Synergies between Musical Source Separation and Instrument Recognition

Master Thesis Report

Juan José Bosch Vicente

MASTER THESIS UPF / 2011

Master in Sound and Music Computing

Master thesis supervisor:

Dr. Jordi Janer

Master thesis advisors:

Perfecto Herrera, Ferdinand Fuhrmann

Department of Information and Communication Technologies

Universitat Pompeu Fabra, Barcelona

UNIVERSITAT POMPEU FABRA

Acknowledgements

I would first like to thank Prof. Xavier Serra for giving me the opportunity to join the SMC Master's program at the Pompeu Fabra University, and the whole Music Technology Group for providing an excellent intellectual framework for the completion of this thesis, specially the members and SMC students which have somehow influenced the development of this work. Special thanks to Perfecto Herrera and Ferdinand Fuhrmann for all their ideas, recommendations, revisions, and for the provision of the code and original dataset that I used as a basis for the instrument recognition in this work. Many thanks to thank Ricard Marxer for the clarifications, as well as Yamaha for the opportunity they provided me to work with their separation algorithm. Special thanks as well to Jordi Janer for his supervision, clarifications, suggestions, and patience.

I would like to specially thank my family for all their love and support throughout the years, without which I could not have written this thesis. Thanks as well to friends and everyone else that supported me during this year, such as Andrea with her computing hardware (and patience).

Many thanks as well to "La Caixa" Fellowship program for their confidence and their financial support during this year.

Abstract

Due to the increasing amount of digital music available, there is a clear need of a proper organization and effective retrieval. Automatic instrument recognition techniques are useful for satisfying such needs, by labeling music pieces with their instrumentation, but also as support to the extraction of other semantic information such as the genre. Source separation has also recently been applied to facilitate the analysis of musical data, as well as to other applications such as karaoke or post production. In contrast with the huge need for both algorithms, the results obtained so far show that there is still much room for improvement.

The main purpose of this thesis is to find synergies between instrument recognition and source separation algorithms in two different tasks: 1) the separation of a target instrument from the accompaniment, and 2) the automatic labeling of songs with the predominant music instruments. Several combination strategies are presented, aimed at overcoming some of the limitations of current state-of-the-art algorithms. In the first task, instrument recognition is used to detect the presence of the target instrument in order to apply or bypass the separation algorithms. In the second task, source separation is used to divide the polyphonic audio signal into several streams, given as input to the instrument recognition models. Promising results were obtained in the conducted experiments, showing that this is a path to be further investigated.

Table of contents

List of figures

List of tables

1 Introduction

1.1 Motivation and goals

During the last 15 years, we have seen how the amount of music we have access to, has been increasing in a way that it goes far beyond the time we have to listen to it. Due to the amount of data available both locally and remotely, there is a clear need of a proper organization (such as cataloguing or indexing) and an effective retrieval of the musical data we are interested in.

The instrumentation of musical pieces is a very useful descriptor which can be successfully exploited for their retrieval at several levels. A use case in which this usefulness is obvious would be when a user is interested in finding songs with the presence or absence of certain instruments. Additionally, the retrieval of music of a certain genre can certainly be enhanced by knowing the instrumentation of a song. A simple example would be that the knowledge of the presence of a banjo in a song makes the piece more likely to be country or folk than classical music. Furthermore, the instrumentation in a song is also one of the most important cues for the perceived similarity between two songs [1]. Thus, the labeling of pieces of music with the most relevant instruments which are present can help to bridge the semantic gap. The semantic gap is due to the misleading connections between low level acoustical descriptors (attributes computed directly from the raw audio signals), and the higher level data which represents the semantic interpretation of the audio [2]. This is considered to be the main problem for increasing the performance of Music Information Research (also known as Music Information Retrieval, or simply MIR) algorithms. Due to the semantic gap, it is very hard to go above 75% accuracy (glass ceiling) in many of the MIR tasks [3]. Thus, musical instrument recognition can help bridging this gap [4], which would be very relevant for both research and industry: music recommenders or automatic taggers of large music databases would highly benefit from it. The automatic classification of the instruments present in a musical piece would also be an important step towards the realization of the semantic web, since it deals with one if the major bottlenecks: the manual annotation of data.

Independently, source separation algorithms have been proven to be useful for many applications [5]. Audio source separation deals with the problem of recovering the original signals from a mixture by computational means. Even though the quality of the source

separation in real world musical signals can still be much improved, separating or at least increasing the presence of a source or a group of sources in a mixture (e.g. harmonic-percussive separation) helps to increase the results within MIR tasks, such as chord detection, melody extraction [6], genre classification, etc, which could also help bridging the semantic gap. There are several approaches to source separation depending on the number of mixture channels, prior knowledge about the characteristics of the sources, etc. The more knowledge about the sources, leads to an easier and better source separation. This process is thus enhanced by the identification of the instruments present in the mixture. On the other hand, the identification of instruments is easier in monophonic than in polyphonic mixtures, therefore, a source separation pre-step should improve the detection of musical instruments.

The goal of this master thesis is the study of the relation between source separation and instrument recognition algorithms, and the investigation of the synergies between them. The main motivation is my interest in both areas and the relevance of this research question: there have been previous attempts to combine both source separation and instrument recognition, but it is not a solved question, and there is still much room for improvement. This master thesis aims thus for the application of source separation algorithms in order to enhance the recognition of music instruments in polyphonic mixtures and vice versa. Several source separation methods are considered, so as to compare different approaches. One of the frameworks for source separation has been developed by the MTG in collaboration with Yamaha, and has been mainly focused on the separation of the singing voice in polyphonic mixtures. On the other hand, the instrument recognition framework considered is based on the work presented by Fuhrmann and Herrera in [4], which deals with the tagging of music excerpts with the most relevant instruments that are present.

This work will hopefully be of relevance to the research community, contribute to the body of knowledge and the state-of-the-art in the field, and eventually to improvements in the application of source separation techniques in MIR, and vice versa. Additionally, it can be useful for the industry, in terms of the previously introduced applications.

1.2 Structure of the thesis

This document is divided into five different chapters. The first chapter introduces the reader into the motivation and goals of this work, and presents the structure. The second chapter describes relevant work for this thesis, including theoretical background and the state-of-the-art techniques and algorithms in both musical source separation and instrument recognition. The methodology for the combination of both algorithms is presented in Chapter 3, along with the data and the evaluation measures employed. Chapter 4 details the experiments conducted, and presents the results of the application of instrument recognition to improve source separation and vice versa. Finally, Chapter 5 presents the contributions and conclusions obtained with this work, and directions for further research are proposed.

2 State of the art

This chapter presents the theoretical background relevant for this thesis and a review of relevant work, mainly dealing with instrument recognition and source separation. Finally, it presents some of the approaches in the literature which have used timbral models to improve source separation, and approaches which use source separation as a prior step to instrument recognition.

2.1 Theoretical background

The theoretical background upon which the rest of the thesis is built is introduced in this section, assuming familiarity with basic concepts within signal processing, statistics and musicology. Timbral features are introduced, along with some statistical methods. Both of them form the basis of the automatic instrument recognition, and also play a crucial role in source separation.

2.1.1 Timbral features

Timbre is the term used to differentiate sounds which have the same pitch, intensity, and duration. Even though this is probably the most accepted definition of timbre, there have been many others, due to its difference in meaning in several contexts. Timbre is thus a vague word, encompassing many parameters of perception.

Humans use timbre information to discriminate between musical instruments, and a considerable number of studies have investigated this ability. Martin [7] provided a review of the findings of the work of several authors in his PhD thesis, including the instruments which were more difficult to be identified, or the difference of accuracy between musicians and non musicians. Additionally, he conducted experiments testing both the human and the computer's ability to recognize western orchestral instruments. Herrera et al. [8] also provided some conclusions which can be extracted from the literature; first, the recognition of instruments by humans is easier if they are presented musical phrases instead of isolated notes; second, it is easier to recognize families of instruments (e.g. chordophones, aerophones, etc.) than instruments; third, the accuracy in the classification decreases with a higher number of categories (instruments); and finally, the musical training helps in the recognition.

Much work related with timbre has been undertaken by specialists in several disciplines. In 1977, Grey conducted listening tests to create a multidimensional space, with the most representative dimensions of the timbre of a musical instrument [9]. A technique called multidimensional scaling was used to capture the mental representation of the stimuli, by exploring the perceived similarity between them. More recently, Iverson and Krumhansl [10], and McAdams et al [11] have further worked in the creation of timbral spaces, finding low level acoustic features which correlate with the perceptual dimensions. Timbral features refer to acoustic descriptors computed directly from the audio signal, with several possible temporal scopes. Peeters et al. [12] provide a profound explanation and possible applications of the timbre descriptors standardized in the ISO standard MPEG-7. Several features have been found to explain the dimensions of the timbre spaces, such as the spectral centroid, log-attack time, spectral flux or the attack synchrony. Based on the findings by several authors such as Schouten, Burred [13] considers several factors to be important for the perception of the timbre of a musical instrument: the temporal and spectral envelopes, the degree of harmonicity of the partials, noise content and transients. In this work a timbre model is presented, which is based on a compact representation of the spectral envelope, with a detailed characterization of the temporal evolution.

However, the probably most common features used to characterize timbral information are the well known MFCCs, which were firstly used in speech recognition systems [14]. These features stands on the source-filter model of speech production, in which speech signals are considered to be the convolution of a source signal coming from the vocal cords, and the impulse response of the vocal tract. The MFCCs are usually computed following the following steps: 1) Taking the Fourier Transform of a signal (or a part of it), 2) Map the power spectrum to a Mel scale, with triangular overlapping windows, 3) Calculate the logarithm of the powers in each Mel frequency, 4) Calculate the Discrete Cosine Transform. MFCC values are the result of the previous calculations, and they represent the coarse shape of the power spectrum of a signal. It is common to use only the lower n coefficients, the number n being dependent on the application: e.g. 13 coefficients are typically used for the representation of speech. A smaller amount of them has been used in some music related applications (such as genre recognition [15]). Other authors such as Logan and Salomon [16], and Essid et al. [17] have reported the use of a higher amount of coefficients in other applications such as music similarity and instrument recognition. The

MFCCs are often used along with their first order derivatives (e.g. difference of the MFCC vectors in two consecutive frames), in order to consider the temporal dimension. Other authors which have used MFCCs in the instrument recognition application are Heittola et al. [18]. All previous approaches report the use of the magnitude spectrum as input for the computation of the MFCC, but it is also possible to consider a modified version, in which the features are not computed on the spectral envelope. Marxer et al. [19] follow this approach for timbre description, by considering the MFCCs calculated on the Harmonic Spectral Envelope (HSE). The HSE is obtained by interpolating the values of the magnitude spectrum at the positions of the partials, using the Akima interpolation method [20].

Alluri and Toiviainen [1] recently presented a study on polyphonic timbre in which they study the correlation of several features with three perceptual dimensions: activity, brightness and fullness. Their findings suggest that there may be regularities in the way people perceive polyphonic timbre, and that there are similarities with the perception of monophonic timbre. An unexpected finding was that the MFCCs do not correlate considerably with any of the perceptual dimensions, even though they are so widely used. This contrasts with the work of many other authors, such as Terasawa et al. [21] which suggests that MFCC are a good perceptual representation of timbre.

The approach by Marxer et al. [19] corresponds with the framework developed in the MTG for source separation, which will be used in this master thesis; therefore, the modified version of the MFCCs will be considered. The instrument recognition framework used in this master thesis uses the classical MFCCs and also many other features [22][4].

2.1.2 Statistical classification

In a generic way, classification is related to the task of assigning labels to observations. The labels correspond to classes or categories in which we organize a certain domain, or a part of the world which is of interest.

A common way to perform such classification is with taxonomies, which organize the categories in a hierarchical form. A richer and more complex structure can be obtained with ontologies, which allow more diverse relations between classes. According to Bowker [23], the ideal classification structure should: 1) have consistent and unique principles to perform the

classification, 2) consider classes which are mutually exclusive, and 3) be complete, by fully covering the part of the world it intends to consider.

Statistical classification deals with the automatic classification of new observations by means of supervised learning, using a model which has been trained with previously annotated data. According to the number of classes involved, classification can be considered as binary or multiclass. In binary classification, only two classes are considered, while more than two classes are considered in multiclass classification. Multiclass classification is commonly avoided, since most methods work with binary classification. In the case of requiring a classification of more than two classes, several binary classifiers are typically combined. There are different strategies for combining the classifiers, such as the one versus one with pair-wise coupling [24], or the one-versus-all approach, in which only the presence or absence of a class is considered. In a one versus one approach, a classifier is trained for each of the possible combination of classes. For instance, in instrument classification it would be: clarinet vs. trumpet, clarinet vs. violin, violin vs. trumpet, and so on, for all possible combinations of instruments. In the one-versus-all approach, the classifier discriminates between the target class, and an artificial class which contains the rest of classes, e.g: violin vs not violin. Of course, in the latter approach, a smaller amount of binary classifiers is needed. Finally, the output probabilities of the binary classifiers are combined to decide the class membership.

An important classification method for this thesis is Support Vector Machines (SVM), which has been used in the categorization of instruments both in monophonic and polyphonic mixtures, based on the acoustic features used to model their timbre. SVM is a non-probabilistic linear and binary classifier, which is based on the creation of a hyperplane of a high dimensionality, to separate between the elements of two classes. It is a supervised learning method, and thus uses training data, in order to find the hyperplane with the largest distance to the points in the training set of any of the two involved classes. This is supposed to provide the best classification results, since it lowers the generalisation error of the classifier. Additionally, it is relatively fast to train and use a SVM classifier, and provides good accuracy with reduced over-fitting. SVM can be used for several tasks, such as classification, or regression. The implementation of the SVM used in this thesis is LIBSVM [25].

2.1.3 Dimensionality reduction

In order to properly analyze huge amounts of data, a suitable representation needs to be found. Such a representation should make explicit the latent structure of the data, and reduce the number of dimensions, in order to apply further methods [26]. The techniques presented in the following subsections are very polyvalent, and can potentially be used in very different applications or with different purposes.

Principal Component Analysis and Independent Component Analysis

Principal Component Analysis (PCA) is a very popular dimensionality reduction technique, introduced by Karl Pearson in 1901. The main goal is to decorrelate a set of input variables, by converting them into a smaller set of uncorrelated variables, while maximizing the variance of the projected data. The variables in the new space, which are named principal components, are a linear combination of the original variables. The whole set of principal components has the same dimensionality as the original set. In order to reduce the dimensionality, a smaller set of principal components is selected. This produces that some information is lost; however, PCA is conceived to minimize this loss. Applications of PCA include data compression, image processing, and data visualization. Since it also serves as a signal decomposition technique, the relation to source separation becomes evident.

An extension to PCA is ICA, which searches for a linear transformation of the original variables in order to minimize the statistical dependence between the components of two vectors. The difference with PCA is that it does not only deal with a second order independence, and does not just provide solutions which are orthogonal. ICA has been widely used in source separation, as it will be introduced in the corresponding subsection.

