Automatic Musical Instrument Recognition and Related Topics

Arie Livshin

THÈSE DE DOCTORAT

Spécialité
ACOUSTIQUE, TRAITEMENT DU SIGNAL ET INFORMATIQUE
APPLIQUÉS À LA MUSIQUE

IDENTIFICATION AUTOMATIQUE DES INSTRUMENTS DE MUSIQUE

Présentée par Arie A. Livshin
Directeur de Thèse Xavier Rodet

IRCAM Centre Pompidou
Paris 6 - Université Pierre-et-Marie-Curie
Ecolé doctorale EDITE

pour obtenir le grade de
DOCTEUR de l'UNIVERSITÉ PARIS 6 - PIERRE ET MARIE CURIE

soutenue le 12/12/2007

devant le jury composé de

Xavier Rodet	Directeur de Thèse
Régine Andre-Obrecht	Rapporteur
Shlomo Dubnov	Rapporteur
Frédéric Bimbot	Examinateur
Gaël Richard	Examinateur
Geoffroy Peeters	Examinateur
Mark Sandler	Examinateur

Résumé

Cette thèse traite de divers aspects d'Identification Automatique d'Instruments de Musique (IAIM). L'IAIM signifie, d'une manière intuitive, que pour un enregistrement musical donné, l'ordinateur essaie d'identifier quels instruments de musique sont utilisés dans quelles parties de l'enregistrement.

La recherche en IAIM s'est développée au cours des 10 dernières années en particulier grâce à son utilisation en tant que composant d'un moteur de recherche "intelligent" pour la musique. Ce moteur de recherche peut trouver la musique sur internet ou sur des lecteurs MP3 selon des critères "intelligents" comme par exemple le style ou le genre de musique alors que des moteurs de recherche classiques utilisent seulement l'information textuelle liée aux fichiers musicaux. D'autres utilisations de l'IAIM concernent d'autres algorithmes de recherche dans la musique, comme par exemple la transcription automatique et l'alignement de partition, ou encore les logiciels dédiés à la composition musicale ou à l'enregistrement en studio.

L'IAIM est composée de plusieurs étapes qui constituent chacune un défi pour les chercheurs. Les différentes étapes, présentées dans cette thèse, sont les suivantes: obtenir et formater les bases de données de sons pour l'apprentissage et l'évaluation, calculer les descripteurs des sons, procéder au nettoyage automatique des bases de données, attribuer des poids aux descripteurs et réduire leur dimension, et, enfin, classer les sons selon leur appartenance aux différents instruments. Mener une évaluation correcte du déroulement de l'AMIR constitue aussi un travail fondamental.

Ce travail traite en détail des différentes étapes du processus de l'IAIM et, tout en comblant des lacunes et des défaillances dans l'état de l'art, introduit de nouvelles techniques et de nouvelles méthodes pour le perfectionner: il permet d'identifier les instruments de musique à partir des tons séparés, des solos, de la musique polyphonique et multi-instrumentale.

Mots-clefs : indexation automatique multimedia, extraction automatique du contenu, identification d'instrument de musique, méthodes d'évaluation,

PhD Thesis

Automatic Musical Instrument Recognition and Related Topics

Written by Arie A. Livshin
Supervisor Xavier Rodet

IRCAM Centre Pompidou
1, place Igor Stravinsky
75004 Paris, France
http://www.ircam.fr

Paris 6 - Université Pierre-et-Marie-Curie

2007

Part of the requirements for PhD in Informatique of
Paris 6 - Université Pierre-et-Marie-Curie

Abstract

The thesis deals with various aspects of Automatic Musical Instrument Recognition (AMIR). AMIR means, intuitively speaking, that given a musical recording, the computer attempts to identify which parts of the music are performed by which musical instruments.

AMIR research has gained popularity over the last 10 years especially due to its applicability as a component inside "Intelligent" music search-engines, which can allow searching the Internet or mass-storage devices in personal "MP3" players for music using "intelligent" criteria such as musical style or composition - as opposed to searches involving only textual information provided with the musical files. Other usages of AMIR include integration and improvement of other Musical Information Retrieval tasks such as Automatic Transcription and Score Alignment, and as a tool in applications for composers and recording studios.

AMIR is a compound process involving many challenging stages. The various stages of the AMIR process as presented in this thesis include obtaining and formatting of Learning and Test sound databases, computing feature descriptors on the sounds, automatic purging of the databases, feature weighting and dimension reduction of the feature descriptor space and finally, classification of the sounds as belonging to different instruments. Performing informative evaluation of the AMIR process is also important and non-trivial.

This work deals in detail with the different stages of the AMIR process and while "filling holes" in the theory it introduces new techniques and methods for performing many of the tasks, accomplishing AMIR of separate tones, Solo performances and polyphonic, multi-instrumental music.

Keywords: Musical instrument recognition, Music information retrieval, Pattern recognition, Database purging, Classification, Evaluation methods, Feature selection

Acknowledgment

First of all I would like to thank Professor Xavier Rodet for supervising my doctorate and allowing me much flexibility and freedom in my work.

Thanks to Geoffroy Peeters for helping me getting a good start on my work, using his Feature computation routines and providing help whenever asked.

Thanks to Chunghsin Yeh for using his multiple-*f0* estimation program for my AMIR experiments with polyphonic, multi-instrumental music.

Finally, I would like to dedicate this thesis to my parents for supporting me morally and financially whenever needed.

This work was partly supported by the Chateaubriand scholarship by the French Ministry of Foreign Affairs and by the "ACI Masse de données" project “Music Discover”.

Table of contents

Table of contents

Table of contents

Table of contents

CHAPTER 1 OVERVIEW

This PhD thesis deals with "Automatic Musical Instrument Recognition" (AMIR), which means, intuitively speaking, that given a musical recording, the computer attempts to identify which parts of the music are performed by which musical instruments.
The AMIR process is complex and requires many different stages.

This section describes the AMIR process and brings an overview of each chapter in the thesis.

1.1 THE AMIR PROCESS

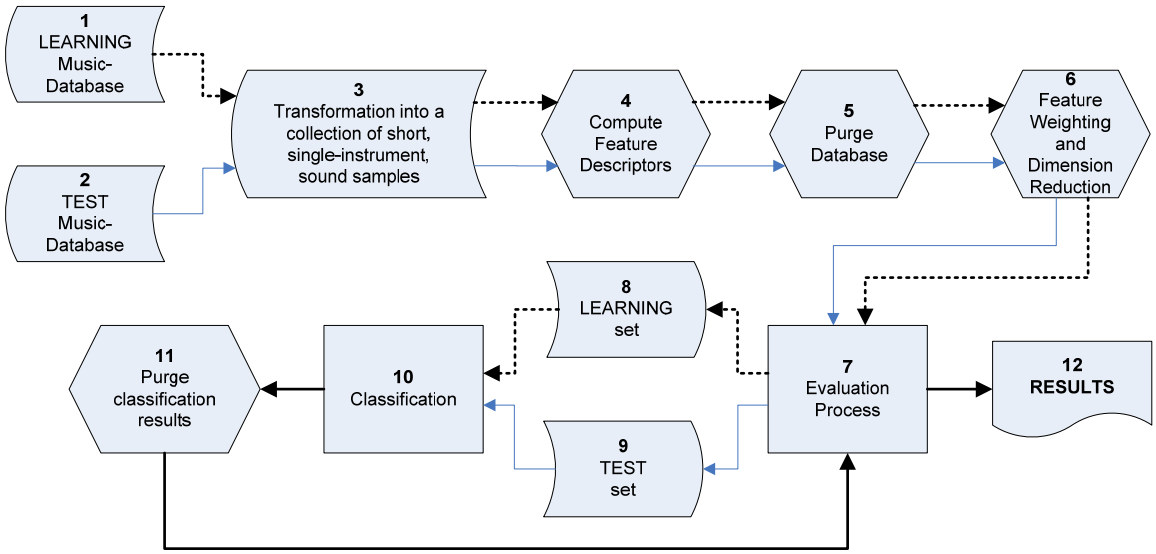

Figure 1-A. A flowchart of the AMIR process

Figure 1-A depicts a flowchart of the AMIR process as performed in this thesis:

