke 3 (beater), REMO Clear Ambassador (front w/hole)
- Ludwig 22x14" Psychedelic Swirl Red, fuzzy beater, drumheads: REMO Clear Black Dot (beater), Smooth Ludwig White Logo (front)
- Gretsch 24x18" Grape Stain, drumheads: REMO Power Stroke 3 (beater), REMO Clear Ambassador (front)

- Gretsch 24x16" Pink Stain, drumheads: REMO Power Stroke 3 (beater), REMO Clear Ambassador (front)
- Pork Pie 24x20" Puke Green Sparkle, drumheads: REMO Power Stroke 3 (beater), REMO Clear Ambassador (front)
- Pork Pie 22x20" Red, White and Blue Glass Sparkle, drumheads: REMO Power Stroke 3 (beater), REMO Smooth Ambassador (front)
- Pork Pie 20x18" Mustard Seed Brown, drumheads: REMO Power Stroke 3 (beater), REMO Clear Ambassador (front)
- Pork Pie 24x18" Tobacco Sunburst, drumheads: REMO Power Stroke 3 (beater), REMO Clear Ambassador (front)

A.1.2 Snare drums

- Ludwig 14x6.5" 70's Black Beauty, drumheads: REMO Clear Black Dot
- Ludwig Vistalite 14x5", drumheads: REMO Clear Black Dot
- Tama Signature Series - Stewart Copeland 14x5", drumheads: REMO Coated Ambassador
- Tama Rosewood 14x3.5", drumheads: REMO Coated Black Dot
- Zildjian–Noble & Cooley Alloy 14x7", drumheads: REMO Coated Emperor
- DW Classic 14x5", drumheads: REMO Clear Black Dot
- Tama Superstar "Big Red" 14x8", drumheads: REMO Emperor X
- Ludwig 70's Supraphonic 14x6.5", drumheads: REMO Emperor X
- Ayotte wood hoop Green Day stain 14x5", drumheads: REMO Emperor X
- Zildjian–Noble & Cooley 14x5", drumheads: REMO CS Coated Black Dot
- Pork Pie Glamouflage 12x6", drumheads: REMO Clear Black Dot
- Tama Rosewood 14x8", drumheads: REMO Coated Emperor X

A.1.3 Tom–tom drums

- HI Pork Pie 10x8" Tobacco Satin, drumheads: REMO Coated Emperor (top), REMO Clear Ambassador (bottom)
- MID Pork Pie 12x9" Tobacco Satin, drumheads: REMO Coated Emperor (top), REMO Clear Ambassador (bottom)
- FLOOR Pork Pie 16x14" Tobacco Satin, drumheads: REMO Coated Emperor (top), REMO Clear Ambassador (bottom)
- HI Ludwig Vistalite 12x8", drumheads: REMO Clear Black Dot (top), REMO Clear Ambassador (bottom)
- MID Ludwig Vistalite 13x9", drumheads: REMO Clear Black Dot (top), REMO Clear Ambassador (bottom)
- FLOOR Ludwig Vistalite 16x16", drumheads: REMO Clear Black Dot (top), REMO Clear Ambassador (bottom)
- HI DW Classic Radio King (Reissue) 12x8", drumheads: REMO Suede Ambassador (top), REMO Clear Ambassador (bottom)
- FLOOR DW Classic Radio King (Reissue) 16x16 ", drumheads: REMO Suede Ambassador (top), REMO Clear Ambassador (bottom)
- HI Gretsch 10x8" Natural Blonde Maple, drumheads: REMO Coated Emperor (top), REMO Clear Ambassador (bottom)
- MID Gretsch 12x9" Natural Blonde Maple, drumheads: REMO Coated Emperor (top), REMO Clear Ambassador (bottom)
- FLOOR Gretsch 16x14" Natural Blonde Maple, drumheads: REMO Coated Emperor (top), REMO Clear Ambassador (bottom)
- HI Pork Pie 13x10" Glamouflage, drumheads: REMO Coated Emperor (top), REMO Clear Ambassador (bottom)
- MID Pork Pie 16x14" Glamouflage, drumheads: REMO Coated Emperor (top), REMO Clear Ambassador (bottom)
- FLOOR Pork Pie 18x16" Glamouflage, drumheads: REMO Coated Emperor (top), REMO Clear Ambassador (bottom)
- RACK Ludwig 13x9" Psychedelic Swirl Red, drumheads: REMO Clear Black Dot (top), REMO Clear Ambassador (bottom)

- FLOOR Ludwig 16x16" Psychedelic Swirl Red, drumheads: REMO Clear Black Dot (top), REMO Clear Ambassador (bottom)

A.2 *Addictive Drums* drum pieces listing

Addictive Drums, developed by *XLN Audio*, is another virtual drum instrument used to generate the necessary dataset for prior subspaces creation and audio features analysis.
A list of the drum kit pieces used is provided below, adding informations about the constructor and the size of every single drum component involved. For more details, see [2].