Non-Negative Matrix Factorisation

Non-Negative Matrix Factorisation (NMF) is an technique used to decompose a matrix $V \in \mathbb{R}^{\geq 0, n \times m}$ into two factors $W \in \mathbb{R}^{\geq 0, n \times r}$ and $H \in \mathbb{R}^{\geq 0, r \times m}$, where r is called the decomposition factor. V is approximated to the product of two matrices with non negative elements: $V \approx \hat{V} = W \cdot H$.

Paatero and Taaper presented this technique under the name “Positive Matrix Factorisation” in 1996 [27], and three years later, Lee and Seung presented further investigations on the

algorithm, using the name NMF [28], which has become widely used. In this work, PCA and NMF were compared in two different tasks, showing that different results are achieved, partly due to a difference in constrains.

NMF can be used for dimensionality reduction, and is increasing in popularity due to the reported good results in several domains, including source separation. There are several forms of NMF, depending on the measure of the divergence between V and $\hat{V}$. In order to find the best approximation, the distance measure $D(V \mid\mid \hat{V})$ is to be minimized. Several distances can be used, such as the simple Frobenius norm (square norm) used by Lee and Seung [28], or the generalized Kullback-Liebler distance which is also commonly used for source separation:

$$D(V \mid\mid \hat{V}) = \sum_{ij} (V_{ij} \cdot log \frac{V_{ij}}{\widehat{V_{ij}}} - V_{ij} + \widehat{V_{ij}}) \tag{2.1}$$

In order to reduce the selected distance measure, several iterative update methods can be used, such as gradient descent algorithms, or a multiplicative update algorithm, as presented by Lee and Seung [28].

Adding sparseness constrains to the NMF provides solutions which are easier to interpret [26]. Sparseness refers to the fact of having only a small number of coefficients not equal to zero in a vector or matrix. The sparseness is maximum (its value is 1) when only one component is not equal to zero and the sparseness is minimum (equal to 0) when all components are non-zero. The sparseness can be applied to both W and H, depending on the application [26].

2.2 Automatic instrument recognition

2.2.1 Overview, principles and applications

The automatic recognition of instruments is based on the previously presented timbre models and features. The timbre of an instrument can be characterized with some audio features, such as MFCCs or MPEG-7 features, and by means of a set of training data, a statistical classifier can learn how to categorize previously unseen audio excerpts into classes, which may correspond to instruments, or groups of instruments, depending on the approach. It is thus very important to know how the features can be used, how to compute them, how to properly select the most relevant ones, and how could they be transformed in order to have a better distribution, which

would allow a more robust classification [8]. Dimensionality reduction is also an important step, which can lead to a better classification.

The automatic recognition of musical instruments is an important task in MIR, with applications such as the automatic annotation of databases with information about the orchestration. This can be helpful to bridge the semantic gap, since the perceived similarity between songs is in a high degree dependent on their instrumentation. Additionally, the knowledge of the musical instruments present in a song supports other MIR tasks such as genre classification, and thus allows moving towards the realization of the semantic web, since it helps dealing with its major bottleneck: manual annotation.

2.2.2 Isolated musical instrument classification

Most of the work in automatic instrument recognition has been focused on isolated musical instrument classification. An extensive review of such approaches can be found in the work by Herrera [8], with accuracies that reach 90%, a number of classes below ten, and several classification techniques. The classification on isolated notes allows the simplification of the signal processing needed to extract relevant features, and has the advantage that there are sound databases which can easily be used to test the algorithms, such as the RWC database [29].

2.2.3 Polyphonic instrument recognition

More recent works deal with polyphonic instrument recognition, which is a more demanding and realistic problem, both with and without source separation. There have been some attempts to perform instrument recognition in polyphonic mixtures without using source separation as a pre-step. Some early approaches focused on the detection of specific instruments or voice, such as Tzanetakis in 2004 [30]. Heittola tried detecting the presence of several instruments (bowed, electric guitar, piano, saxophones and vocals) by using MFCCs and their derivatives ΔMFCCs and Hidden Markov Model (HMM) classifiers [18]. The performance was reported to be different for each of the instruments, getting the best results for the detection of bowed instruments and voice. For the rest of instruments, the results were not much above chance. Better accuracies were reported in the same work for the detection of drums. A more recent work by the same author dealt with the use of source separation as a pre-step for the instrument classification, as will be presented in a further section. In 2005, Essid [17] used a taxonomy

based hierarchical classification approach, training the classifiers not on the instruments themselves, but on a combination of them, such as: double bass, drums, piano and tenor sax.

In a more recent approach, Fuhrmann [22] approached the automatic recognition of predominant instruments, with SVM classifiers trained with features extracted from polyphonic audio. Figure 2.1 shows a schema of the supervised classification system in this work:

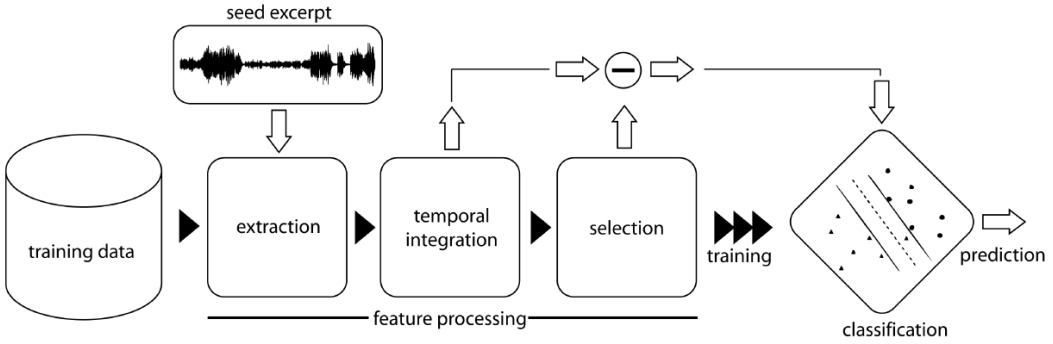


Figure 2.1: Supervised approach for predominant instrument recognition in polyphonic music [22]

The black arrows in the figure denote the workflow followed in the training stage, while the white arrows are followed in the classification stage, when unseen excerpts are to be annotated with the predicted tags.

A different approach is presented by Fuhrmann and Herrera [4], where the focus is not set on the recognition of the instruments in a frame basis, but on the most predominant instruments in a whole audio excerpt. Several strategies for labeling the music pieces are proposed, which include exploiting the temporal dimension of music: segments in which there is a predominant instrument are found, and then the labels of each of the segments are combined to provide the confidence of each instrument to be present in the whole piece. As in the case of the human ability of discriminating between musical instruments, automatic approaches also have more difficulties with certain kind of sounds. Fuhrmann et al [22] report that the accuracy of detecting the sax being the predominant instrument in polyphonic music is the lowest, around 40%, while the average classification accuracy for pitched instruments, with 11 different classes is 63%.

A comparison of the results between the approaches is not straightforward due to several reasons. The number of categories used is typically different, which certainly influences the results. Additionally, the classification task may be different in each work, e.g.: classification based on families of instruments instead of instruments, interest on the predominant instruments

or on all present instruments, the use of real world musical signals or artificially created mixtures, etc.

2.3 Source Separation

Sound signals are commonly a mixture of several signals. Sound Source Separation (SSS) deals with the problem of recovering the original audio signals from a mixture by computational means. A typical example is the *cocktail party problem*, in which one tries to follow the conversations held simultaneously in a room, with music and other noises. This is a relatively easy task for humans, which are able to concentrate the attention on a specific source within a mixture of signals which may even have interfering energies. However, it is much more difficult to teach a machine how to do this.

The interest in this problem began in the mid 1980's, and the attention of the research community to source separation increased in the 1990's, with the use of Independent Component Analysis (ICA) [31], and the Computational Auditory Scene Analysis (CASA). The CASA approach tries to imitate the mechanisms involved in human perception, which allow the recognition of sources in a mixture. Bregman introduced the cognitive process called Auditory Scene Analysis (ASA) in 1990 [32], proposing five principles used by the brain to group and isolate sounds: proximity, similarity, good continuation, closure and common fate. These principles, which are similar to the ideas of the Gestalt, are applied to both frequency and time domains of the audio signals. Based on the work by Bregman, several approaches have been proposed to deal with the computational modeling of ASA. Wang and Brown presented in 2006 a detailed literature review in this field [33]. This master thesis will be based on source separation methods built on a mathematical basis, exploiting the statistical properties of the sources and mixtures, instead of using the approach by CASA.

An overview of Source separation methods can be found in the work by Vincent et al. [34], Siamantas et al. [35], and more recently by Burred [13]. In this work, Burred divides source separation methods according to the assumptions made on the statistical nature of the models of the sources. If little or no assumptions are made, they are said to be *Blind Source Separatio*n (BSS) methods, which include ICA, and time-frequency masking methods. If more advanced models of the sources are used, they are classified as *Semi-Blind Source Separation* (SBSS) methods. Examples of SBSS include sinusoidal models, and supervised methods in which a

database of sounds is used for training the SS algorithms. Finally, *non-blind source separation* methods make use of other information than the mixture, such as the musical score.

Vincent et al. proposed in 2003 a topology for the classification of the applications of source separation, dividing them into two groups [36]: *Audio Quality Oriented* (AQO) applications and *Significance Oriented* (SO) applications. The former (AQO) applications deal with a full separation of the sources, with the best possible quality, and include: unmixing, remixing, hearing aids or postproduction. The latter (SO) are less demanding, and are more feasible with the current state-of-the-art techniques. SO applications deal for instance with several tasks within MIR, and therefore, could help to bridge the semantic gap. Some tasks that can benefit from the use of source separation include: instrument recognition, chord detection, melody extraction, audio genre classification, etc. Burred [13] complements the previous classification, devising four different paradigms: *Understanding without separation*, in which the mixture itself is used to gain knowledge about the constituent source signals, *Separation for understanding*, which corresponds to the SO scenario, *Separation without understanding*, which deals with Blind Source Separation (BSS), and finally, *Understanding for separation*, which deals with supervised source separation, based on a training database.

2.3.1 Overview and principles

This subsection provides an overview and the most important principles of the source separation problem.

Mixing models

Several mixing models can be applied to combine several sources into a mixture. Each of the models corresponds to a real world situation. The most basic is the *linear mixing model*, in which the mixture is a combination of the original sources, with a possible amplitude scaling. The mathematical formulation is:

$$x_i(t) = \sum_{j=1}^{N} a_{ij}\, s_j(t) \qquad i = 1,\ldots,P \tag{2.2}$$

In the previous equation, $x(t) = [x_1(t),\cdots,x_P(t)]^T$ is the vector of observed mixtures, $s(t) = [s_1(t),\cdots,s_N(t)]^T$ is the vector of the original sources, and A corresponds to the mixing

matrix, which is used to transform from the signal space to mixture space, P is the number of sensors or mixtures, and N the number of sources. The mixing matrix has a size of $N \times P$, and its elements are the coefficients a_{ij}.

In a linear model, sound source separation deals thus with solving the system $X = AS$:

$$\begin{pmatrix} x_1(t) \\ \vdots \\ x_P(t) \end{pmatrix} = \begin{pmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{P1} & \cdots & a_{PN} \end{pmatrix} \cdot \begin{pmatrix} s_1(t) \\ \vdots \\ s_N(t) \end{pmatrix} \tag{2.3}$$

In this system, X is known, S is unknown, and A is in most of the approaches also unknown. Depending on P (the number of sensors or mixtures) and N (the number of sources), the system is overdetermined if $P > N$, determined if $P = N$, or underdetermined if $P < N$. The most challenging task is solving underdetermined systems.

In a *delayed model*, each of the sources needs some time to arrive to each sensor, and thus the mixture is a combination of the original sources with different delays. The formulation is as follows:

$$x_i(t) = \sum_{j=1}^{N} a_{ij} s_j(t - t_{ij}) \qquad i = 1,\ldots,P \tag{2.4}$$

In a *convolutive mixing model*, there is a filtering process between the sources and sensors, such in the case of a reverberant room, where the sources can follow several paths to arrive to the sensors:

$$x_i(t) = \sum_{j=1}^{N} \sum_{k=0}^{+\infty} a_{ijk} s_j(t - t_{ijk}) \qquad i = 1,\ldots,P \tag{2.5}$$

The contribution of the source j to a sensor i can be modeled with the impulse response of a filter $a_{ij}(t)$, where $*$ is the convolutional product:

$$x_i(t) = \sum_{j=1}^{N} a_{ij}(t) * s_j(t) \qquad i = 1,\ldots,P \tag{2.6}$$

It can also be expressed in a matrix notation as:

$$X = AS \tag{2.7}$$

In musical source separation, which is the scope of this thesis, most of the mixtures are underdetermined, since we typically deal with one or two observations, for mono and stereo respectively ($P = 1$ or $P = 2$), and more than two sources (instruments) present in the mixture. This makes necessary the simplification of the problem, by taking some assumptions on the statistical nature of the sources, or use models which increase the feasibility of the separation.

Signal models

In digital signal processing, it is common to assume that the signals can be decomposed into a weighted sum of expansion functions, and the choice of a function depends on the context or application [13]. Some common models which make use of a fixed basis function for the representation of a signal in the frequency domain are the Discrete Fourier Transform (DFT) and the Discrete Cosine Transform (DCT). In order to consider the time along with the frequency domain, the Short- Time Fourier Transform (STFT) is commonly used. Increasing the resolution in one of the domains decreases the resolution in the other domain, which is related to the uncertainty principle in signal analysis.

The previously introduced PCA and ICA are also a specific case of a signal model, in which the expansion functions are extracted from the signal itself, and thus they are data-driven functions.

It is important to note that signal decomposition is very related to source separation. In fact, some of the approaches used for source separation, such as ICA, have also been successfully applied to signal decomposition.

Solving the system

The main problem in source separation is solving the previously introduced system $X = AS$, in which typically both A and S are unknown. This system can be solved in two ways: by firstly estimating the mixing matrix and then the sources in a staged manner, or in a joint manner. The mixing matrix estimation deals with finding the coefficients of the matrix A, or similarly, the mixing directions (the columns of the mixing matrix). A possible manner of estimating the mixing matrix is with the use of ICA. As it has been previously introduced, there

is a need for sparsity, meaning that the coefficients in some domain are zero or close to zero. Sparsity is related to a peaked probability distribution in any domain, and to the coefficients (values of the signal in a certain domain, such as time sample or the time-frequency bin) being concentrated around the mixing directions, which allows an easier estimation of the coefficients of the matrix.