- Shapes #1 and #2 – the Learning Music-Database is a collection of instrument-tagged musical recordings which is used for performing AMIR of the untagged Test Music-Database. These music databases could contain separate tones (Chapter 11), Solo performances (Chapter 12) or Multi-Instrumental, Polyphonic (MIP) music (Chapter 14). Chapter 5 details the different sound databases used in this thesis.

  A single musical collection may be used for evaluation by being split into Learning and Test databases, as performed in AMIR of separate notes (Chapter 11) and Solos (Chapter 12), or the Learning and Test databases could be separate, as done in recognition performed on MIP music (Chapter 13). Note that the material contained in the Learning and Test databases could be completely different and they may require different processing, including different Transformation (shape 2), different purging (shape 5), etc. For this reason, two separate arrows are drawn in the flowchart in order to indicate the separate flow of the Learning and Test musical databases throughout the AMIR process.

- Shape #3 – whatever the original musical-databases may be, the AMIR methods in this thesis require their conversion into short sound samples, each containing only sounds of a single musical instrument. In the case of AMIR in separate tones (Chapter 11) this is trivial as the databases come already in the desired form. When performing AMIR on Solos (Chapter 12), the Solos are cut into small, overlapping pieces, monophonic or polyphonic depending on the instruments. When the classified music

is MIP, Chapter 13 proposes several methods in order to obtain separate sound samples of the musical notes.

- Shape #4 – after the short sound samples are ready, various statistics are computed upon each sample in order to capture the different characteristics of its sound. These statistics are called Features. See Chapter 6 for a list of the feature descriptors used in this work.

- Shape #5 – in cases where the Learning sound database may contain samples which are badly recorded or mislabeled, it may be beneficial to 'clean' it in order to prevent classification errors. It may also be possible to remove or label freak samples from the Test set which are likely to be classified incorrectly, depending upon the requirements of the application. Such cleansing process is called 'database purging' and is discussed in Chapter 10.

- Shape #6 – in order to improve class separation and reduce data dimensionality, the feature descriptors are weighted and the feature matrices are transformed into a lower-dimensional feature space. See Chapter 7.

- Shape #7 – performing evaluation of an AMIR process requires careful consideration. While the ideal evaluation method should check whether an AMIR process recognizes "concept instruments" (see Section 2.2 for further discussion), that is, regardless of recording conditions, some evaluation methods may inadvertently only check whether it could learn to recognize very limited types of samples. Chapter 9 presents several evaluation methods and discusses evaluation validity.

- Shapes #8 and #9 – an evaluation method may perform several classification rounds, each time using different data collections from the Learning and Test sets for classification. The sets in shapes #8 and #9 are subsets of the initial music databases in shapes #1 and #2 respectively.

- Shape #10 – the feature vectors are classified by the classification algorithm into classes according to the desired musical instrument taxonomy. For the classification algorithms used in this thesis, see Chapter 8. While the sounds in this thesis are mostly classified as being played by specific musical instruments, the classification classes could be different. See Chapter 4 for several alternative classification taxonomies.

- Shape #11 – semi-purging of the classification results could be done after the sounds are classified by the classification algorithm by calculating confidence levels, marking some of the classifications as likely to be erroneous and deleting or labeling them. See short discussion inside Section 15.5 about some methods for calculating confidence levels. One of the possible methods for detecting and removing suspicious

classifications is to utilize the same purging algorithms used in Chapter 10 (shape #5) as shortly mentioned in Section 13.3.3.

- Shape #12 – after the evaluation process is finished the AMIR evaluation results are reported.

1.2 THESIS CHAPTERS

As described in Section 1.1, this thesis deals with the different stages of the AMIR process.

Chapter 2 - Introduction
This chapter is an introduction to AMIR. After providing a formal definition of the task, Section 2.1 explains why AMIR is an important research area and for which practical applications it is useful; AMIR is mostly applicable not as an end product, i.e., providing instrument recognition results to the user, but rather as a software module integrated into various applications and algorithms such as intelligent music searches over the Internet, automatic music transcription, software for composers and others. Section 2.2 mentions some of the different challenges a researcher in AMIR has to deal with, including classification accuracy and generality, erroneous data, overlapping sounds in polyphonic music, pattern-recognition issues, etc.

Chapter 3 - History
This chapter tells the history of Instrument Recognition research which mainly evolved in three stages - recognition of isolated note samples, recognition of Solos, or as frequently called, “musical phrases”, and finally, Multi-Instrumental Polyphonic (MIP) music.

Chapter 4 - Taxonomies
This chapter mentions some alternative taxonomies and classification hierarchies to the flat instrument taxonomy used in this thesis.

Chapter 5 - Data Sets
This chapter describes the different sound sets and databases used throughout this document: separate tone databases, authentic Solo instrument recordings, authentic Duos and MIP music created by mixing several authentic Solos together.

Chapter 6 - Feature Descriptors
A full list of the feature descriptors used for classification throughout this document is presented along with appropriate references.

Chapter 7 - Feature Weighting and Selection
This chapter discusses the feature selection and weighting techniques used in the thesis. First, Linear Discriminant Analysis (LDA) is described. Afterwards, the new “GDE”

feature selection algorithm is presented and demonstrated. CFS, another feature selection algorithm used in Chapter 11, is described next.

Chapter 8 - Classification Algorithms
Two classification algorithms are used in this thesis – "LDA+KNN", which is my chosen method and is used in most of the thesis chapters, and Backpropagation Neural-Network, used as an alternative classification algorithm in Chapter 9.

Chapter 9 - Different Evaluation Techniques and The Importance of Cross-Database Evaluation
This chapter deals with the issue of evaluation of AMIR techniques; it presents different evaluation techniques including several new cross-evaluation methods: "Minus-1-DB", "Mutual Classification", "Minus-1-Instance" and "Minus-1-Solo". The chapter proves that the new cross-evaluation techniques are preferred over the "Self-Classification" evaluation method, which was very common until quite recently.

Chapter 10 - Improving the Consistency of Sound Databases
The issue of self-consistency of Learning databases for AMIR is addressed. A sound database may contain samples which are badly recorded or mislabeled thus increasing classification errors. The common Interquantile Range technique is compared with two new database purging algorithms – "MIQR" and "SCO". Their performance is evaluated using a sound database contaminated with four different outlier types.