A.2.1 Kick drums

- DW Collector's 20x16"
- Ludwig Vintage 20x16"
- Ludwig Vintage 22x16"
- Ludwig Vistalite 22x16"
- Pearl Masterworks 22x16"
- Pearl Reference 22x16"
- Tama Starclassic 22x18"
- Ludwig Vistalite 24x16"
- Sonor Designer 24x18" Kick
- Ludwig Vistalite 26x14"

A.2.2 Snare drums

- Pearl Masterworks 10x4" Piccolo
- DW Collector's Copper 14x5"
- Ludwig Supraphone 402 14x5"
- Ludwig Vintage 14x5"
- Ludwig Vistalite 14x5"

- Pearl Reference 14x5”
- Sonor Designer 14x5”
- DW Craviotto 14x5.5”
- Tama Starclassic 14x5.5”
- Ludwig Supraphone 402 14x6.5”
- Pearl Sig.Ferrone 14x6.5”

A.2.3 Tom–tom drums

- DW Collector’s 8x7”
- DW Collector’s 10x8”
- Pearl Reference 10x8”
- Tama Starclassic 10x8”
- Sonor Designer 10x9”
- Pearl Reference 12x10”
- Sonor Designer 12x10”
- DW Collector’s 12x8”
- Ludwig Black Oyster 12x9”
- Ludwig Vistalite 12x9”
- Tama Starclassic 12x9”
- Ludwig Black Oyster 13x10”
- Ludwig Vistalite 13x10”
- Sonor Designer 13x11”
- Ludwig Blue Oyster 13x9”
- Ludwig Blue Oyster 14x10”
- DW Collector’s 14x11”
- Ludwig Vistalite 14x12”

- Pearl Reference 14x14”
- Sonor Designer 14x16”
- DW Collector’s 16x14”
- Ludwig Black Oyster 16x16”
- Ludwig Blue Oyster 16x16”
- Ludwig Vistalite 16x16”
- Pearl Reference 16x16”
- Sonor Designer 16x16”
- Tama Starclassic 16x16”
- Ludwig Blue Oyster 18x18”

Bibliography

[1] *Ableton Live* website. `http://www.ableton.com/live`.

[2] *Addictive Drums* website. `http://www.xlnaudio.com/`.

[3] *BFD Drums* website. `http://www.fxpansion.com/`.

[4] David S. Alves, Jouni Paulus, and Josè Fonseca. Drum transcription from multichannel recordings with non-negative matrix factorization. pages 894–898, August 2009.

[5] F.R. Bach and M.I. Jordan. Kernel independent component analysis. pages 1–48, 2002.

[6] D. Barry, D. FitzGerald, E. Coyle, and B. Lawlor. Drum source separation using percussive feature detection and spectral modulation. In *ISSC*, Dublin, 2005.

[7] A.J. Bell and T.J. Sejnowski. An information-maximisation approach to blind separation and blind deconvolution. 1995.

[8] Jeffrey Adam Bilmes. Timing is of the essence: Perceptual and computational techniques for representing, learning, and reproducing expressive timing in percussive rhythm. Master's thesis, MIT, 1993.

[9] Jean-François Cardoso. Infomax and maximum likelihood for blind source separation, 1997.

[10] M.A. Casey and A. Westner. Separation of mixed audio sources by independent subspace analysis. In *ICMC*, 2000.

[11] P. Comon. *Independent component analysis – a new concept: signal processing*. 1994.

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

[13] Derry FitzGerald, Bob Lawlor, and Eugene Coyle. Drum transcription in the presence of pitched instruments using prior subspace analysis. In *ISSC*, Limerick, 2003.

[14] Derry FitzGerald, Bob Lawlor, and Eugene Coyle. Prior subspace analysis for drum transcription. In *AES*, Amsterdam, 2003.

[15] Derry FitzGerald and Juoni Paulus. Unpitched percussion transcription. *Signal Processing Methods for Music Transcription*, 2006.

[16] Olivier Gillet and Gaël Richard. ENST-Drums: an extensive audio-visual database for drum signals processing. In *ISMIR*, 2006.

[17] Olivier Gillet and Gaël Richard. Supervised and unsupervised sequence modelling for drum transcription. In *ISMIR*, 2007.

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

[19] Masataka Goto and Yoichi Muraoka. A sound source separation system for percussion instruments. *The Transactions of the Institute of Electronics, Information and Communication Engineers D-II*, 1994.

[20] Fabien Gouyon, Perfecto Herrera, and Pedro Cano. Pulse-dependent analysis of percussive music. In *AES 22nd*, 2002.

[21] Fabien Gouyon, Francois Pachet, and Olivier Delerue. On the use of zero-crossing rate for an application of classification of percussive sounds. In *Conference on Digital Audio Effects (DAFx-00)*, 2000.

[22] D. Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 32(2):236–243, 1984.

[23] Toni Heittola and Anssi Klapuri. Locating segments with drums in music signals. *IRCAM*, 2002.

[24] Marko Helen and Tuomas Virtanen. Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine. *EUSIPCO*, 2005.

[25] Perfecto Herrera, Alexandre Yeterian, Re Yeterian, and Fabien Gouyon. Automatic classification of drum sounds: A comparison of feature selection and classification techniques. In *Proceedings of 2nd International*

Conference on Music and Artificial Intelligence, pages 69–80. Springer, 2002.