A commonly used method for the estimation of sources is the time frequency masking. Yilmaz and Rickard [37] used this approach for the separation of speech mixtures, and it has also been used in the musical domain in several contributions, such as Vinyes et al. [38], which additionally exploit the spatial information (panning) for improving the results. This approach deals with the use of the STFT to transform a signal from the time domain to the time-frequency domain. In this domain, a mask can be used to select only certain coefficients which are supposed to correspond to the source of interest. The selection of the coefficients can be performed in a simple form with a binary mask, which sets to 0 the coefficients which are not of interest. This approach relies on the fact that there is reduced overlap between several sources in the time frequency domain [37]. The sound is then synthesized by estimating the signal in the time domain from the filtered spectrum, with the Inverse Discrete Time Fourier Transform (IDTFT). One of the drawbacks of binary time frequency masking is that it produces “artifacts” known as musical noise.

However, time frequency masking will be considered in this master thesis, as a relatively simple method, which provides fast results in comparison with NMF based source separation, and can even be used for online source separation, as in the approach recently presented by Marxer [19].

2.3.2 Fundamental frequency estimation

A melody can be defined as an organized sequence of notes and rests, where each of the notes has a pitch, an onset time, and an offset time. The melody followed by an instrument is a very important cue for many of the source separation approaches.

The knowledge of the melody of the instrument to be separated could come from the (MIDI) score of the music piece, but most usually, the fundamental frequency is estimated by computational means. The transcription of a melody is commonly performed by estimating the trajectory of the fundamental frequency (f0). Many algorithms have been proposed on the

literature for melody extraction, such as Dressler [39], or the multipitch estimation by Klapuri [40]. Marxer et al. [19] recently presented a method for low latency pitch estimation, and a technique for the detection and tracking of a pitched instrument.

The previously introduced time-frequency masking methods make use of the f0 trajectory to create the appropriate masks for the selection of the time-frequency bins where the mask is to be applied. In the case of the NMF approaches that use the source-filter model, the information about the estimated f0 trajectories is used to initialize the parameters of the source part of the model, since the information about the pitch is related to this part. In the case of the approach by Durrieu [41], the interest is on the separation of the main instrument; thus, the f0 estimation is only for the predominant instrument. In the case of Heittola [18], the interest is not just on the main instrument, but on all present instruments, and therefore, a multipitch estimation approach is necessary, in this case Klapuri's. More details about this work will be presented in the following section, dealing with instrument classification based on source separation.

2.3.3 A Flexible Audio Source Separation Framework (FASST)

FASST is a framework for source separation recently presented by Ozerov et al. [42], which aims to generalizing several existing source separation methods, and allows creating new ones. It is based on structured source models, which allow the introduction of constrains according to the available prior knowledge about the separation problem.

The framework can be used for many different use cases, such as speech separation, or for professionally produced music recordings. This framework has been considered for the thesis since the (MATLAB) source code is available, and it allows to perform the separation of audio excerpts into four different sources: drums, bass, melody (either singing voice or a leading melodic instrument), and the remaining sounds. The first step of this separation is performed by computing the time-frequency transform of the input, with the STFT or with the auditory motivated Equivalent Rectangular Bandwidth (ERB). Then, the model parameters are estimated e.g. with a Expectation Maximization algorithm, and finally the spectral components are separated, with the aid of spectral patterns for e.g. bass, drums. The interest in such separation strategy is introduced in the methodology.

2.4 Source separation based on timbral models

Timbral models based on descriptors such as MFCCs, MPEG7, or more advanced descriptors [13] have been widely used in source separation algorithms, since such models typically provide better results in musical source separation applications, as they can deal better with the separation of signals with overlapping spectrums.

Source separation systems can be classified as being supervised or unsupervised: supervised methods rely on a previous training step to estimate the models from a training database, while unsupervised systems do not need a training step. Supervised methods typically provide a better separation quality, and are able to cope with more demanding situations, but are less generic than unsupervised methods. Two unsupervised methods which have been used for source separation are the previously introduced ICA and NMF. With both methods, the magnitude spectrogram of the mixture is approximated with a weighted sum of basis functions (also named components) with a fixed spectrum. Typically, a clustering process is required to group the basis functions according to each of the sources, since each source is typically the sum of a set of basis. One of the drawbacks of ICA is that the number of sensors must be the same as the number of sources, thus not allowing solving the most demanding, but also more common tasks in which the system is underdetermined. Independent Subspace Analysis (ISA) can deal with this limitation, by performing the analysis in a transformed domain, such as the magnitude or power spectrum. NMF can also cope with underdetermined systems, and seems to provide better results than ISA [43].

As it was previously introduced, NMF introduces non negativity constrains which are appropriate when working with amplitude or power spectrograms. The algorithms proposed by Lee and Seung [28] do the decomposition by minimizing the distance between X and $\hat{X} = A \cdot S$. One of the drawbacks of this basic spectrogram decomposition is that the basis functions used for each instrument are dependent on the pitch of the note being played, and thus the amount of basis functions to be employed is large, and the estimation less reliable. In order to deal with this shortcoming, Virtanen and Klapuri [44] proposed the use of a source filter model with NMF for the analysis of polyphonic audio. In this work, the spectrogram is modeled as a sum of basis functions with varying gains, as in the normal NMF. Furthermore, each basis function is decomposed into a source and a filter. The source refers to a vibrating object and

changes with the pitch. On the other hand, the filter refers to the "resonance structure" which is constant for each instrument, thus reducing the parameters to estimate. Important for this master thesis will be the approach by Durrieu [41], in which NMF is combined with a source filter model for the separation of the singing voice from the accompaniment. In order to have an idea of the possible pitches which each instrument or voice is playing, some kind of fundamental frequency estimation algorithm is typically employed. Timbral models are also employed by Marxer [19] to identify the spectral envelopes of the target instruments, in order to improve the tracking of their pitch. This pitch is then used to generate a time frequency mask to perform the source separation. The features used are the MFCCs extracted from the Harmonic Spectral Envelope (HSE), as previously introduced.

2.5 Instrument classification based on source separation

In previous sections, the state of the art in instrument recognition and source separation has been introduced. This section presents some of the approaches found in the literature which have used source separation as a prior step to the classification of the instruments present in polyphonic mixtures.

Heittola et al. [18] presented an approach to instrument classification based on the use of NMF with a source filter model, based on the previously introduced work by Virtanen and Klapuri [44]. An overview of the system is illustrated in Figure 2.2.

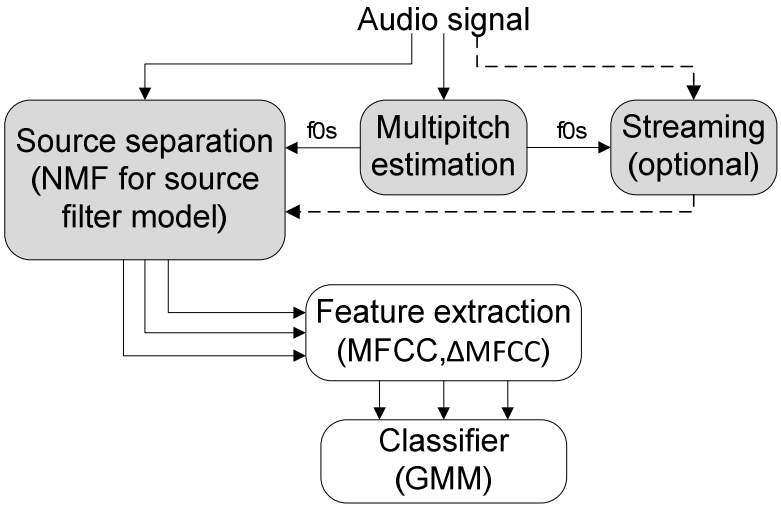


Figure 2.2: Musical instrument recognition with a source filter model for source separation by Heittola [18]

Klapuri's multipitch estimation is used to help to separate the sources, with the aid of an optional streaming algorithm which organizes individual notes into sound sources, using the Viterbi algorithm to find the most likely sequence of notes.

The instrument classification is performed by using MFCC (with a 40 channel filter bank), with their first time derivatives and a polynomial fit. GMM are used to model the instrument conditional densities of the features, and the parameters are estimated using the Expectation Maximizations (EM) algorithm from the training material. A Maximum Likelihood classifier is used for the classification. The reported F1-measure reaches 59.1% with a database of artificially created mixtures (using the RWC), and six note polyphony. The number of classes selected for the experiments was 19, including only pitched instruments.

Burred [13] also presents an instrument classification approach with a stereo Blind Source Separation pre-step, reaching 86.7% accuracy, using Gaussian likelihood as a timbre similarity measure, with a polyphony of 2 instruments, and 5 classes. The results are significantly better than in the case of monaural separation, with an accuracy of 79.8%.

Source separation is thus a useful pre-step to improve the instrument recognition in polyphonic mixtures. However, there are some problems that source separation algorithms could pose for the automatic instrument recognition: they usually add some artifacts such as musical noise, or they even alter the timbre of the instruments. These facts will be investigated in the course of this master thesis, and some solutions proposed, as detailed in the methodology.

3 Methodology

The hypothesis of this work is that after understanding and acknowledging the limitations of current state-of-the-art algorithms in source separation and automatic instrument recognition, it is possible to effectively combine them in order to enhance their results. Such synergies will be investigated following the methodology presented in this chapter.

As previously introduced, instrument recognition algorithms usually generate a set of tags corresponding with the (most predominant) instruments in an audio excerpt. On the other hand, source separation algorithms can provide several kinds of outputs, depending on the intended application: some estimate the contributions of the each of the instruments to the input mix, or the harmonic+percussive components, or for instance, the predominant instrument+ accompaniment. The usual application of both types of algorithms is presented in Figure 3.1.

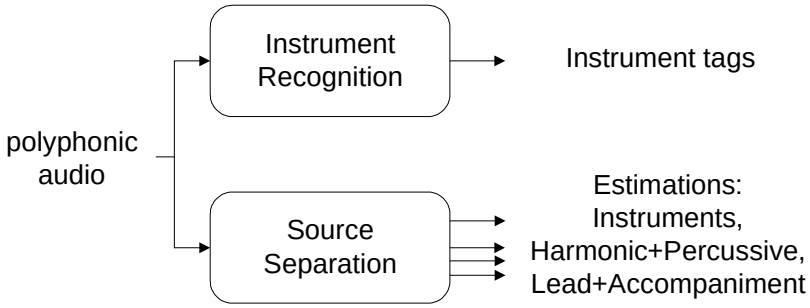


Figure 3.1: Instrument Recognition and Source Separation application

The first section in this chapter deals with the methodology proposed to enhance the source separation with a previous instrument recognition step, and the second section details the methodology proposed to enhance the instrument recognition with a previous source separation step. It is important to note that the main interest is the application of the algorithms to professionally produced western music recordings, as opposed to audio data created by artificially mixing several instrument with no musical relation between them, as used in [18]. This fact adds more complexity to the source separation algorithms, due to several reasons. Firstly, if there is a high melodic and harmonic relation between the instruments, their spectral components usually share same frequencies, which pose problems to the source separation algorithms. Additionally, in professionally produced music recordings, the typically applied effects such as reverbs, delays, etc. make the separation much more difficult. On the other hand,

such scenario allows source separation and instrument recognition algorithms to make use of the panning information, which typically allows achieving a better separation quality.

3.1 Instrument recognition for source separation

The problem of interest in this section is the separation of a music recording into the instrument of interest and the accompaniment. Many separation algorithms produce errors due to an imperfect matching of the spectral components to the instrument to be removed, causing the removal of other instruments, instead of the one of interest. A similar problem is that, even when the instrument of interest is not present, other instruments are removed. The hypothesis is that the integration of an instrument recognition algorithm prior to the source separation can enhance the quality of the separation, by avoiding the separation to take place in the parts of the audio in which the instrument of interest is not present. The proposed way for the integration is presented in Figure 3.2.

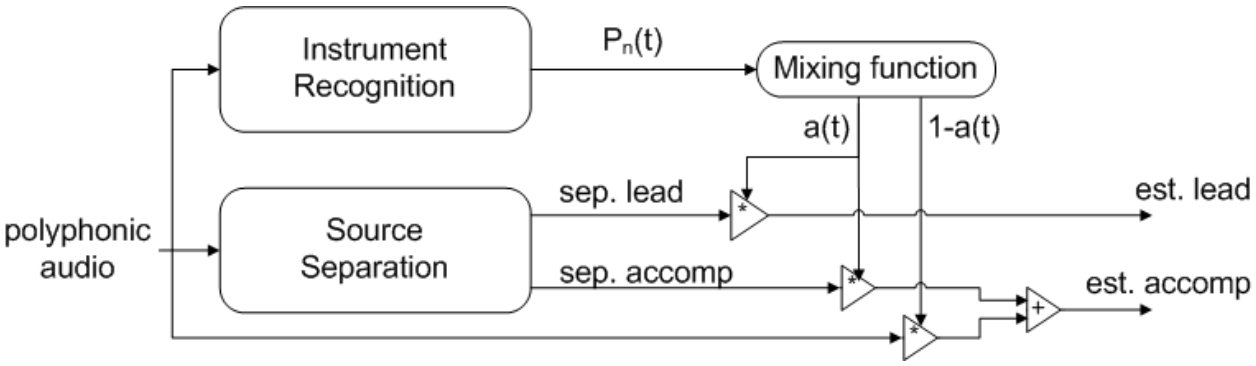


Figure 3.2: Schema of the integration of instrument recognition into the source separation algorithm. The instrument recognition module gives as output the probabilities of presence of the n classes used in the classifier. These probabilities are then used to compute the mixing weights to be applied to the output of the source separation algorithm.

The instrument recognition module is based on the system by Fuhrmann [4], [22], depicted in Figure 2.1. The input of this system is polyphonic audio, and the output is the probabilities of presence $P_n(t)$ of n=11 different instruments (cello, clarinet, flute, acoustic guitar, electric guitar, hammond organ, piano, saxophone, trumpet, violin, and voice). The original system uses one vs. one SVM classifiers and Pair Wise Coupling to combine the probabilities. However, in this scenario the focus should be set in detecting the presence of the instrument of interest in the polyphonic mixture. Therefore, it was decided that it would also be appropriate to train a binary classifier with n = 2 classes: the instrument of interest, and a second artificial class, containing

the rest of instruments. Since the voice is typically the most important element of a music recording, and there are applications which would benefit of the voice removal, such as karaoke, it was decided to initially focus on it. The procedure followed for the training of the SVM classifier of the instrument recognition algorithm is detailed in the next subsection.

Several source separation algorithms have been considered with the following output: lead instrument + accompaniment. In order to reduce the errors produced by the separation algorithms, the probabilities of presence are used to remix the separated components, by computing the weight a(t) with a mixing function. When the probability of presence of the instrument of interest is high, the weight tends to 1, while with a low probability of presence, the weight tends to 0. The mixing function can be either binary or non binary. In the first case, the mixing coefficients are {0,1}, and in the second case, intermediate values are allowed, in the range [0,1]. The binary mixing logic is presented in the following equations:

$$P_{voice}(t) >= P_i(t), \forall i \neq voice \Rightarrow a(t) = 1 \tag{3.1}$$

$$P_{voice}(t) < P_i(t), \forall i \neq voice \Rightarrow a(t) = 0 \tag{3.2}$$

$P_{voice}(t)$ represents the probability of the voice being present in the excerpt, and $P_i(t)$ represents the probability of presence each of the classes i considered. If a binary classifier is used, the classes considered are: voice or non-voice (n = 2), while in the original non binary classifier, the number of classes corresponds to each of the considered instruments (n = 11). Note that it is possible to use a binary classifier with non binary mixing.