Chapter 11 - AMIR of Separate Tones and the Significance of Non-Harmonic "Noise" Vs. The Harmonic Series
This chapter begins by performing AMIR of separate tones of 10 musical instruments. The chapter then explores the instrument discrimination power of the harmonic series and the residuals as compared to the original, complete sounds. While it is common to treat the Harmonic Series as the main characteristic of the timbre of pitched musical instruments it seems that no direct experiments were performed so far to prove this assumption. In order to check it, using Additive Analysis/Synthesis, each sound sample is resynthesized using solely its Harmonic Series. These "Harmonic" samples are then subtracted from the original samples to retrieve the non-harmonic Residuals. AMIR is performed on the original samples, the Resynthesized ones and the Residuals and the results are compared and discussed. Using the CFS algorithm for feature selection, the best 10 feature descriptors for instrument recognition are found and presented for the Original, Harmonic and Residual sound sets.

Chapter 12 - AMIR in Solos
This chapter deals with Instrument recognition in authentic Solo recordings, which are monophonic or polyphonic musical phrases performed by a single instrument. AMIR in Solos is different and more complicated than dealing with separate note databases as performed in the previous chapter, as the time evolution of each sound (attack, decay, sustain, release) is not well defined, the notes are not separated, there are superpositions of concurrent sounds and room echo, different combinations of playing techniques, etc.

First the Solos are classified "offline" using the complete set of feature descriptors, and then a specially reduced set of descriptors is presented for performing realtime AMIR of Solos.

Chapter 13 - AMIR in Multi-Instrumental, Polyphonic Music
The chapter presents three techniques for performing AMIR on MIP music – Naïve Solo Classifier, Source-Reduction (SR) and Harmonic-Resynthesis (HR). Both SR and HR utilize a multiple-*f0* estimation program in order to extract single notes out of MIP music, SR using filtering and HR using Additive Synthesis, and classify these using Solos in the Learning set. These AMIR techniques are evaluated on authentic recordings of Duos played by seven musical instruments and on mixes of authentic Solo recordings performed by five different instruments, with two to five instruments playing concurrently.

Chapter 14 - Summary
This chapter summarizes the various contributions and results in each thesis chapter.

Chapter 15 - Future Work
AMIR research does not come to an end with this thesis. The chapter introduces various ideas for further research for AMIR improvement.

Appendix A
A table of acronyms and abbreviations used in the thesis.

Appendix B
Six published papers in which I am the main author:
(Livshin, Peeters and Rodet 2003), (Livshin and Rodet 2003), (Livshin and Rodet 2004a), (Livshin and Rodet 2004b), (Livshin and Rodet 2006a) and (Livshin and Rodet 2006b).

I recommend reading four of them - (Livshin, Peeters and Rodet 2003), (Livshin and Rodet 2003), (Livshin and Rodet 2004b) and (Livshin and Rodet 2006b), as (Livshin and Rodet 2004a) and (Livshin and Rodet 2006a) are expanded and updated in later papers.

CHAPTER 2 INTRODUCTION

Intuitively speaking, "Automatic Musical Instrument Recognition" is a process where given a musical recording, the computer attempts to identify which parts of the music are performed by which musical instruments.

More formally, suppose that a given audio signal contains *K* notes (or other types of audio segments), $n_1, n_2, \ldots, n_k, \ldots, n_K$. The instrument identification process consists of two basic subprocesses: feature extraction and a posteriori probability calculation. In the former process, a feature vector consisting of some acoustic features is extracted from the given audio signal for each note. Let $\boldsymbol{x}_K$ be the feature vector extracted for note n_k. In the latter process, for each of the target instruments, $\omega_1, \ldots, \omega_m$, the probability $p(\omega_i|\boldsymbol{x}_k)$ that the feature vector $\boldsymbol{x}_K$ is extracted from a sound of the instrument ω_i is calculated. Based on the Bayes theorem, $p(\omega_i|\boldsymbol{x}_k)$ can be expanded as follows:

$$p(\omega_i|\boldsymbol{x}_k) = \frac{p(\boldsymbol{x}_K|\omega_i)p(\omega_i)}{\sum_{j=1}^{m} p(\boldsymbol{x}_K|\omega_j)p(\omega_j)}$$

where $p(\boldsymbol{x}_k|\omega_i)$ is a probability density function (PDF) and $p(\omega_i)$ is the a priori probability with respect to the instrument ω_i. The PDF $p(\boldsymbol{x}_k|\omega_i)$ is trained using data prepared in advance. Finally, the name of the instrument maximizing $p(\omega_i|\boldsymbol{x}_k)$ is determined for each note n_k.

When classifying musical phrases (Solos) or Multi-Instrumental, Polyphonic (MIP) music, extracting separate notes out of the music is non-trivial. In Solos it is difficult to detect exact note boundaries, i.e., where one note begins and another ends, while in MIP music, in addition, several notes, possibly of different instruments, may be played at the same time and their intermixed waveforms are very difficult to separate.

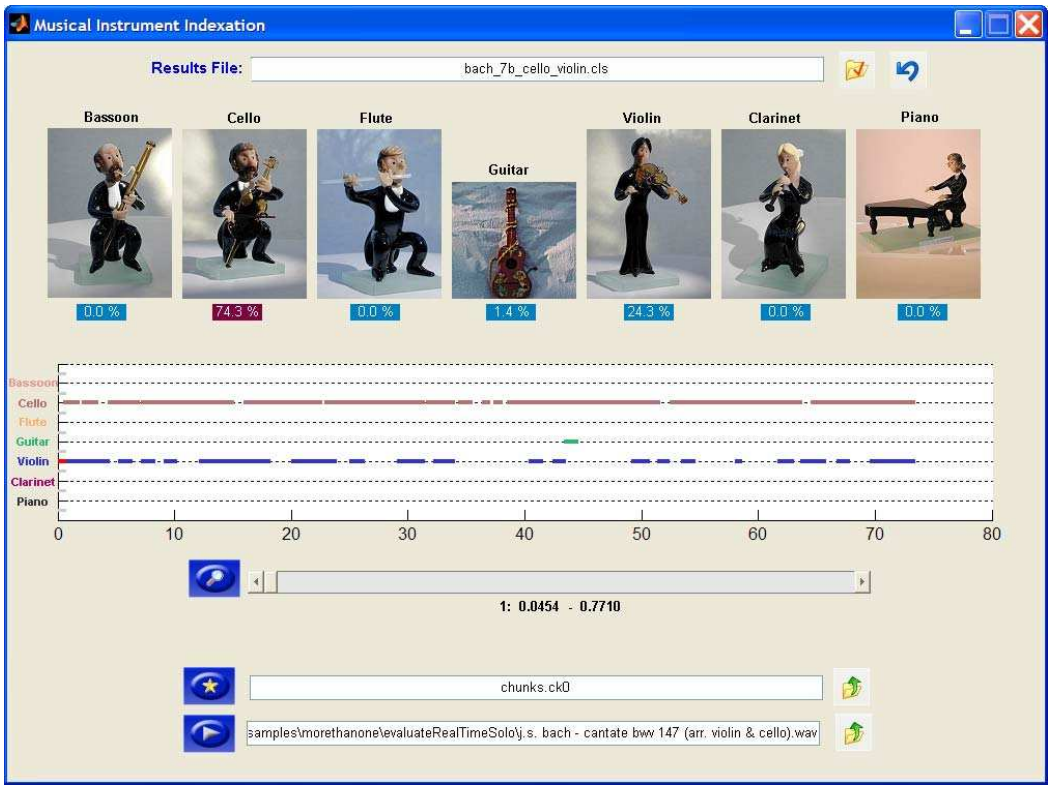

Figure 2-A. An AMIR output example

This screenshot displays the results of running my AMIR application using the Source-Reduction technique (see Section 13.1.2) on the Cantata BW 147 by J.S. Bach Duo performed by violin and cello. The vertical axis indicates the musical instrument while the horizontal axis shows at which points in time it is playing[1].