[26] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. pages 411–430, 2000.

[27] Aapo Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit, 1998.

[28] Aapo Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis, 1999.

[29] Aapo Hyvärinen. Independent component analysis. *Neural Computing Surveys*, 2, 1999.

[30] Kristoffer Jensen. *Timbre Models of Musical Sounds*. PhD thesis, University of Copenhagen, Denmark, 1999.

[31] I.T. Jolliffe. *Principal Component Analysis*. Springer, 1896.

[32] H.P. Kramer and M.V. Mathews. A linear coding for transmitting a set of correlated signals. In *IRE Transactions Information Theory, IT-2*, pages 41–46, 1956.

[33] D.D. Lee and H.S. Seung. Algorithms for non-negative matrix factorization. In *Neural Information Processing Systems*, pages 556–562, Denver, USA, 2001.

[34] Von Hornbostel Erich M. and Curt Sachs. Classification of musical instruments. *The Galpin Society Journal (Galpin Society), Vol. 14, pp. 3–29*, March 1961. Translated from the Original German by Anthony Baines and Klaus P. Wachsmann.

[35] Hiroki Hashiguchi Ryuichi Oka Masataka Goto, Takuichi Nishimura. Rwc music database: Popular, classical, and jazz music databases. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR)*, 2002.

[36] J.A. Moorer. *On the Segmentation and Analysis of Continuous Musical Sound by Digital Computer*. PhD thesis, Dept. of Music, Stanford University, 1975.

[37] Arnaud Moreau and Arthur Flexer. Drum transcription in polyphonic music using non-negative matrix factorisation. In *ISMIR*, 2007.

[38] Pekka Paalanen. *Bayesian classification using gaussian mixture model and EM estimation.* Lappeeranta University of Technology, Finland, 2004.

[39] Jouni Paulus and Tuomas Virtanen. Drum transcription with non negative spectrogram factorisation. *EUSIPCO*, 2005.

[40] Jouni K. Paulus and Anssi P. Klapuri. Conventional and periodic n-grams in the transcription of drum sequences. In *International Conference on Multimedia and Expo (ICME)*, 2003.

[41] Geoffroy Peeters. A large set of audio descriptors (similarity and classification) in the cuidado project. Technical report, IRCAM, April 2004.

[42] R. Plomp and W. J. M. Levelt. Tonal consonance and critical bandwidth. *The Journal of the Acoustical Society of America*, 38(4):548–560, 1965.

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

[44] J. Seppänen. Tatum grid analysis of musical signals. *IEEE Workshop on Application of Signal Processing to Audio and Acoustics*, 2001.

[45] William A. Sethares. *Tuning Timbre Spectrum Scale.* Springer, 1 edition, January 1998.

[46] M. Slaney. *Pattern Playback in the 90s*, volume Advances in Neural Information Processing Systems 7. MIT Press, 1996.

[47] Andrio Spich, Massimiliano Zanoni, Augusto Sarti, and Stefano Tubaro. Drum music transcription using prior subspace analysis and pattern recognition. In *13th Int. Conference on Digital Audio Effects (DAFx-10)*, Graz, Austria, September 2010.

[48] J.P. Stautner. Analysis and synthesis of music using the auditory transform. Master's thesis, MIT EECS Department, 1983.

[49] S. Subhash. *Applied Multivariate Techniques.* 1996.

[50] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. In *IEEE Transactions On Audio, Speech, And Language Processing, Vol. 10, No. 5*, July 2002.

[51] C. Uhle, C. Dittmar, and T. Sporer. Extraction of drum tracks from polyphonic music using independent subspace analysis. In *Proceedings of ICA*, pages 843–848, Nara, Japan, April 2003.

[52] Tuomas Virtanen. Sound source separation using sparse coding with temporal continuity objective. In *International Computer Music Conference, ICMC*, Singapore, 2003.

[53] N. Ward. The application of principle component analysis to the implementation of a set of compositional tools for electroacoustic music. Master's thesis, Dublin Institute of Technology, 2002.

[54] Te won Lee, Mark Girolami, Anthony J. Bell, and Terrence J. Sejnowski. A unifying information-theoretic framework for independent component analysis, 1999.

[55] Kazuyoshi Yoshii, Masataka Got, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. An error correction framework based on drum pattern periodicity for improving drum sound detection. In *ICASSP*, 2006.

[56] Kazuyoshi Yoshii, Masataka Goto, and Hiroshi G. Okuno. Automatic drum sound description for real-world music using template adaptation and matching methods. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 184–191, 2004.

[57] Kazuyoshi Yoshii, Masataka Goto, and Hiroshi G. Okuno. Drum sound recognition for polyphonic audio signals by adaptation and matching of spectrogram templates with harmonic structure suppression. 2007.

[58] A. Zils, F. Pachet, O. Delerue, and F. Gouyon. Automatic extraction of drum tracks from polyphonic music signals. In *Proceedings of WedelMusic 2002*, December 2002.