As previously mentioned, the non-binary mixing allows having values between 0 and 1, and thus the effect is more subtle. This kind of mixing would be appropriate when the probability of presence of the instrument of interest is not very high, or when the results of the instrument recognition algorithm cannot be fully trusted.

The final estimation of the voice provided by the integrated system (*est. lead*) tends to the estimation produced by the separation algorithm (*sep. lead*) if the $P_{voice}(t)$ is high, and tends to zero if the $P_{voice}(t)$ is low. On the other hand, the final estimation of the accompaniment (*est. accomp*) tends to the estimation produced by the separation algorithm (*sep. lead*) if the probability of presence is high, and tends to the input mix if the probability of presence is low.

Note that in the case of a perfect recognition of the presence of the instrument of interest (in our case, the voice), the quality of the separated audio is perfect in the parts of the input audio excerpt in which no voice is present. However, in the parts with a voice the quality is limited by the quality of the separation algorithm.

3.1.1 Data

Two different sets of data have been used for training and testing. Firstly the SVM classifier of the instrument recognition algorithm has been trained with a large set of data, and secondly, the combination of both modules has been evaluated with a reduced set of multitrack data.

The original training data of the instrument recognition module was a large set of features extracted from short excerpts of audio from different western music genres (pop, rock, jazz, classical), which included the annotation with the predominant instrument, as described in [4]. This set of features includes typical features in the description of timbre, such as the MFCCs, along with other features, such as HPCP (Harmonic Pitch Class Profiles), LPC, inharmonicity, spectral moments, crest, rolloff, and RMS energy features, and their time derivatives [22]. As described before, the focus was set on the voice, and an additional binary classifier was trained to classify audio excerpts into: voice and non-voice. The training process began thus with the division of the training instances into these two classes. Since the non-voice training data had a larger number of instances in the original dataset compared to the voice, a sub-sampling of the original collection of non-voice data was performed, in order to even up their sizes. Additionally, care was taken to select a similar number of samples of each of the instruments in the original collection, so as to have a quite uniform distribution. Finally, the number of training instances was around 800 for each of both classes.

The testing data were extracted from 6 multitrack recordings of professionally produced music. The excerpts in the created testing dataset have a lead instrument (in this case, vocals) which is present in a segment of the excerpt, and absent in another part of the excerpt. As explained in the following subsection, it is necessary that the testing dataset includes the original separated tracks, or at least the vocals + accompaniment, since the evaluation is based in comparing the output of the separation algorithms against this ground truth.

3.1.2 Evaluation method

For the evaluation of the quality of the source separation, the SiSEC evaluation campaign measures have been used. This campaign deals with speech and music datasets, synthetic mixtures, microphone recordings and professional mixtures and divides the source separation problem into four tasks: source counting, mixing system estimation (mixing gains), source signal estimation (mono source signals), and source spatial image estimation (contribution of each source to the two mixture channels). The objective evaluation measures proposed deal with the comparison of the estimates of the spatial images of the source j in the channel i $\hat{s}_{ij}^{img}$, against the true source images s_{ij}^{img}. The first step deals with the decomposition of $\hat{s}_{ij}^{img}$ into the target source and distortion:

$$\hat{s}_{ij}^{img} = s_{ij}^{img} + e_{ij}^{spat} + e_{ij}^{\mathrm{int}erf} + e_{ij}^{artif} \tag{3.3}$$

Three kind of distortions are considered: e_{ij}^{spat}, $e_{ij}^{\mathrm{int}erf}$ and e_{ij}^{artif} which correspond to the spatial (or filtering) distortion, the interference, and artifacts respectively. This decomposition is motivated by the distinction in the auditory system between the sound from the target source: $s_{ij}^{img} + e_{ij}^{spat}$, the sounds from other sources: $e_{ij}^{\mathrm{int}erf}$, and the gurgling or musical noise, e_{ij}^{artif} as presented by Vincent [45]. Three energy ratios are derived from this decomposition: Source Image to Spatial distortion Ratio (ISR), which considers the sounds from the target source, Source to Interference Ratio (SIR), which considers sounds from other sources and Sources to Artifacts Ratio (SAR), related to the “gurgling noise” or other artifacts due to the separation. The ratios are defined by the following equations:

$$ISR_j = 10\log_{10}\frac{\sum_{i=1}^{I}\sum_t s_{ij}^{img}(t)^2}{\sum_{i=1}^{I}\sum_t e_{ij}^{spat}(t)^2} \tag{3.4}$$

$$SIR_j = 10\log_{10}\frac{\sum_{i=1}^{I}\sum_t (s_{ij}^{img}(t) + e_{ij}^{spat}(t))^2}{\sum_{i=1}^{I}\sum_t e_{ij}^{\mathrm{int}erf}(t)^2} \tag{3.5}$$

$$SAR_j = 10\log_{10}\frac{\sum_{i=1}^{I}\sum_t (s_{ij}^{img}(t) + e_{ij}^{spat}(t) + e_{ij}^{\text{int}erf}(t))^2}{\sum_{i=1}^{I}\sum_t e_{ij}^{artif}(t)^2} \tag{3.6}$$

Additionally, the Source to Distortion Ratio (SDR) is a compound of all previous measures:

$$SDR_j = 10\log_{10}\frac{\sum_{i=1}^{I}\sum_t s_{ij}^{img}(t)^2}{\sum_{i=1}^{I}\sum_t (e_{ij}^{spat}(t) + e_{ij}^{\text{int}erf}(t) + e_{ij}^{artif}(t))^2} \tag{3.7}$$

The results of the previously presented energy ratios are in decibels (dB). A higher value means that the quality of the separation is better. If the values are negative, it means that there is more distortion than source components. Usually, values range between 0 and 20 dB, but this depends on the algorithm, the data, and the energy ratio, as presented in the experiments section.

As previously mentioned, in order to compute these measures, it is necessary to have the original data with the vocals and accompaniment tracks (which serve as a reference). A dataset of excerpts of 6 songs has been created, with the original set of audio excerpts (vocals, accompaniment, mix), and the annotation of the presence of the lead instrument. Additionally a ground truth annotation of the instrument recognition algorithm output has been manually created for each of the files, in order to be able to measure the upper limit in separation quality that the combination of both algorithms would have. If the instrument recognition was perfect, the separation would be improved in the segments where there is no lead instrument, and would remain the same in the segments of the excerpt where a lead instrument is present.

A further possibility was to use the Perceptual Evaluation methods for Audio Source Separation (PEASS) toolkit as proposed by Emiya et al. [46]. This software allows the calculation of objective measures to assess the perceived quality of estimated source signals, based on perceptual similarity measures obtained with the PEMO-Q auditory model [47]. The PEASS toolkit is distributed under the terms of the GNU Public License version 3, but a fee is required for the use of PEMO-Q, so this option was finally not considered.

3.2 Source Separation for instrument recognition

The interest in this part of the thesis is to improve the recognition of pitched instruments in polyphonic audio. The algorithm used by Fuhrmann in [4] is considered as the instrument

recognition method to be enhanced. This algorithm is conceived to output a set of labels corresponding with the most predominant instruments in an excerpt of polyphonic music. The interest is focused on the following pitched instruments: cello, clarinet, flute, acoustic guitar (acguitar), electric guitar (eguitar), organ, piano, saxophone, trumpet, violin, and also the voice. A problem found in this system is that it too often misses some of the instruments in the case that the piece has more than one predominant instrument.

The hypothesis is that in order to enhance its performance, a previous step could be performed, in which the input audio data is separated into several streams. These streams would then be processed by the instrument recognition algorithm separately, which would output several labels.

Several separation processes can be considered, as well as different strategies for the label combination, and also several models used in the instrument recognition. The schema in Figure 3.3 illustrates the combination of a separation process followed by the instrument recognition:

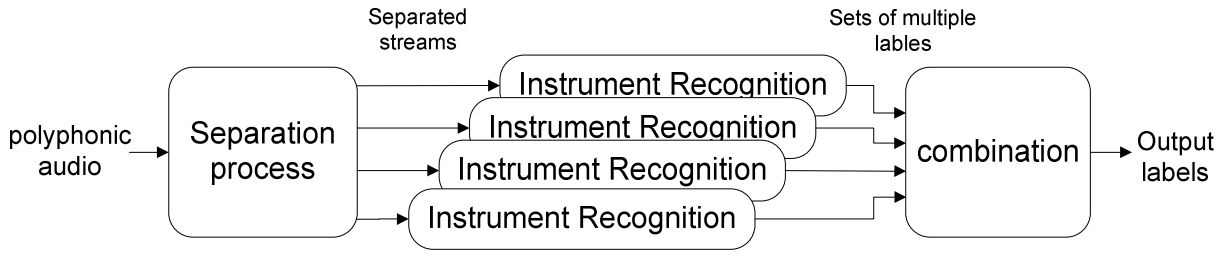


Figure 3.3: Generic schema of the application of source separation as a previous step to the instrument recognition.

In this work, the previously introduced FASST algorithm is used to separate the input polyphonic audio into “drums”, “bass”, “melody”, and “other” streams. This separation algorithm was selected, because of the characteristics of the instrument recognition algorithm which is used in combination with it, as it does not consider the bass or drums for the classification. In the case of the ideal source separation, the “melody” stream would contain the main instrument to be recognized, and the “other” stream would contain the rest of the instruments, with no presence of bass and percussive instruments. The recognition of instruments in these streams of audio, with no presence of drums or bass should be easier than in the case of the polyphonic mixture.

However, a common limitation found in most source separation algorithm is that they produce artifacts and errors in the separation, producing some leakage of instruments in

estimations where they should not be present. This leakage may affect the recognition of instruments due to the changes in timbre it produces.

It is thus interesting to investigate in the course of this work if a classifier could learn how a source separation algorithm behaves, by training models on the separated audio estimations. In this case, what the models would learn could be expressed in simple words as: these are the features of the estimated {“drums”, “bass”, “melody”, “other”} stream, when the predominant instrument of the audio is a {cello, clarinet, flute, acoustic guitar, electric guitar, organ, piano, saxophone, trumpet, violin, or voice}. The use of different models for each of the separated streams would allow the usage of a different set of (automatically selected) features, as well as different parameters for training the classifiers, as further detailed in Chapter 4. The following subsection details the data used for training and for testing the proposed system.

3.2.1 Data

Two different datasets have been created, based on the database originally compiled by Fuhrmann in [4], [22]. Firstly, the training dataset contains annotated short excerpts of 3 seconds duration in which only one instrument is predominant. There is a total amount of around 6700 excerpts, with a minimum number of excerpts corresponding to each of the instruments of 388, and a maximum of 778. Secondly, the testing set was created, with around 6500 excerpts annotated with one to three predominant instruments. This set was created by dividing the original music pieces of Fuhrmann’s database into segments with the following properties:

- The predominant instruments are the same throughout all the excerpt
- The excerpts are between 5 and 20 seconds long. Shorter segments are discarded, and longer segments are divided into segments with a length in this range
- The excerpts are in stereo

The first property deals with the fact that the predominance of instruments in a music piece typically changes amongst or even within sections. This property allows not considering the segmentation of the songs into the evaluation of the instrument recognition. The second property deals with two different issues. Firstly, the 20 second limitation is due to the memory limitations in Matlab. Since some of the algorithms used have been implemented in Matlab, it was necessary to divide the longer segments into smaller segments. Secondly, the 5 second limitation is in order to ensure that the instrument labeling process has enough information to output the labels with a

certain confidence. The third and last property corresponds to the use case of interest, which in this case are professionally produced music recordings, in stereo format.

3.2.2 Evaluation method

The evaluation method compares the labels outputted by the algorithm against the ground truth in the annotation files, and computes the confusion matrix (as in the traditional information retrieval evaluation), by calculating the true positives (tp), true negatives (tn), false positives (fp) and false negatives (fn) for each of the instruments (labels). We consider L the closed set of labels $L = \{l_i\}$, with $i = 1...N$, where N is the number of instruments, and the dataset $X = \{x_i\}$, with $i = 1...M$, where M is the number of excerpts. We define $\hat{Y} = \{\hat{y}_i\}$, with $i = 1...M$ as the set of ground truth labels, and $Y = \{y_i\}$, with $i = 1...M$, and $y_i \subseteq L$, the set of predicted labels assigned to each instance i. Precision and recall are defined for each of the labels l in L as:

$$P_l = \frac{tp_l}{tp_l + fp_l} = \frac{\sum_{i=1}^{M} y_{l,i}\hat{y}_{l,i}}{\sum_{i=1}^{M} y_{l,i}}, \text{ and } R_l = \frac{tp_l}{tp_l + fn_l} = \frac{\sum_{i=1}^{M} y_{l,i}\hat{y}_{l,i}}{\sum_{i=1}^{M} \hat{y}_{l,i}} \tag{3.8}$$

where, $y_{l,i}$ and $\hat{y}_{l,i}$ are boolean variables referring to the specific instance i, which indicate the presence of the label l in the set of predicted labels, or in the set of ground truth labels respectively. Additionally, we define the F1-measure, as the harmonic mean between precision and recall:

$$F_l = \frac{2P_lR_l}{P_l + R_l} \tag{3.9}$$

Furthermore, macro and micro averages of the previous metrics are defined, in order to obtain more general performance metrics, which consider all labels. The macro is here understood as an unweighted average of the precision or recall taken separately for each label (average over labels).

$$P_{macro} = \frac{1}{|L|}\sum_{l=1}^{L} P_l \text{ , and } R_{macro} = \frac{1}{|L|}\sum_{l=1}^{L} R_l \tag{3.10}$$

On the other hand, the micro average is an average over instances, and thus, labels with a higher number of instances have more weight in the computation of the average measures.

$$P_{micro} = \frac{\sum_{l=1}^{L} tp_l}{\sum_{l=1}^{L}\left(tp_l + fp_l\right)} = \frac{\sum_{l=1}^{L}\sum_{i=1}^{M} y_{l,i}\hat{y}_{l,i}}{\sum_{l=1}^{L}\sum_{i=1}^{M} y_{l,i}}, \tag{3.11}$$

$$R_{micro} = \frac{\sum_{l=1}^{L} tp_l}{\sum_{l=1}^{L}\left(tp_l + fn_l\right)} = \frac{\sum_{l=1}^{L}\sum_{i=1}^{M} y_{l,i}\hat{y}_{l,i}}{\sum_{l=1}^{L}\sum_{i=1}^{M} \hat{y}_{l,i}} \tag{3.12}$$

The macro and micro F1-measures are defined as the harmonic mean of respectively, the macro and micro averages.

$$F_{macro} = \frac{2P_{macro}R_{macro}}{P_{macro} + R_{macro}},\ F_{micro} = \frac{2P_{micro}R_{micro}}{P_{micro} + R_{micro}} \tag{3.13}$$

The following chapter details the experiments executed based on the presented methodology.