2.1 MOTIVATION

Automatic musical instrument recognition algorithms can have different practical applications:

2.1.1 INTELLIGENT SEARCH OF MUSIC

By 1996 the Internet entered common daily usage, allowing information to be shared and exchanged conveniently and quickly around the world. Although public Internet was very slow in the beginning, the MPEG-1 Audio Layer 3 (MP3) audio compression algorithm (MPEG 1992) allowed music to be compressed several tenfold while still sounding like a faithful reproduction of the original uncompressed audio to most listeners, thus making it

[1] One can notice a recognition mistake at the 44'th-second, when a "phantom" guitar note was detected.

practical to transfer music even through slow Internet. Peer-to-Peer applications, especially since the release of Napster, allowed and encouraged masses of users around the world to share and exchange music files. The invention of personal audio players capable of playing MP3 compressed files has boosted the rate of music exchange over the Internet even further. Today there are billions of musical pieces scattered over the Internet. Personal digital-audio players have large storage devices which may contain tens of thousands of songs.

To the computer or personal music players, however, the music files are merely streams of bits in some coding scheme which are converted into sounds when played. Today, we have internet search engines that can identify text documents matching a user's query, but multimedia documents are opaque to search engines. Virtually all today's systems have no way of performing "intelligent" searches of music, such as looking for songs by similarity, by genre or by performing instruments. Although efforts have begun to define standardized "descriptors," or *meta-data* formats, for multimedia data (MPEG Requirements Group, 1999), yet there are still no tools that can extract the relevant information automatically. The producer of the data must add the meta-data by hand.

In the future, AMIR could be performed automatically on musical pieces scattered on the Internet and label them. Musical instrument information, in addition to allowing searches to be performed by the playing instruments, can also help greatly in searches by similarity, genre and other "intelligent" criteria.

2.1.2 STRUCTURED-AUDIO ENCODING

Today, with high bandwidth music distribution channels, including high capacity swappable storage such as blu-ray, PDD and even conventional DVD's, high speed Internet and non-swappable high storage capacities of personal music players, musical recordings could be distributed in *structured* formats (like described in Vercoe et al., 1998) that preserve the isolation of individual sounds until the time of playback but require much more storage space than pre-mixed music.

Structured-media formats make automatic multimedia annotation easier. In addition, they give the end user more control over the media playback. For example, an audio enthusiast could take better advantage of a seven-speaker playback setup if the audio material was not pre-mixed for stereo playback. Musicians could "mute" a particular part of a recording and play along.

Although structured formats provide immense advantages over their nonstructured counterparts (such as the current generation of compact discs and DVDs), there is currently no automatic way of adding structure to unstructured recordings.

At the recording stage, the different musical instruments are often recorded and stored using separate channels of a multi-track recording system. Performing realtime Solo

AMIR while recording the instruments separately or using archived multi-track master tapes, as described in Section 12.4 and (Livshin and Rodet 2004b), can allow computing instrument Meta-data automatically and preserving it throughout the production process.

In the future, by combining robust tools from AMIR, music transcription, and speech recognition, it may be possible to build fully or partly automated tools for unstructured-to-structured encoding. See for example (Livshin et al. 2005) for a fully automatic Wave-to-MIDI conversion system.

2.1.3 MUSIC INFORMATION RETRIEVAL (MIR)

Successful Musical Instrument Recognition can benefit other MIR research fields, such as *f0*-estimation and score alignment, by allowing their algorithms to assume spectral, temporal or other qualities of the specific instruments participating in the analyzed musical piece.

2.1.3.1 Automatic Transcription

The process of listening to a piece of music and reconstructing the notated score is known as *transcription*. More generally, transcription is the process of determining *which* musical notes were played *when* (and by *which* instrument) in a musical recording or performance. In the general case of music played by multiple instruments (or a single polyphonic instrument such as the guitar or piano), the task is one of polyphonic pitch tracking. This is difficult—humans require extensive training in order to transcribe music reliably. Nevertheless, as transcription is an important tool for music theorists, music psychologists, and musicologists— not to mention music lovers who want to figure out what their favorite artists are playing in rapid passages—it would be wonderful to have tools that could aid the transcription process, or automate it entirely. Polyphonic pitch tracking research demonstrates that the task may be made simpler if good—and explicit—models of the sound sources (the musical instruments) are available (Kashino and Murase, 1998). By integrating sound source recognition with a transcription engine, the end result can be improved.

In addition to the possibility of improving the transcription engine by providing it with the instrument modules, AMIR is indispensible for instrument segmentation of the notes; in an unpublished report (Livshin et al. 2005), we have integrated an instrument recognition module with a multiple-*f0* estimation module to create a system which automatically creates partituras out of recorded music, by first finding the different notes in a musical piece and then arranging them into staves by their recognized musical instruments.

2.1.4 A TOOL FOR COMPOSERS AND SOUND EDITORS

An AMIR option in sound editing applications can allow quick searches inside the music for parts where a certain instrument is playing.

2.2 CHALLENGES

AMIR poses many challenges and questions:

2.2.1 ACCURACY

How can we distinguish between similar instruments - is it possible, for example, to distinguish among seemingly identical sounds coming from different instruments, such as equally pitched sounds of viola and violin?

2.2.2 GENERALITY

The hypothetical goal of building an ideal classifier which could recognize all the sound variations of a musical instrument is a wholly different task than successfully classifying a specific sound database. What are the distinguishing qualities of a “Concept Instrument” which define, bound and fundamentally separate it from all other instrument classes, regardless of the recording conditions or the specific instrument (e.g., a specific Stradivarius violin) being used?

2.2.3 TAXONOMY

What should be the instrument classes, and which instruments should be classified in the same class when categorizing into instrument families? Should sounds recorded in different settings and playing techniques be classified in the same class? Should recordings of a string ensemble, for example, and a pizzicato sound of a single violin considered the same instrument class? Acoustic and electric guitars?

2.2.4 DATA VALIDITY

Does the instrument recognition algorithm learn enough sound variations of an instrument (“encapsulation”)? Are all the learned sound samples really beneficial for the recognition or maybe some actually sabotage it (“consistency”)? Are there “bad” samples or misclassified sounds (“database errors”)?

2.2.5 POLYPHONICITY

Being able to distinguish Solo recordings of one instrument from another, does not suffice in the multi-instrumental case, where several instruments play concurrently. The sounds are intermixed and even influence each other. How do we handle instrument recognition in MIP music?

2.2.6 PATTERN RECOGNITION ISSUES

Misclassification may occur due to different pattern recognition issues:

2.2.6.1 Classification Process

- Badly defined sound classes, or different classes with virtually identical sounds
- Inappropriate or weak classification algorithms
- Feature descriptors which do not cover enough distinguishing qualities or mislead

2.2.6.2 Sound sets

- Misrepresenting or insufficient Learning set
- Misrepresenting classified sounds

CHAPTER 3 HISTORY

During the last 30 years, different automatic musical instrument recognition (AMIR) systems have been constructed, using different approaches, scopes, and levels of performance. Most of these systems have dealt with AMIR of single, isolated tones (either synthesized or natural). More recent works, from the last 10 years, have employed authentic recordings of musical phrases (Solos), and since 2003, research on AMIR in multi-instrumental, polyphonic (MIP) music began to gain popularity.

3.1 ISOLATED TONES

In (Cosi et al., 1994; De Poli and Prandoni, 1997), a series of Kohonen Self-Organizing-Map (SOM) neural networks were constructed using as inputs feature descriptor vectors, most often MFCC, computed on isolated tones of a specific pitch. One tone per instrument was used with up to 40 instruments in a single experiment. The dimension of the feature vectors was reduced sometimes using Principal Component Analysis (PCA) (Pearson 1901). Unfortunately, the presented recognition rates are unreliable as the Test set was not independent of the Learning set (see Chapter 9 for the importance of independent evaluation).

The instrument classification abilities of a feedforward neural network and a K-Nearest Neighbor classifier (KNN) were compared in (Kaminskyj and Materka 1995). The classifiers were trained on feature descriptors based on the temporal envelopes of isolated tones. The two classifiers achieved recognition rates of about 98% classifying tones of four instruments (guitar, piano, marimba, and accordion), over a one-octave pitch range. Again, like the paper mentioned in the previous paragraph, while the recognition rate seems high, both the Learning and Test data were recorded in the same recording

conditions - same instruments, same players and the same acoustic environment. Adding to that the fact that the four recognized instruments have sounds which are very different from each other implies that it is doubtful whether such high recognition rates will be obtained adding additional instruments or even using independent Learning and Test sets.

Traditional pattern-recognition methods were applied by different authors for performing AMIR in isolated-tones. In (Bourne 1972), perceptually-motivated Feature Descriptors were used as a training set for a Bayesian classifier, including the overall spectrum and the relative onset times of different harmonics, extracted from 60 clarinet, French horn, and trumpet tones. The Test set included 15 tones, with eight tones which did not appear in the Learning set. All tones except one were correctly classified (around 93% recognition rate).

In an unpublished report, Casey (1996) describes a novel recognition framework based on a "distal learning" technique. Using a commercial waveguide synthesizer to produce isolated tones, he trained a neural network to distinguish between two synthesized instruments (brass and single-reed) and to recover their synthesizer control parameters. Although "recognition" results were not quantified as such, the low "outcome error" reported by Casey demonstrates the success of the approach in the limited tests.

In (Fujinaga 1998) a KNN classifier was used with a Learning set consisting of features extracted from 1338 spectral slices representing 23 instruments playing a range of pitches. Using leave-one-out crossvalidation with a genetic algorithm to identify good feature combinations, the system reached a recognition rate of approximately 50%.

(Martin and Kim 1998) exemplified the idea of testing very long lists of features and then selecting only those shown to be most relevant for performing classifications. Martin and Kim worked with log-lag correlograms to better approximate the way our hearing system processes sonic information. They examined 31 features to classify a corpus of 14 orchestral wind and string instruments. They have found the following features to be the most useful: vibrato and tremolo strength and frequency, onset harmonic skew (i.e., the time difference of the harmonics to arise in the attack portion), centroid related measures (e.g., average, variance, ratio along note segments, modulation), onset duration, and select pitch related measures (e.g., value, variance). The authors noted that the features they studied exhibited non-uniform influences, that is, some features were better at classifying some instruments and instrument families and not others. In other words, features could be both relevant and non-relevant depending on the context. The influence of non-relevant features degraded the classification success rates between 7% and 14%.

Brown (1999) used cepstral coefficients from constant-Q transforms (instead of computing them using FFT-transforms); she also clustered feature vectors in a way that the resulting clusters seemed to be coding some temporal dynamics.

Eronen and Klapuri (2000) used non-Mel scaled Cepstral Coefficients, combining these features with a long list (up to 43) of complementary descriptors; their list included,

among other features, the centroid, rise and decay time, frequency/amplitude modulation (FM/AM) rate and width, fundamental frequency and fundamental-variation-related features for onset and for the remainder of the note. In a more recent study, using a large set of features (Eronen, 2001), the features which turned out to be most important were the MFCCs, their standard deviations and their deltas (differences between contiguous frames), the spectral centroid and related features, onset duration, and the crest factor (especially for instrument family discrimination).

In (Kaminskyj 2001) the main author used the RMS envelope, the Constant-Q frequency spectrum, and a set of spectral features derived from Principal Component Analysis (PCA).

Note: Most of the papers on AMIR of isolated tones, including Martin and Kim 1998, Fraser and Fujinaga 1999, Kaminskyj 2001, Agostini et al. 2001, Peeters and Rodet 2002 and others, have used tones randomly selected from a single sound database in both the Learning and Test sets for evaluation of their AMIR techniques. In (Livshin and Rodet 2003) we have shown that results of such evaluation techniques do not necessarily indicate the general ability or performance of an AMIR classification technique. Read further on this issue in Chapter 9.

In (Livshin, Peeters and Rodet 2003) we describe a classification process which produces a high recognition rate with Self-Classification evaluation (see Section 9.4.1). Over 95% recognition rate was achieved classifying 18 instruments and results of three classification algorithms were compared: Multidimensional Gaussian, K-Nearest Neighbors (KNN) and Learning Vector Quantization neural network (LVQ). Lower results were achieved with the Minus-1 (DB) evaluation method. This paper deals with many aspects of instrument recognition, including feature selection, removing outliers, evaluation techniques and more.

In (Livshin and Rodet 2006b) we have taken a similar approach to (Martin and Kim 1998), (Peeters and Rodet 2002) and (Livshin, Peeters and Rodet 2003), and used a large collection of feature descriptors, creating a weighted list of the most relevant features using Linear Discriminant Analysis on the Learning set. A KNN classifier was used to classify the sounds in a flat “all vs. all” classification of the instrument classes, as unlike in (Peeters and Rodet 2003), preliminary tests using hierarchies of instrument families did not show improvements in classification results over the flat model.
A large and diverse collection of sounds was used – tones of 10 different instruments taken from 13 sound databases. The recognition results were high – 94.84% using the Minus-1-Instance evaluation method (see Section 9.7.1). Comparing with lower results using the same classification method, a very similar feature set and the Minus-1-DB evaluation method, which also uses independent Test and Learning sets, in (Livshin and Rodet 2003a and Livshin and Rodet 2003b), we can see that these classification techniques suffice for achieving high recognition rates when using a Learning database which is large and diverse enough. At the time of publishing (Livshin and Rodet 2006b)

this was the state-of-the-art for AMIR recognition in Separate tones. Read Chapter 11 for full details.