4 Experiments and results

After the implementation of the algorithms introduced in the methodology, several experiments have been designed and conducted. Firstly, the experiments dealing with instrument recognition for source separation are presented, and secondly the experiments focused on the source separation for instrument recognition are introduced.

4.1 Instrument recognition for source separation

Several experiments have been conducted, based on the schema of Figure 3.2. First, the source separation algorithms have been evaluated without the integration of the instrument recognition. In the case of Marxer's algorithm [19] introduced in section 2.3, two different experiments have been conducted in order to test the benefit of using Pan-Frequency filtering, which allows restricting the range in the stereo field and the spectrum in which the fundamental frequency of the voice is searched for. Another test has been performed to evaluate how useful the internal voice models of Marxer's approach are. With internal voice models it is meant the SVM model trained with the MFCC's of the spectrum envelope of the voice, as introduced in section 2.1.1. When the voice models are not used, the algorithm always performs separation of the most predominant instrument, independently of which one it is.

Then, the FASST algorithm introduced in section 2.3.3 is employed to estimate the melody+accompaniment, where the accompaniment contains the *bass+drums+other* components. Finally, the algorithm based on source/filter separation by Durrieu [41] is also evaluated. The experiments conducted without the integration of the instrument recognition are:

- *Marxer_noPF*: No Pan-Frequency filtering, with internal voice models
- *Marxer_PF*: With Pan-Frequency filtering, with internal voice models
- *Marxer_NoVoiceModel*: With Pan-Frequency filtering, no internal voice models
- *FASST*
- *Durrieu*

All following experiments with Marxer's approach used Pan-Frequency filtering, and internal voice models, since they provided the best results.

Secondly, the upper limit of the combination of the instrument recognition and source separation has been investigated, by using the ground truth instrument recognition:

- *IR_Marxer_GT*
- *IR_FASST_GT*
- *IR_Durrieu_GT*

Then several experiments have been conducted to test different implementations of the combination of the instrument recognition with Marxer's separation approach. Four combinations were tested, depending on the use of binary classification or not, and depending on the use of binary mixing or not:

- *IR_Marxer_binClassif_binMix*
- *IR_Marxer_notbinClassif_notbinMix*
- *IR_Marxer_binClassif_notbinMix*
- *IR_Marxer_notbinClassif_binMix*

The integration was also tested on other state-of the art separation algorithms, in order to investigate the improvement in the quality of the results. Since the best combination found for the integration was the use of a binary classification and a not binary mixing strategy, only these results are reported here:

- *IR_FASST_binClassif_notbinMix*
- *IR_Durrieu_binClassif_notbinMix*

At last, in the *No Separation* experiment, the output estimated vocals is equal to the original mix with 24dB attenuation, and the estimated accompaniment was the original mix. Thus no separation has taken place. This experiment was conceived to test the adequacy of the evaluation measures used. Table 4.1 contains the results of the previously introduced experiments.

Additionally, the figures in the Annex A show the detailed SDR per song, for each of the experiments performed. In the first figure, the ground truth is the accompaniment, and in the second the vocals. Some of the songs are clearly more difficult for the algorithms to separate. For instance, the first song corresponds to an Alanis Morissette song excerpt in which the vocals are very soft, which certainly poses problems to both the source separation and instrument recognition algorithms. Figure 4.1 and Figure 4.2 show the output probabilities in this challenging excerpt, for both the multiclass and binary classifier, showing the benefits of the

binary classifier. The voice, only present in the second half of the excerpt is better identified when a binary classifier is used.

Some conclusions can be obtained by analyzing all the data in Table 4.1. A first conclusion is that filtering the input mix in pan and frequency with Marxer's approach, some quality improvement can be achieved, as seen in *MarxerPF* experiment. A second conclusion is that the results obtained by both Durrieu and FASST approaches are better than Marxer's. The difference in quality is mostly due to the Signal to Artifacts Ratio, which is worse in Marxer's approach. This is reasonable since this is approach is based on time frequency masking, which usually produces more artifacts than other approaches such as NMF, but the advantage is that it is much faster, which is the main interest of Marxer's approach. Additionally, the online approach estimates the fundamental frequency in each frame, instead of using the information of the whole excerpt to calculate the fundamental frequency trajectory, what makes the process faster, but typically less reliable.

Table 4.1: SiSEC quality measures for different experiments in the own database of multitrack data. The following abbreviations have been used: PF – use of Pan-Frequency filtering, IR – combination of the source separation with Instrument Recognition, GT - combination of the source separation with a Ground Truth instrument recognition, binClassif: use of binary classification, binMix: use of binary mixing.

Experiment	Lead (vocals)				Accompaniment			
	SDR	SIR	ISR	SAR	**SDR**	SIR	ISR	SAR
Marxer_noPF	**0.37**	3.99	12.02	1.77	**7.41**	17.58	13.27	7.95
Marxer_PF	**0.89**	5.84	9.48	0.78	**7.93**	15.32	16.29	9.11
Marxer_NoVoiceModel	**0.05**	4.31	10.54	0.63	**6.47**	17.31	13.04	6.84
FASST	**2.17**	2.81	6.32	5.38	**9.16**	12.36	14.79	14.48
Durrieu	**3.00**	6.90	9.12	4.25	**10.00**	15.45	16.96	12.80
IR_Marxer_GT	**2.53**	10.97	9.47	1.43	**9.56**	15.95	21.36	11.24
IR_FASST_GT	**2.94**	6.09	6.32	4.34	**9.97**	12.78	18.12	14.84
IR_Durrieu_GT	**5.02**	14.10	9.60	5.61	**12.01**	16.32	23.49	14.74
IR_Marxer_binClassif_binMix	**1.79**	9.94	6.66	-0.13	**8.82**	13.19	22.13	11.57
IR_Marxer_notbinClassif_notbinMix	**1.19**	9.01	3.36	-2.97	**8.22**	10.12	27.79	16.01
IR_Marxer_binClassif_notbinMix	**2.19**	9.03	6.16	0.52	**9.23**	12.63	21.64	12.45
IR_Marxer_notbinClassif_binMix	**1.10**	7.11	4.33	-2.90	**8.14**	10.96	24.64	13.38
IR_FASST_binClassif_notbinMix	**2.72**	4.73	5.36	3.94	**9.75**	11.84	19.04	16.30
IR_Durrieu_binClassif_notbinMix	**3.71**	11.44	6.04	3.78	**10.72**	12.74	23.53	15.64
No Separation	**0.46**	0.56	-6.96	49.06	**7.03**	36.28	7.04	218.57

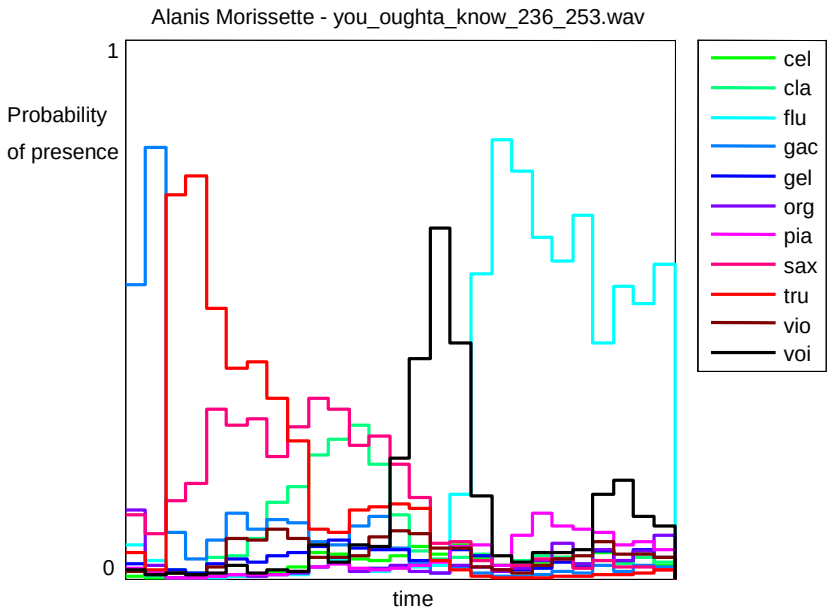


Figure 4.1: Probability of presence of the eleven instruments (including voi – the voice), given as output of the non binary classifier. A flute is detected at the end of the excerpt where a voice should be detected.

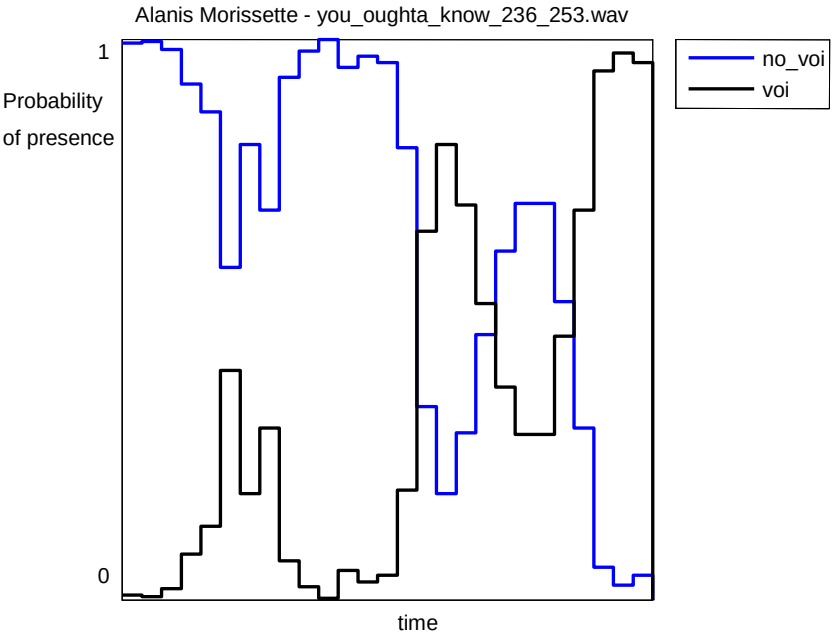


Figure 4.2: Probability of the two classes of the binary classifier: voice, and non-voice. The binary classifier provides more accurate recognition results.

The third conclusion is that the combination of instrument recognition with source separation provides an improvement in quality. In the case of a perfect recognition of the presence of the lead instrument (by using the manually created ground truth), the tables show a clear increase in the performance of the source separation algorithms, as it can be seen in experiments containing GT. In fact, if Marxer's algorithm is combined with the instrument recognition, it can potentially achieve similar separation quality levels than *Durrieu* or *FASST*, as shown in the experiment *IR_Marxer_GT*. As it can be observed in Table 4.1, the quality is increased by 2.6dB in the case of vocals, and by more than 1.5 dB for the accompaniment, with respect to *MarxerPF*. The experiments with the different combinations of source separation with the instrument recognition provide different results. The best results are obtained with a non binary mixing, and binary classification, achieving a SDR only 0.3dB lower than the upper limit, obtained the ground truth instrument recognition, as it can be observed when comparing the results of the experiments: *IR_Marxer_binClassif_notbinMix* and *IR_Marxer_GT*.

If the instrument recognition is applied to other separation algorithms, their separation quality is also potentially enhanced with the ground truth instrument recognition, but also with the real algorithm (*IR_FASST_binClassif_notbinMix* and *IR_Durrieu_binClassif_notbinMix*).

The *No Separation* experiment seems to provide evidence that it does not make sense to consider the SDR as a unique measure to evaluate the quality of the source separation. Even though no separation at all has taken place, the SDR for both vocals and accompaniment is very similar to the one obtained with Marxer. This is due to the fact that in the *No Separation* experiment; the SAR is excellent, since by definition the original mix has no artifacts. However, the SIR is very bad, especially for the vocals, since the rest of the mix is acting as interference. And also in the accompaniment, the complete presence of the vocals reduces the SIR to its minimum possible value.

4.2 Source separation for instrument recognition

Four different experiments have been conducted to investigate the benefits of the separation of the audio signal into different streams prior to the application of an instrument recognition algorithm. Firstly, the original algorithm proposed by Fuhrmann in [4] was used to identify the instrument present in polyphonic audio (Experiment 1). Secondly, the same system was then applied to the recognition of the instruments present in four different streams of audio,

corresponding to the estimations of the bass, drums, melody and other sources by the previously introduced FASST separation algorithm (Experiment 2). Then, four SVM models were trained on the separated audio outputted by FASST, and then used for the labeling (Experiment 3). Then, a simple separation of the polyphonic audio into left, right, mid and side (LRMS) streams was used as input to the instrument recognition algorithm with the original model.

Initially, the strategy for combining the labels in the four first experiments (in the case of using more than one model) was a simple union of the predicted labels. The SVM models were initially trained with the same parameters in the four experiments: the ones from the original recognition system, which optimized the performance in Experiment 1. The model used was a polynomial kernel, of degree 4, and a cost parameter = 0.1.

Then, an additional set of experiments were designed to try to optimize the performance, by using different label combination strategies, and tuning the SVM parameters for each of the models (Experiment 5).

4.2.1 Experiment 1: original algorithm

The first conducted experiment (Experiment 1) was to evaluate the original algorithm by itself, without any previous separation step as shown in Figure 4.3:

Figure 4.3: Original instrument recognition algorithm without previous separation.

The labels obtained in this experiment are named “n” from: “no separation”. The results are included in Table 4.2.

4.2.2 Experiment 2: FASST separation + original models

In this experiment, the FASST (bass, drums, melody and other) separation is used, along with the original models for the instrument recognition, as shown in Figure 4.4:

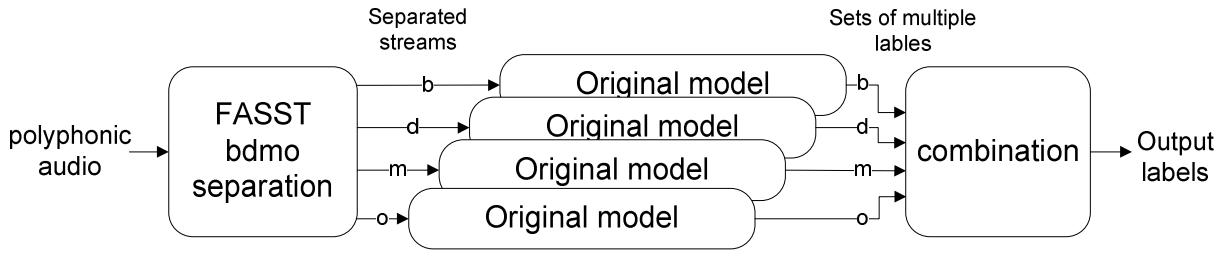


Figure 4.4: FASST separation into the drum, bass, melody and other streams, combined with the instrument recognition using the original models.