For several other historical topics related to AMIR in separate tones, see (Herrera, Peeters and Dubnov 2003).

3.2 SOLO PERFORMANCES

Until 1998, there were practically no published reports of musical instrument recognition systems that could operate on realistic musical recordings.

A vector-quantizer based on MFCC features was used in (Dubnov and Rodet 1998) as a front-end to a statistical clustering algorithm. The system was trained with 18 short excerpts from as many instruments. Although the classification results were not reported, it seems that the vector-quantizer have captured something about the "space" of instrument sounds.

A classifier that distinguishes oboe from saxophone recordings was described in (Brown 1999). For each instrument, a Gaussian mixture model (GMM) was trained on constant-Q cepstral coefficients, using one minute of music from each instrument. The recognition rate of the system was 94% for independent, noisy samples from commercial recordings. This work was extended later on, in (Brown Houix and McAdams 2001), where four wind instruments (flute, sax, oboe and clarinet) were classified using combinations of four feature types, reaching a recognition rate of 82% with the best combination of parameters and training material.

In (Marques and Moreno 1999), eight fairly different instruments (bagpipes, clarinet, flute, harpsichord, organ, piano, trombone and violin) were classified using one CD per instrument for learning and one for classification; they compared three feature types using two different classification algorithms, achieving 70% recognition rate. The best classifiers used MFCC features, correctly classifying approximately 72% of the test data. Performance dropped to approximately 45% when the system was tested with "non-professional" recordings, suggesting that the classifier has not generalized in the same way as humans do. The "non-professional" recordings were a subset of the student recordings.

(Martin 1999) has classified sets of six, seven and eight instruments, reaching 82.3% (with violin, viola, cello, trumpet, clarinet, and flute), 77.9% and 73% recognition rates respectively. Martin has used up to three different recordings from each instrument; in each experiment one recording was classified while the others were learned. The feature set was relatively large for the time and consisted of 31 one-dimensional features.

In our paper, (Livshin and Rodet 2004a), we have presented a process for recognition of Solos of seven instruments (including two highly polyphonic – guitar and piano), using independent test data (unlike some other papers), which yielded a rather high recognition rate (88%) and could also operate in realtime with just a small compromise on the recognition score (85%). At the time of publishing this was the state-of-the-art for AMIR recognition in Solos, both offline and realtime. See Chapter 12 for full details.

3.3 MULTI-INSTRUMENTAL MUSIC

Only a few studies have attempted instrument recognition for polyphonic music; these systems were mostly tested on limited and artificial examples.

A template-based time domain approach was used in (Kashino and Murase 1999). Using three different instruments (flute, violin and piano) and specially arranged ensemble recordings they achieved 68% recognition rate with both the true fundamental frequencies (*f0*s) and the onsets supplied to the algorithm. With the inclusion of higher level musical knowledge, most importantly voice leading rules, recognition accuracy improved to 88%.

A frequency domain approach was proposed in (Kinoshita et al. 1999), using features related to the sharpness of onsets and the spectral distribution of partials. *F0*s were extracted prior to the instrument classification process to determine where partials of more than one *f0* would coincide. Using random two-tone combinations from three different instruments (clarinet, violin, piano), they obtained recognition accuracies between 66% and 75% (73% - 81% if the correct *F0*s were provided), depending on the interval between the two notes.

(Eggink and Brown 2003) have proposed an approach based on missing feature theory to enable instrument recognition in situations where multiple tones may overlap in time. This approach is motivated by a model of auditory perception which postulates a similar process in listeners; since target sounds are often partially masked by an interfering sound, it can be inferred that listeners are able to recognize sound sources from an incomplete acoustic representation (Cooke et al. 2001). Classifiers based on Gaussian mixture models (GMMs) are easily adapted to work with incomplete data (Drygajlo and El-Maliki, 1998). The approach was tested with artificial mixtures of two instrument tones from five different instruments achieving 66% recognition rate, and a single “natural” Duo recording of flute and Clarinet consisting of 12 tones achieving correct recognition of all tones.
Note that Eggink has used only the harmonic series of the sounds for instrument recognition, assuming that these contain sufficient distinguishing information. See Chapter 11 for a comprehensive research of this issue.

(Vincent and Rodet 2004) investigate the use of Independent Subspace Analysis (ISA) for AMIR. They represent short-term log-power spectra of possibly polyphonic music as weighted non-linear combinations of typical note spectra plus background noise. These typical note spectra are learned either on databases containing isolated notes or on Solo recordings of different instruments. The technique is first evaluated on 20 five-second excerpts from 10 Solos of five monophonic musical instruments, and identifies the playing instrument correctly in 90% of these excerpts. When the technique is tested on a single polyphonic excerpt (from a difficult duo), the model is able to identify the right pair of instruments and to provide an approximate transcription of the notes played by each instrument. While the technique was enhanced and improved in a later paper (Vincent 2006), the emphasis drifted away from AMIR to source-separation, which is performed using instruments known beforehand.

(Essid, Richard and David 2006a), instead of attempting to recognize each instrument individually have used classification classes consisting of combinations of instruments played simultaneously. The classification was performed hierarchically where the top levels contained several instrument combinations. The hierarchy was shown to produce higher recognition results than flat, single-level classification. The taxonomy hierarchy was automatically constructed using clustering of different instrument combinations and thus seems to be highly influenced by the specific musical pieces used for testing and their various quantities.
In order to reduce the huge number of possible instrument combinations, i.e., nodes in the hierarchy, they used hypotheses related to genre and orchestration and limited the work to real recordings of Jazz quartets. Feature selection was performed for "one vs. one" classifications from a large collection of features. The system was tested on up to 4 concurrent instruments with seemingly high results, although somewhat difficult to quantify.
This method has the disadvantage that it cannot be used for automatic music transcription applications (see Section 2.1.3) such as MusicXML (Good 2001) as it identifies which group of instruments plays in a time-frame rather than the specific instrument playing each note, which is required for transcribing notes in the appropriate staves according to the performing musical instruments.

(Kitahara et al. 2007) have performed instrument recognition addressing specifically three main issues: feature variations caused by sound mixtures, the pitch dependency of timbres, and the use of musical context. For the first issue, templates of feature vectors representing timbres are extracted from not only isolated sounds but also sound mixtures. Because some features are not robust in the mixtures, features are weighted according to their robustness by using linear discriminant analysis. For the second issue, an *f0*-dependent multivariate normal distribution was used, which approximates the pitch dependency as a function of fundamental frequency. For the third issue, when the instrument of each note is identified, the a priori probability of the note is calculated from the a posteriori probabilities of temporally neighboring notes – this technique is based on the assumption of "well-behaved" music, where an instrument keeps playing above or below another instrument's note-pitches and does not "cross its stream".