Different combinations of the labels have been tried in order to investigate which combination produces the better recognition results, e.g: dbmo means that the measures have been computed with the output labels which are the aggregation of the labels outputted in: d (drums) + b (bass) + m (melody) + o (other). Table 4.2 shows the results for the micro and macro averages of the precision, recall and F1-measure in this experiment.

Table 4.2: Results of the combination of source separation with instrument recognition, with the original models. The combinations of 3 labels have been omitted in this table, since they were not significant

	MacPrec	MacRec	MicPrec	MicRec	MacF1	MicF1
dbmon	0.336	0.455	0.373	0.492	0.387	0.424
dbmo	0.310	0.370	0.330	0.385	**0.337**	**0.355**
dbmn	0.363	0.438	0.405	0.474	0.397	0.437
dbon	0.341	0.406	0.391	0.461	0.371	0.423
dmon	0.385	0.403	0.391	0.409	0.394	0.399
bmon	0.356	0.419	0.389	0.432	0.385	0.409
db	0.294	0.199	0.371	0.273	0.238	0.315
dm	0.412	0.263	0.369	0.256	0.321	0.302
do	0.354	0.223	0.336	0.231	0.274	0.274
dn	0.483	0.333	0.510	0.352	0.394	0.417
bm	0.365	0.268	0.360	0.253	0.309	0.297
bo	0.301	0.231	0.328	0.233	0.261	0.273
bn	0.530	0.330	0.513	0.362	0.407	0.424
mo	0.364	0.241	0.320	0.186	0.290	0.236
mn	0.447	0.336	0.502	0.308	0.383	0.382
d	0.359	0.123	0.395	0.164	**0.183**	**0.232**
b	0.269	0.098	0.360	0.138	**0.144**	**0.200**
m	0.308	0.184	0.381	0.133	**0.230**	**0.197**
o	0.335	0.151	0.321	0.116	**0.209**	**0.170**
n	0.578	0.249	0.708	0.258	**0.349**	**0.378**

The results show that the original algorithm without source separation (labeled with n) provides better results than any of the combinations of the dbmo labels, obtained by means of the separation into streams. This decrease in the performance probably occurs due to the fact that the separation is not perfect, there is some energy of instruments in streams where there should not be, and their timbre is modified. With the separated audio, the best results of the F1-measure, not using the original labels (n) are obtained with the combination of all separated tracks: (drums+bass+melody+other) d+b+m+o. In this case, the recall is better than with the original algorithm (n), but the precision is quite worse, so the F1-measure is lower.

4.2.3 Experiment 3: FASST separation + models trained with separated audio

The schema of the conducted experiment is shown in the following figure:

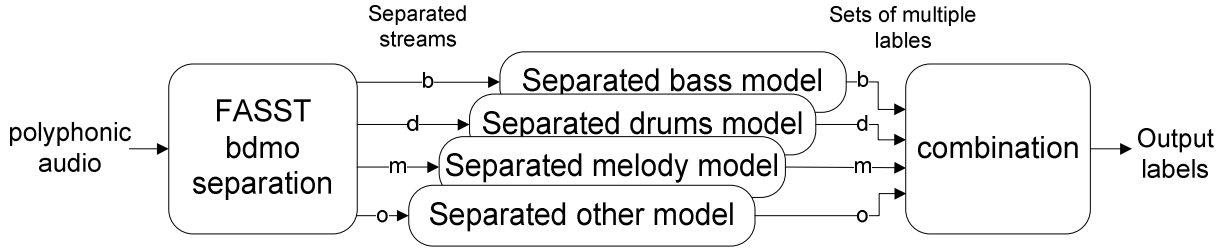


Figure 4.5: FASST separation into the drum, bass, melody and other streams, combined with the instrument recognition using models trained on the separated audio.

In this case, the models used for the classification in the instrument recognition module have been trained with the separated audio, as described in Chapter 3. Four different models have been created, one for each of the output streams of the FASST bdmo separation algorithm. Each of the models makes use of a different set of features, selected automatically during the training process. Table 4.3 shows the results for this experiment. Note that the Experiment 1 results is also included in the table, in row: n (no separation), for an easier comparison. Since it would also be possible to add the “n” label to the “bdmo” sets of labels, these combinations are also included in Table 4.3.

Table 4.3: Experiment 3 results, showing that using the models trained on separated data provides better results than using the original models. The combinations of 3 labels have been omitted in this table.

	MacPrec	MacRec	MicPrec	MicRec	MacF1	MicF1
dbmon	0.475	0.373	0.593	0.403	**0.418**	**0.480**
dbmo	0.490	0.306	0.625	0.347	**0.377**	**0.446**
dbmn	0.493	0.343	0.614	0.371	0.405	0.462
dbon	0.492	0.336	0.613	0.363	0.399	0.456
dmon	0.491	0.366	0.612	0.395	0.419	0.480
bmon	0.493	0.357	0.614	0.382	0.414	0.471
dm	0.544	0.252	0.692	0.287	0.345	0.406
do	0.518	0.210	0.680	0.254	0.299	0.370
dn	0.543	0.287	0.670	0.308	0.375	0.422
bm	0.550	0.218	0.690	0.241	0.312	0.357
bo	0.503	0.172	0.662	0.202	0.256	0.310
bn	0.546	0.269	0.671	0.282	0.360	0.397
mo	0.547	0.267	0.684	0.294	0.359	0.411
mn	0.539	0.312	0.666	0.331	0.395	0.442
d	0.554	0.131	0.740	0.166	**0.212**	**0.271**
b	0.482	0.058	0.641	0.067	**0.103**	**0.122**
m	0.594	0.199	0.740	0.219	**0.299**	**0.338**
o	0.551	0.148	0.707	0.172	**0.233**	**0.276**
n	0.578	0.249	0.708	0.258	**0.349**	**0.378**

The “n” labels provide a Micro F1 of 0.378. The combination of the “m” and “o” labels already improves the results, obtaining a Micro F1 equal to 0.411. The best micro F1-measure obtained without the “n” labels is 0.446, by combining the labels outputted by the four different models: dbmo. If the “n” labels are also combined, the F1-measure increases to 0.480.

The true positives, true negatives, false positives, false negatives, and the derived measures (precision, recall and F1) obtained for each of the instrument in this experiment, with the dbmo (drums+bass+melody+other) configuration are shown in Table 4.4:

Table 4.4: Performance per instrument in bdmo with the models trained on the separated data of each stream

	cello	clarinet	flute	acguitar	eguitar	organ	piano	saxophone	trumpet	violin	voice
tp	36	6	36	139	268	78	241	107	51	56	568
tn	2418	2547	2425	2060	1687	2180	1683	2245	2426	2384	1586
fp	128	48	96	98	79	126	49	108	69	87	64
fn	69	50	94	354	617	267	678	191	105	124	433
prec	0.22	0.11	0.27	0.59	0.77	0.382	0.831	0.498	0.425	0.392	0.899
rec	0.34	0.11	0.28	0.28	0.3	0.226	0.262	0.359	0.327	0.311	0.567
F1	0.27	0.11	0.27	0.38	0.44	0.284	0.399	0.417	0.370	0.347	0.696

Figure 4.6 illustrates visually the recognition performance per instrument, when the models used for the recognition have been trained with the separated data.

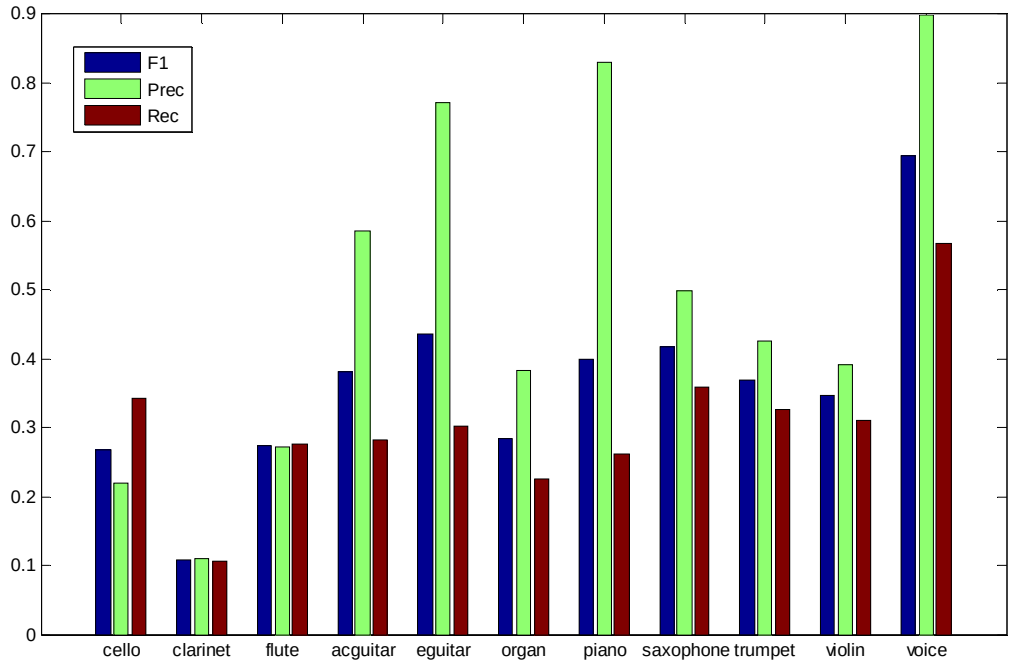


Figure 4.6: Recognition performance for each of the instruments with a previous FASST bass, drums melody and other separation, and with the models created specifically for the separated data in Experiment 3

The best results are obtained with the voice, achieving a 0.90 precision, 0.57 recall and 0.70 F1-measure. The clarinet seems to be the most challenging instrument to be recognized, with a F1-measure of about 0.11. A further observation is that the performance of the instrument recognition depends on the stream and the model. For instance, the recognition in the bass stream was better for those sounds with low frequency content, such as the excerpts containing a cello, while they were not so well recognized in the rest of the streams.

4.2.4 Experiment 4: Left-Right, Mid-Side (LRMS) separation + original models

In this experiment, the audio was separated into four streams in a very simple manner, with l = Left, r = Right, n = Left+Right (the Mid), and s = Left-Right (the Side), and the original model was used.

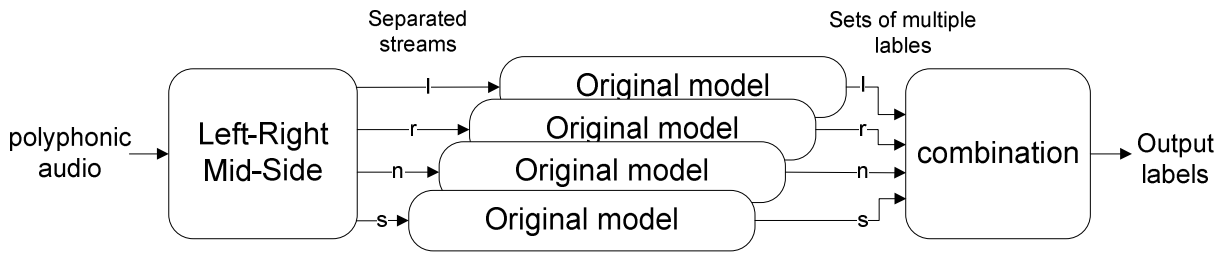


Figure 4.7: Left-Right-Mid-Side separation into lrns streams, used as input of the original instrument recognition models (with no training on this specific separation method)

As it can be observed in Table 4.5, the obtained results show that there is not a considerable difference in the performance with this simple separation, with a maximum Micro F1 = 0.451, compared to the case of using the time consuming FASST source separation with the original labels (n), achieving a maximum F1 = 0.480.

Table 4.5: L-R+M-S separation results, which are only slightly worse than with more complex and time consuming separation algorithms.

	MacPrec	MacRec	MicPrec	MicRec	MacF1	MicF1
n	0.578	0.249	0.708	0.258	**0.349**	**0.378**
s	0.538	0.193	0.586	0.214	0.284	0.313
ns	0.501	0.306	0.595	0.334	0.379	0.427
l	0.578	0.249	0.708	0.258	0.349	0.378
r	0.590	0.249	0.720	0.258	0.350	0.380
lr	0.544	0.301	0.672	0.314	0.388	0.428
nslr	0.485	0.338	0.582	0.367	**0.398**	**0.451**

4.2.5 Experiment 5: Optimizing the performance of the FASST separation + models trained with separated audio

This experiment aimed at improving the results obtained in Experiment 3, with: FASST separation + models trained with separated audio. Since different models are used for each of the 4 streams of separated audio, it is possible to perform an optimization of the parameters for each of them. Furthermore, the initial strategy for the combination of labels in the previous experiments was very simple: the output labels were the union of the predicted labels by all models. In this experiment, different combinations are explored based on the requirement of a degree of overlap N between the outputs of the models. This means that the output labels correspond to the ones present in more than N of the sets of labels predicted by the models.

After running the experiments, it was found (as expected) that if the value of N was increased, the precision increased as well, at the expense of a lower recall. With N = 0, which means that no overlap is required, the obtained micro F1 is equal to 0.446. If N = 1, which is

equivalent to outputting only the labels which had been predicted by at least two of the classifiers, the micro precision is increased at the maximum level from all experiments: 0.733, but the recall is considerably reduced, and thus the F1 decreases to 0.354. The effect that increasing the degree of overlap has in the performance of the integration can be observed in Figure 4.8.

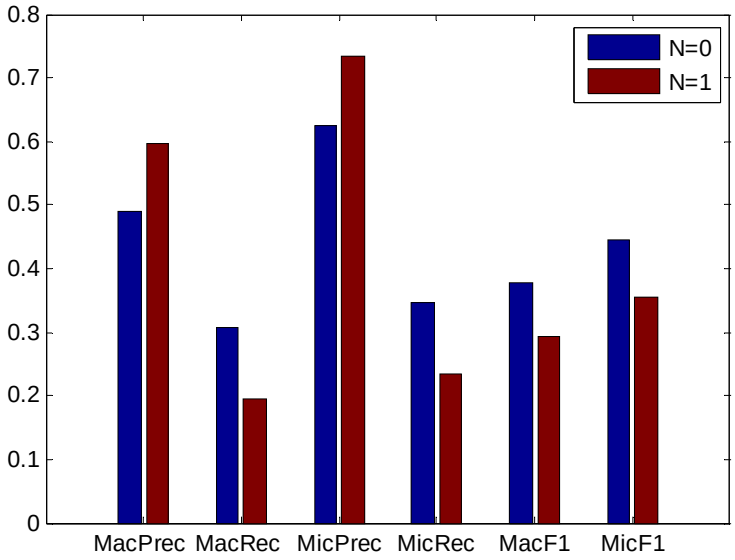


Figure 4.8: Effect of increasing the minimum degree of overlap N in the labels outputted by the classifiers, with dbmo streams. The precision is increased, but the recall and F1-measure decreased.