This system requires correct *f0* information. It was not evaluated on real musical recordings but was rather tested with "nonrealistic" musical pieces mixed artificially from individual samples - separate tones of 5 musical instruments were mixed according to three MIDI files. Experimental results yielded average recognition rates of 84.1% for Duo, 77.6% for Trio, and 72.3% for Quartet.
Note that Kitahara has used only the harmonics of the sounds for instrument recognition, saying that "although actual musical instrument sounds contain nonharmonic components which can be factors characterizing sounds, (they) focus only on harmonic ones because nonharmonic ones are difficult to reliably extract from a mixture of sounds." In Chapter 11 this issue is tackled and it is shown that the harmonic series by itself indeed contains enough information in order to achieve high recognition rates comparable with results obtained from using the complete signals.

In this thesis two approaches for instrument recognition in polyphonic, multi-instrumental music are presented, both described fully in Chapter 13. These methods use a multiple-*f0* recognition module, currently (Yeh, Röbel and Rodet 2005).
My "Source-Reduction" method (SR) uses a multiple-*f0* estimation program to find the different notes in a musical piece, for each detected note over a minimal length to filter out everything except its partials and then classify this "cleansed" note using Solos in the training set. This method was used in (Livshin and Rodet 2004b) with 18 authentic Duo recordings of seven instruments – Bassoon, Flute, Clarinet, Guitar, Piano, Violin and Cello, achieving an average "Mutual" grade (See Section 13.2 for explanation) of 81.33% with resolution of 0.5second, recognizing correctly both participating instruments in 17 out of the 18 Duos tested. Using a large number of mixed authentic Solos of five instruments – bassoon, flute, clarinet, violin and cello, this method produced an average Instrument grade (See Section 13.3) of 54.1% for 2 – 5 instruments playing concurrently, with 0.5second resolution.

My "Harmonic-Resynthesis" method is based on my findings in (Livshin and Rodet 2006b - see Chapter 11), where it is shown that using only information in the harmonic series of a note is enough for achieving high instrument recognition rates. The Harmonic-Resynthesis method uses a multiple-*f0* estimation program to find the different notes in a musical piece and for each detected note over a minimal length, to resynthesize it using its estimated harmonic series and classify it using a Learning set consisting from notes resynthesized from harmonic series of notes detected in Solos. Harmonic-Resynthesis achieved a "Mutual" score of 83.2% with authentic Duo recordings of seven instruments – bassoon, flute, clarinet, violin, cello, piano and guitar. Using a large number of mixed authentic Solos of five instruments (excluding the guitar and piano), the method produced an average "Precise" recognition rate (See Section 13.3) of 56.9% for 2 - 5 instruments playing concurrently with resolution of 0.5second.

CHAPTER 4 TAXONOMIES

In pattern recognition, "taxonomy" means the classes into which the data is classified. While the great majority of work in this thesis deals with sound samples classified directly into classes according to the specific instrument which plays them, this is not the only option.
This short chapter brings a few examples of interesting alternative methods.

Hierarchical Taxonomies
In (Peeters and Rodet 2002), (Livshin, Peeters and Rodet 2003) and (Peeters and Rodet 2003) three different taxonomy levels were used - Pizzicato/Sustain, Instrument Families and Specific Instruments, as portrayed in Figure 4-A.

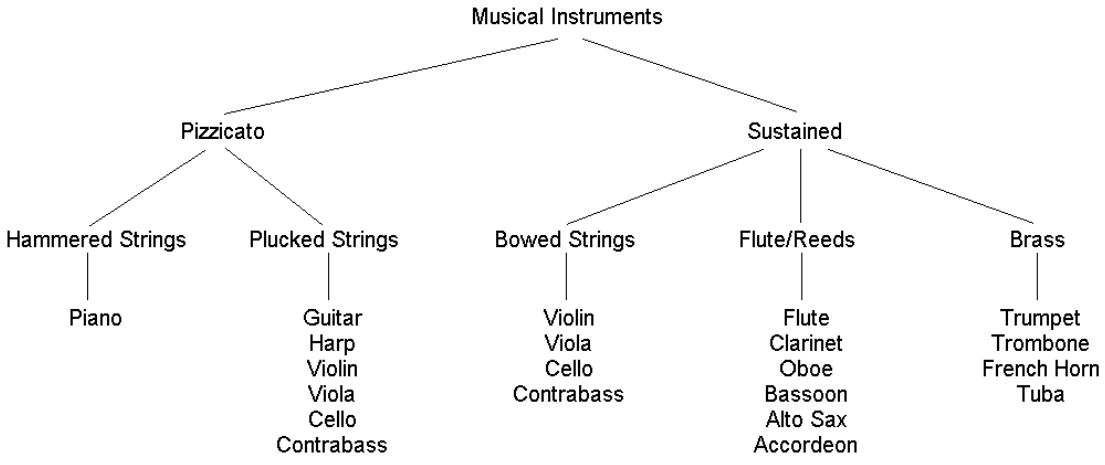

Figure 4-A. A hierarchical taxonomy of musical instruments according to instrument types

The Pizzicato/Sustained, Instrument Families and Specific Instrument classifications could be treated as three completely separate taxonomies, but also as levels in a hierarchical classification tree, where the ultimate goal is to classify the musical samples into specific musical-instrument classes.
While (Peeters and Rodet 2003) achieves higher recognition rates using hierarchical classification with this taxonomy, i.e., first classifying into Pizzicato/Sustained, then continuing down the tree with Instrument Families and finally classifying into Specific Instruments, in this thesis I use a simple, "flat" taxonomy of the Specific Instrument classes as my preliminary tests using the above 3-layer hierarchy did not show improvements in classification results over direct classification using the "flat", Specific Instrument Taxonomy.

Automatic Hierarchical Taxonomy
(Kitahara, Goto and Okuno 2004) created automatically a hierarchical taxonomy for separate tones, which they call "AcoustMIH". This taxonomy is based on acoustical similarity of musical instruments.

(Essid, Richard and David 2006) automatically create a hierarchical taxonomy which maximizes recognition rate for short fragments of MIP Jazz music, with taxonomy nodes corresponding to combinations of several instruments playing concurrently.

Non-Registered Instruments
In (Livshin and Dubnov 1998) monophonic sounds of "Unknown" musical instruments, i.e., instruments unrepresented in the Learning set, were classified into 11 instrument families, such as "Brass", "human voice", "non-pitched percussion", etc. In (Kitahara, Goto and Okuno 2004), sounds of non-registered instruments were classified into instrument families using an automatically generated, hierarchical taxonomy tree.

Perceptual Taxonomies
A somewhat different type of classification arises when the target is not an instrument class but a cluster of sounds that can be judged to be perceptually similar. In that case, classification does not rely on culturally shared labels but on timbre similarity measures and distance functions derived from psychoacoustical studies (Grey, 1977; Krumhansl, 1989; McAdams, Winsberg, de Soete and Krimphoff, 1995). This type of perceptual classification or clustering was addressed to provide indexes for retrieving sounds by similarity, using a query by example strategy.

CHAPTER 5 DATA SETS

During this work, several rather large sound databases were composed for research purposes from collected musical data. These databases are used for constructing the Learning and Test sets for the AMIR experiments - see the AMIR process overview, shapes #1 and #2 (Section 1.1).

5.1 SEPARATE TONES

"Separate Tone" databases are sound databases where each sample is a recording of a single musical note. The Attack transient is included in the sample, i.e., the note starts playing during the sample recording. The Release, i.e., the moment the instrument stops playing and the sound decays and disappears, is not necessarily included in the samples used in this work and many samples are truncated after a few seconds during the sustained part of the sound.