The use of such strategy would only be useful if the requirement was to have a better precision. However, the F1-measure decreased, and thus the overall performance could be considered worse.

On the other hand, the use of a different configuration for the training of each of the four models led to some improvements in the results. The configuration which allowed improving the performance the most was found to be the use of a lower degree in the polynomial kernel of the SVM classifiers. More specifically, the following combination was found to provide the best results: a second degree polynomial kernel for the bass, melody and other models, and a third degree polynomial kernel for the drums model, in combination with a cost parameter of C=0.1 in all of them.

Figure 4.9 shows an overview of the results of the experiments, in which the minimum degree of overlap between labels was set to N=0, which provided the best results in terms of the F1-measure. The output labels were thus the union of all labels predicted by each of the models.

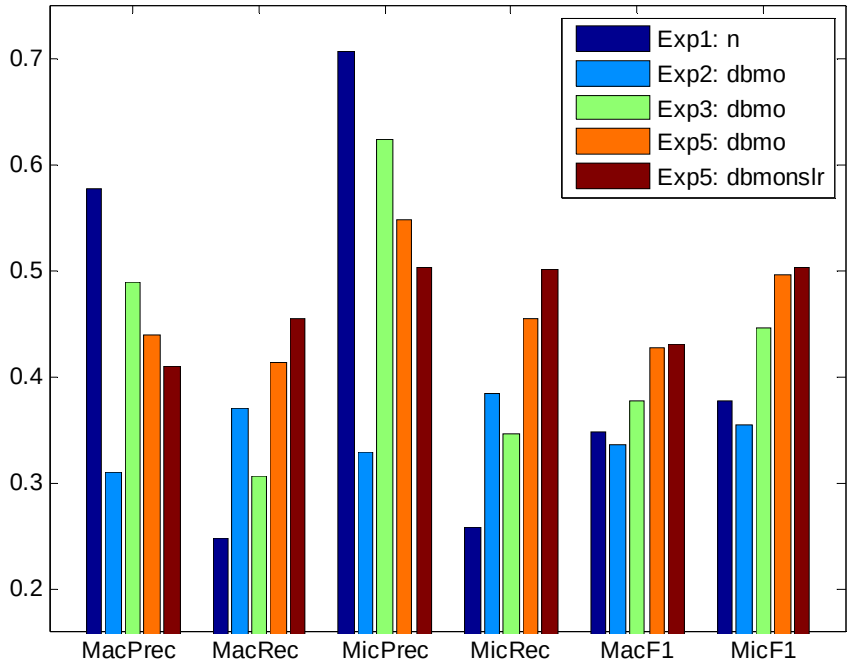


Figure 4.9: Comparison of the instrument recognition performance obtained with several configurations of the experiments. "Exp1: n" corresponds to the results obtained with the n label, with no separation as a pre-step. "Exp2: dbmo" corresponds to the results obtained, with the FASST drum +bass +melody+other separation, without a training of the models on the separated audio. "Exp3: dbmo" corresponds to the same combination of labels, but using models trained with separated audio. "Exp5: dbmo" corresponds to the experiment which used different model parameters for each of the four streams. "Exp5: dbmonslr" corresponds to the combination of the labels from "Exp5: dbmo" and the labels which were obtained in Experiment 4, with the LRMS separation.

The first column of each of the evaluation measures in Figure 4.9 corresponds to the results from Experiment 1. As a reminder, these are the results obtained with the instrument recognition algorithm by itself, which are to be improved by the combination with the source separation. It can be observed that "Exp1: n" represent the most precise results, at the expense of having a low recall, which provides a medium F1-measure. Experiment 2 makes use of the FASST dbmo separation as a pre step to the instrument recognition, but the precision drops, and the results can be considered as worse, since the F1-measure is lower. The worse results were considered to be due to the errors and artifacts produced by the separation, and therefore, Experiment 3 was

designed to acknowledge these problems, by using models of the instruments trained on source separated data. As it can be observed in Figure 4.9, the results obtained in “Exp 3: dbmo” are considerably better than with “Exp 2: dbmo”, and also “Exp1: n”, in terms of F1-measure. The results from “Exp5: dbmo” show that it is possible to further improve the instrument recognition by tuning the parameters of each of the dbmo models. Finally “Exp5: dbmonslr” corresponds to the best results obtained in any of the automatic instrument recognition experiments, by combining “dbmo” labels with the tuned models and the “nslr” labels obtained with the Left-Right-Mid-Side separation from Experiment 4. The detailed results for all possible combination of labels can be found in Annex B, as well as a figure with the evaluation measures for each of the instruments to be recognized. The best micro F1-measure obtained goes above 50%, thanks to the recall gained by the combination of all labels. The initial micro F1-measure, obtained with no separation was 37.8%, so we were able to improve a 12.2% in absolute terms, which represents a 32.3% relative to the initial value.

5 Conclusions and Future Work

The present work has focused on the study of the relation between instrument recognition and source separation. The main motivation was to find synergies between both kinds of algorithms, in order to improve their quality, since current approaches still present many limitations in the context of professionally produced music recordings. Overcoming the limitations of the algorithms would be of much importance for both the research community and the industry as there are many areas of application. The positive results obtained in this work show that the followed methodology is promising, with many possibilities for further research, as well as potential applications.

5.1 Contributions

The following list contains the main contributions of this work:

- The analysis of the limitations of the state-of-the-art instrument recognition and source separation algorithms, and the proposal and implementation of effective methods for their integration.
- The improvement of the results of all the considered state-of-the-art source separation algorithms, by using a prior instrument recognition step.
- The innovative use of source separated data to train the classifiers used for automatic instrument recognition, which allowed to considerably improving the quality of the results.
- The proposal and use of a simple separation method such as the Left-Right-Mid-Side (LRMS), which is a fast but effective alternative to slower separation algorithms, when used for the integration with an instrument recognition algorithm.
- The relative improvement of the performance of the automatic instrument recognition by around 32% (in terms of micro F1-measure).

5.2 Conclusions

In relation to the contributions, some conclusions can also be derived. The most important conclusion is that we have been able to find synergies between instrument recognition and source separation, validating the main hypothesis of this work.

Firstly, it is possible to improve the quality of several state-of-the-art source separation algorithms by the combination with an instrument recognition algorithm, as presented in section 4.1. The amount of quality gained depends on the specific manner in which the combination is executed, but it also depends on the database used for the evaluation. As previously introduced, the proposed method is mostly effective in songs where there are sections in which the target instrument (in this case the voice) is not present.

Secondly, the recognition of instruments has also been improved by around 32% of the original performance in terms of the F1-measure, with a previous separation step, as described in section 4.2. However, the way in which the combination is made is very important to be able to improve the results of the algorithms: in section 4.2.2 it was found that the application of a source separation pre-step may not provide a better recognition of the instruments if the models do not consider the limitations and errors of the separation algorithms. Training the SVM models used for classification with the separated audio has been found to be an effective manner of acknowledging the typical source separation errors, leading to a better performance, which can be further enhanced by tuning the parameters of each of the different models used in the instrument recognition.

The main drawback of the use of source separation is that it is typically slow, except for the online approaches, which generally perform worse. In case that the execution time is an important issue, it was concluded that it is also possible to substantially increase the quality of the instrument recognition with a simple and fast LRMS separation.

5.3 Future work

Some positive results have been obtained, and synergies have been found, in a fairly similar degree in both directions. However, there is still room for improvement, and thus much work could still be done to obtain more accurate results. Additionally, further applications are devised.

5.3.1 Improving source separation with instrument recognition

One of the drawbacks of the algorithm used for instrument recognition is that the best recognition is obtained with audio excerpts of around 3 seconds. In the proposed configuration, the best results would be obtained if we had information about the presence or absence of the target instruments with the best temporal resolution as possible. Further work could deal with the investigation of other approaches to instrument recognition that rely on smaller segments of audio.

Anyhow, the presented combination method is only useful for avoiding the separation to take place in the segments of the input audio excerpt in which the target instrument is not present. However, if the target instrument is present, the combination is limited by the quality of the separation algorithm used. Possible further research would be the improvement of the instrument recognition methods which are internally used in the separation algorithms for the estimation of the fundamental frequency trajectory. An additional possibility would be the combination of musicological knowledge to restrict the possible candidate pitches of the instruments, by exploiting the musical context.

Other extensions of this thesis would be the consideration of other instruments apart from the voice, the use of MIDI information to assist the fundamental frequency estimation in combination with the instrument recognition, or the evaluation of the quality of the separation with the use of perceptual measures, such as the ones obtained with PEASS.

5.3.2 Improving instrument recognition with source separation

A possible extension of this thesis would be to further investigate how to improve the instrument recognition, with a more complex consideration of the probabilities of presence in the combination of the predicted labels in each of the streams after the separation. Also, a deeper analysis of the characteristics of each of the separated streams and the instrument recognition performance, could allow a tailored implementation of the models: instead of considering 11 instruments to be recognized by each of the models, it could be possible to focus each model in some of the instruments only, according to the characteristics of the separated stream. Some other possibilities for further work could be the increase of the amount of instruments to be recognized.

5.3.3 Improving other MIR tasks

The positive results obtained in this work and other contributions which employed harmonic+percussive separation to improve chord detection, melody extraction or genre classification, suggest that musical audio separation helps in the semantic analysis of musical data. Many additional tasks within MIR would thus also benefit of the combination with instrument recognition and source separation, thus ensuring many possibilities for further research. Some of these tasks could be the cover song identification or the tempo estimation.

5.3.4 Applications

Some potential applications which could be endeavoured in relation to this thesis are:

- **Musical hearing aids based on timbral information:** Listeners with hearing loss benefit of processes such as gain, compression and equalisation when dealing with musical audio. Combined with research about the perception of timbre by hearing-impaired listeners, it could be possible to develop a hearing aid system which analyses the musical audio content and adapts it to the appropriate input for the user, by considering the instruments present in the piece. It could also be possible for the user to adjust the parameters of the hearing aid in order to have the most pleasant musical experience as possible.
- **Graphical User Interface (GUI) assisted source separation and remixing based on Pan-Frequency (PF) filtering:** This application would allow a user to select specific regions of the PF space with a GUI and have immediate auditory feedback of the audio content within the selected regions, thanks to the low computational cost of PF filtering. The application would then identify and display the most present instruments in these regions. The user could then chose the ones he/she is interested to separate, or could also adjust their volumes, panning and equalisation to create a new mix.

References

[1] V. Alluri and P. Toiviainen, “Exploring perceptual and acoustical correlates of polyphonic timbre”, *Music Perception*, vol. 27, no. 3, pp. 223–242, 2010.

[2] O. Celma and X. Serra, “FOAFing the music: Bridging the semantic gap in music recommendation”, *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, no. 4, pp. 250-256, Nov. 2008.

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

[4] F. Fuhrmann and P. Herrera, “Polyphonic Instrument Recognition for exploring semantic Similarities in Music”, in *Proc of the 13th Int Conference on Digital Audio Effects DAFx10 Graz Austria*, 2010, pp. 1-8.

[5] E. Vincent, “Musical source separation using time-frequency source priors”, *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 1, pp. 91-98, Jan. 2006.

[6] H. Tachibana, T. Ono, N. Ono, and S. Sagayama, “Melody line estimation in homophonic music audio signals based on temporal-variability of melodic source”, in *ICASSP*, 2010, pp. 425-428.

[7] K. D. Martin, “Sound-Source Recognition: A Theory and Computational Model”, MIT, 1999.

[8] P. Herrera-Boyer, G. Peeters, and S. Dubnov, “Automatic Classification of Musical Instrument Sounds”, *Journal of New Music Research*, vol. 32, no. 1, pp. 3-21, Mar. 2003.

[9] J. M. Grey, “Multidimensional perceptual scaling of musical timbres”, *Journal of the Acoustical Society of America*, vol. 61, no. 5, pp. 1270-1277, 1977.

[10] P. Iverson and C. L. Krumhansl, “Isolating the dynamic attributes of musical timbre”, *Journal of the Acoustical Society of America*, vol. 94, no. 5, pp. 2595-2603, 1993.

[11] S. McAdams, S. Winsberg, S. Donnadieu, G. Soete, and J. Krimphoff, “Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes”, *Psychological Research*, vol. 58, no. 3, pp. 177-192, Dec. 1995.

[12] G. Peeters, S. McAdams, and P. Herrera, “Instrument Sound Description in the Context of MPEG-7”, in *Proc. International Computer Music Conference*, 2000.

[13] J. J. Burred, “From sparse models to timbre learning: new methods for musical source separation”, Technical University of Berlin, 2008.

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

[15] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals”, *Ieee Transactions On Speech And Audio Processing*, vol. 10, no. 5, pp. 293-302, 2002.

[16] B. Logan and A. Salomon, “A music similarity function based on signal analysis”, *Evaluation*, vol. 0, no. C, pp. 2-5, 2001.

[17] S. Essid, G. Richard, and B. David, “Instrument recognition in polyphonic music based on automatic taxonomies”, *IEEE Transactions On Audio Speech And Language Processing*, vol. 14, no. 1, pp. 68-80, 2006.

[18] T. Heittola, A. Klapuri, and T. Virtanen, “Musical instrument recognition in polyphonic audio using source-filter model for sound separation”, in *Proc. 10th Int. Soc. Music Inf. Retrieval Conf*, 2009.

[19] R. Marxer, J. Janer, and J. Bonada, “Low-latency Instrument Separation in Polyphonic Audio Mixtures Using Timbre Models”, *EURASIP Journal on Signal Processing (submitted)*, 2011.

[20] H. Akima, “A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures”, *Journal of the ACM*, vol. 17, no. 4, pp. 589-602, Oct. 1970.

[21] H. Terasawa, M. Slaney, and J. Berger, “The thirteen colors of timbre”, in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005, pp. 323-326.

[22] F. Fuhrmann, M. Haro, and P. Herrera, “Scalability, generality and temporal aspects in automatic recognition of predominant musical instruments in polyphonic music”, in *Proc. of ISMIR*, 2009.

[23] G. C. Bowker and S. L. Star, *Sorting Things Out: Classification and Its Consequences (Inside Technology)*. The MIT Press, 1999, p. 389.

[24] T. Hastie and R. Tibshirani, “Classification by pairwise coupling”, *Annals of Statistics*, vol. 26, no. 2, pp. 451-471, Jul. 1998.

[25] C.-chung Chang and C.-J. Lin, “LIBSVM: a Library for Support Vector Machines”, *ACM Trans. Intell. Syst. Technol*, vol. 2, no. 3, 2011.

[26] P. O. Hoyer and P. Dayan, “Non-negative matrix factorization with sparseness constraints”, *Journal of Machine Learning Research*, vol. 5, pp. 1457-1469, 2004.

[27] P. Paatero and U. Tapper, “Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values”, *Environmetrics*, vol. 5, no. 2, pp. 111-126, 1994.

[28] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization”, *Nature*, vol. 401, no. 6755, pp. 788-791, 1999.

[29] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, “RWC Music Database: Music genre database and musical instrument sound database”, in *Proc. ISMIR*, 2003, pp. 229-230.

[30] G. Tzanetakis, “Song-specific bootstrapping of singing voice structure”, in *Proc IEEE International Conference on Multimedia and Expo ICME*, 2004.

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

[32] A. S. Bregman, *Auditory Scene Analysis*. Cambridge, Massachusetts: MIT Press, 1990, pp. 168-175.

[33] D. Wang and G. J. Brown, *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Wiley-IEEE Press, 2006, pp. 209-250.

[34] E. Vincent, M. Jafari, S. Abdallah, and M. Plumbley, “Model-based audio source separation”, *Queen Mary Univ. of. London, London, U.K., Tech. Rep. C4DM-TR-05-01*, 2006.

[35] G. Siamantas, M. Every, and E. Szymanski, “Separating sources from single-channel musical material: A review and future directions”, in *Proc. of Digital Music Research Network Conference*, 2006, pp. 2-5.

[36] E. Vincent et al., “A Tentative Typology Of Audio Source Separation Tasks”, in *Proc. International Symposium on Independent Component Analysis and Blind Signal Separation*, 2003, pp. 715-720.

[37] O. Yilmaz and S. Rickard, “Blind Separation of Speech Mixtures via Time-Frequency Masking”, *IEEE Transactions on Signal Processing*, vol. 52, no. 7, pp. 1830-1847, Jul. 2004.

[38] M. Vinyes, J. Bonada, and A. Loscos, “Demixing Commercial Music Productions via Human-Assisted Time-Frequency Masking”, in *Proc. of Audio Engineering Society 120th Convention*, 2006, pp. 191 - 199.

[39] K. Dressler, “Extraction of the melody pitch contour from polyphonic audio”, in *Proc. 6th Int. Conf. Music Information Retrieval*, 2005.

[40] A. Klapuri, “Multiple fundamental frequency estimation by summing harmonic amplitudes”, in *Proc. ISMIR*, 2006, pp. 216-221.

[41] J. L. Durrieu, A. Ozerov, C. Fevotte, G. Richard, and B. David, “Main instrument separation from stereophonic audio signals using a source/filter model”, in *Proc. European Signal Processing Conference*, 2009.

[42] A. Ozerov, E. Vincent, and F. Bimbot, “A General Flexible Framework for the Handling of Prior Information in Audio Source Separation”, *INRIA, Tech. Rep. 7453*, 2010.

[43] T. Virtanen, “Monaural Sound Source Separation by Nonnegative Matrix Factorization With Temporal Continuity and Sparseness Criteria”, *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, pp. 1066-1074, Mar. 2007.

[44] T. Virtanen and A. Klapuri, “Analysis of polyphonic audio using source-filter model and non-negative matrix factorization”, in *Advances in Models for Acoustic Processing, Neural Information Processing Systems Workshop*, 2006.

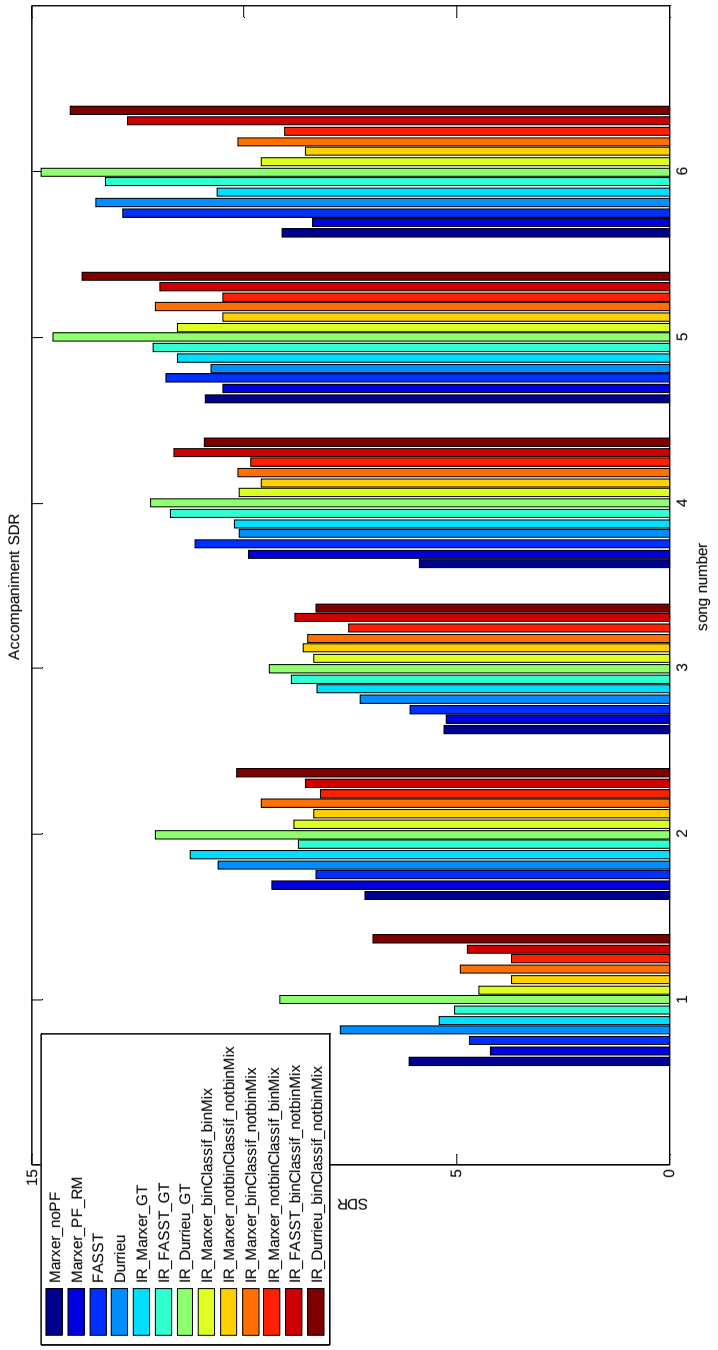
[45] E. Vincent, H. Sawada, P. Bofill, S. Makino, and J. P. Rosca, “First stereo audio source separation evaluation campaign: data, algorithms and results”, in *Proc. of the 7th international conference on Independent component analysis and signal separation*, 2007, pp. 552-559.

[46] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, “Subjective and objective quality assessment of audio source separation”, *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 7, pp. 2046 - 2057, 2011.

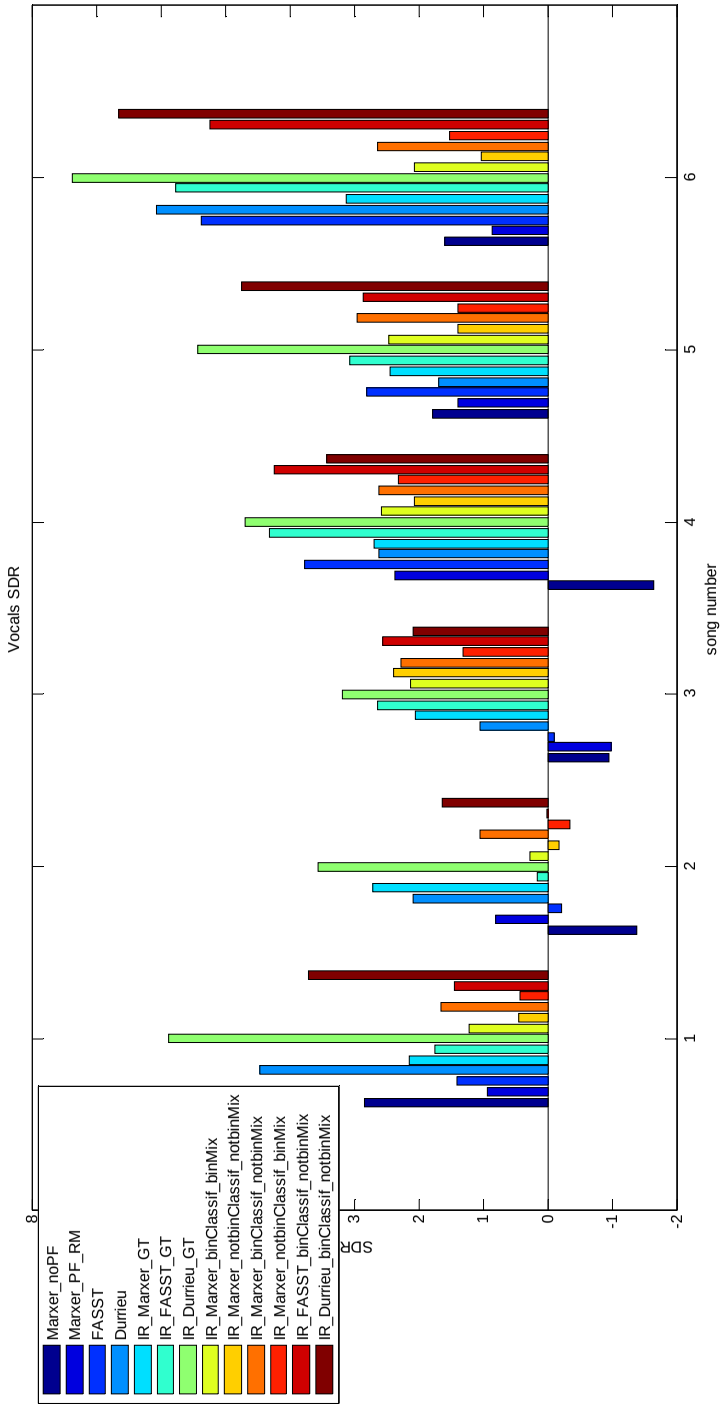
[47] R. Huber and B. Kollmeier, “PEMO-Q - A New Method for Objective Audio Quality Assessment Using a Model of Auditory Perception”, *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 6, pp. 1902-1911, 2006.

Annex A

Source to Distortion Ratio (in dB), for each song and algorithm (target: accompaniment)

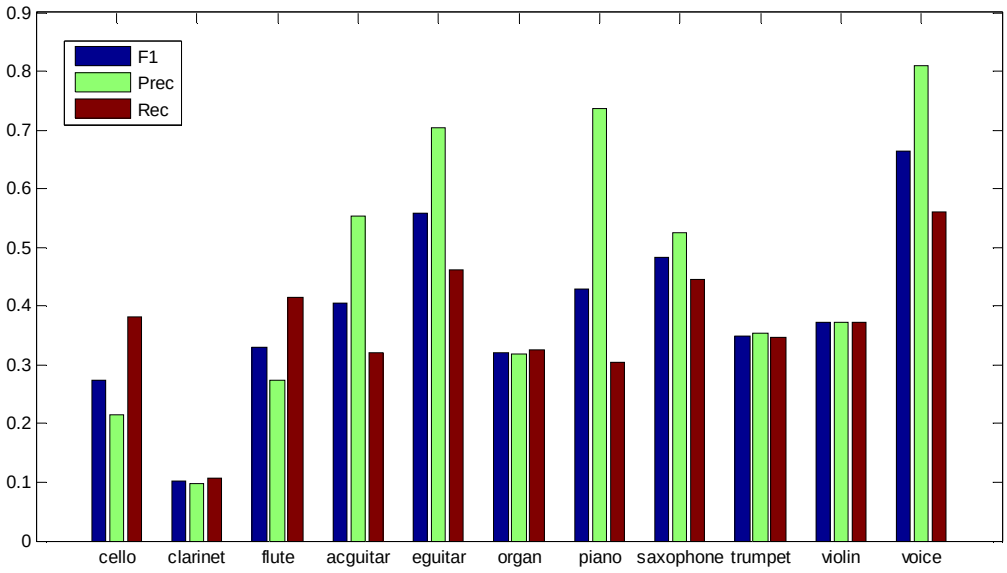


Source to Distortion Ratio (in dB), for each song and algorithm (target: vocals)



Annex B

Best instrument recognition results, obtained in Experiment 5 with the combination of: drum+bass+melody+other tuned models (dbmo labels), and Left-Right-Mid-Side (lrns labels). The evaluation measures are provided for each of the considered instruments. The voice is the easiest to be recognized, while the clarinet is the most difficult.



Macro and micro averages of the evaluation measures for each of the possible label combinations in Experiment 5. The best result is obtained with the combination of all labels ("dbmolrns"), with a micro F-measure of 0.503, compared to 0.378 without source separation ("n")

	MacPrec	MacRec	MicPrec	MicRec	MacF1	MicF1
dbmolrns	**0.410**	**0.455**	**0.504**	**0.501**	**0,432**	**0,503**
dbmon	0.440	0.415	0.549	0.455	0.427	0.497
dmon	0.454	0.399	0.566	0.435	0.425	0.492
bmon	0.459	0.391	0.569	0.429	0.422	0.489
dbmn	0.459	0.393	0.571	0.428	0.423	0.489
mon	0.482	0.373	0.596	0.405	0.420	0.482
nslro	0.458	0.383	0.555	0.422	0.417	0.479
dmn	0.477	0.372	0.595	0.401	0.418	0.479
dbon	0.455	0.369	0.570	0.412	0.408	0.478
dbmo	0.450	0.367	0.563	0.410	0.405	0.475
bmn	0.482	0.363	0.597	0.394	0.414	0.474
don	0.475	0.349	0.595	0.387	0.402	0.469
dmo	0.468	0.348	0.586	0.388	0.399	0.467
dbn	0.485	0.337	0.603	0.372	0.398	0.460
dbm	0.472	0.336	0.589	0.374	0.393	0.458
mn	0.513	0.334	0.637	0.356	0.404	0.457
nslr	0.485	0.338	0.582	0.367	0.398	0.451
bmo	0.471	0.325	0.583	0.363	0.385	0.448
dm	0.495	0.307	0.623	0.341	0.379	0.441
dn	0.515	0.306	0.640	0.333	0.384	0.438
mo	0.503	0.298	0.617	0.330	0.374	0.430
lr	0.544	0.301	0.672	0.314	0.388	0.428
bn	0.518	0.296	0.637	0.322	0.377	0.428
dbo	0.450	0.287	0.581	0.338	0.350	0.428
ns	0.501	0.306	0.595	0.334	0.379	0.427
bm	0.500	0.281	0.616	0.313	0.360	0.415
do	0.472	0.259	0.615	0.305	0.334	0.408
db	0.481	0.230	0.620	0.277	0.312	0.383
r	0.590	0.249	0.720	0.258	0.350	0.380
l	0.578	0.249	0.708	0.258	0.349	0.378
n	**0.578**	**0.249**	**0.708**	**0.258**	**0.349**	**0.378**
m	0.546	0.231	0.673	0.254	0.324	0.368
bo	0.451	0.219	0.591	0.261	0.295	0.362
d	0.517	0.181	0.680	0.218	0.269	0.330
s	0.538	0.193	0.586	0.214	0.284	0.313
o	0.503	0.173	0.642	0.200	0.257	0.305
b	0.470	0.121	0.621	0.151	0.193	0.244