In this thesis, excerpts collected from 12 different commercial and research separate-tone sound databases are used; these databases are practically clean of noise. The databases contain sounds recorded in different recording environments, using different individual instruments (e.g., using different violins in each sound database). The sound sets span the entire pitch range of each of the instruments and include vibrato and non-vibrato sounds where applicable.

As I gradually gathered these sound database excerpts during my PhD studies, different chapters of the thesis may use different tone databases and different instruments according to what was available and applicable at the time of these experiments. Each thesis chapter specifies explicitly which instrument instances were used.

Preprocessing:

All sounds are sampled in 44 KHz, 16 bit, mono.

5.2 SOLO PERFORMANCES

A "Solo" means a monophonic or polyphonic recording of a musical piece performed by a single musical instrument. While plenty of Solo recordings of the piano exist, Solos of other instruments, especially monophonic and semi-monophonic ones are rather rare and hard to find. In this thesis I have insisted that each Solo should be an authentic recording of a different musical piece, recorded in a different environment and played by a different musician in order to perform a fair and meaningful AMIR cross-evaluation (see Chapter 9 and the rest of this section).
These facts have made obtaining my Solo collection (for research purposes) an extremely difficult and time consuming task, and a notable contribution in itself. This Solo database is available as part of the MCI project MusicDiscover (MCI 2004).

The Solo sound database consists of 108 different 'authentic' Solo performances of seven instruments: bassoon, clarinet, flute, classical guitar, piano, cello and violin. These performances, which include classical, modern and ethnic music, were gathered from commercial CD's (containing new or old recordings) and MP3 files collected on the Internet, played and recorded by professionals and amateurs.

As noted above, each Solo is performed by a different musician and there are no Solos taken from the same concert. During the evaluation process the same Solo was never used, neither fully or partly, in both the Learning and the Test sets. The reason for these limitations is that the evaluation process should reflect the system's ability to generalize – i.e., classify new musical phrases which were not learned, and were recorded in different recording conditions and played on different instruments and by different performers than the Learning set. We have proved (Livshin and Rodet 2003; Chapter 9) that the evaluation results of a classification system which *does* learn and classify sounds performed on the same instrument and recorded in the same recording conditions, even if the actual notes are of a different pitch, are much higher than when classifying sounds recorded completely independently. The reason is that such a "non-independent" evaluation process actually shows the system's ability to learn and then recognize specific characteristics of specific recordings and not its ability to generalize and recognize the "concept instrument".

Preprocessing:

In Solo recognition (Chapter 12) and in the Learning set of the Source-Reduction Technique (Chapter 13):

Only the left channel is taken out of stereo recordings[2]. A two-minute piece is taken from every Solo recording and cut into one-second cuts with 50% overlap – a total of 240 cuts out of each Solo[3]. The feature descriptors are computed on each one-second Solo-cut separately.

In the Learning set of the Harmonic-Resynthesis Technique (Chapter 13):

F0-estimation is performed on the Solos and detected notes over a minimal length are resynthesized using Additive Synthesis. See Section 13.1.3.2.

Solo List

	Instruments	Num. Solos	Num. Samples
Monophonic and Semi-monophonic	Bassoon	10	2182
	Clarinet	21	4178
	Flute	14	2724
	Cello	17	3368
	Violin	14	2915
Polyphonic	Piano	16	3619
	Guitar	16	3478
Total:			
Mono&Semi	5	76	15367
All	7	108	22464

Table 5-A. Solo collection

Table 5-A shows the list of Solos in the Solo collection. There are monophonic instruments – bassoon, clarinet and flute, semi-monophonic (where an occasional polyphony of two sounds occurs) – violin and cello, and polyphonic – piano and guitar.

In each row, along with the instrument name, the number of Solos of that instrument is written and the total number of one-second cuts from these Solos.

[2] It could be argued that it is preferable to use a mix of both channels. Which method is actually better, depends on the specific recording settings of a musical piece.

[3] The "Num. Samples" column in Table 5-A shows that because some Solos are shorter than two minutes the total length of the Solos is actually 22464/2=187.2 minutes and not 108*2=216 minutes.

5.3 AUTHENTIC DUOS - REAL PERFORMANCES

	Num. Duos	Bassoon	Clarinet	Flute	Cello	Violin	Guitar	Piano
Only monophonic and Semi-monophonic	**3**		X	X				
	5				X	X		
	3			X	X			
	1			XX				
	1	X		X				
Polyphonic instruments present	**1**		X				**X**	
	1			X			**X**	
	1		X					**X**
	2	X						**X**
Total:								
Mono&Semi	**13**	**1**	**3**	**8**	**8**	**5**		
All	**18**	**3**	**5**	**9**	**8**	**5**	**2**	**3**

Table 5-B. Authentic Duos

Table 5-B describes the authentic Duo recordings used in Chapter 13. "Authentic" means recordings of actual Duo performances, as opposed to "artificial" Mixtures of Solos which are described in the next section. Each row in the table shows how many Duos of a specific instrument combination are present. The last line shows the total number of Duos in which each instrument plays.

5.4 MULTI-INSTRUMENTAL SOLO MIXES

It is very difficult to obtain authentic recordings of "real" music along with their exact transcription, which is required in order to perform precise AMIR evaluation. The intuitive solution, music resynthesized from MIDI files, such as used in (Kitahara et al. 2007), while producing exact performances of the symbolic notation, is rather too different from real musical recordings to substitute them - the sounds are synthetically clean, lacking articulations such as legato, very similar to each other with sounds of the same pitch being completely identical and note boundaries being too well defined. Therefore, evaluation with resynthesized MIDI files is not likely to indicate the ability of the recognition program to handle authentic music recordings.

Artificially mixed authentic Solo performances are a compromise; on the negative side, the Solo sounds of the different instruments do not influence each other and do not create real, common room echo. Unfortunately also, musical composition rules could not be assumed to apply to the mixtures as the mixed Solos have no relevancy to each other.
On the positive side, all the instrument articulations are present, the sounds are authentic - unlike music resynthesized from samples, the notes are not identical each time they are played, and while the resulting music is polyphonic the files could be labeled much easier than actual MIP musical recordings. In most of today's studio recordings, a similar

process takes place anyway – each instrument is recorded on a separate track and at the production stage the tracks are intermixed.

The instrument labeling of the Solo-mixes was performed in the following way:

- A multiple-*f0* estimation program was ran on the original Solos and on their mixture
- Notes detected in the Solo mixtures, which lasted for at least a specific length of time (e.g., 0.25second or more), were searched in the *f0*-estimation data of the Solos which were mixed in that mixture, in the appropriate temporal locations
- When a note from the Solo-mix was found in a participating Solo, it was labeled by the same instrument as the one playing this Solo
- Notes found only in the mixture but not in the participating Solos were discarded

This labeling method works rather well. The average percentage of discarded notes, i.e., those found in the mixes but not found in the Solos, is only 6% as indicated in the results-table in Section 13.3

When creating a Solo-mix database for a specific number of polyphonic notes, the coupling of Solos for creation of each multi-instrumental mix is done by randomly selecting Solos from different classes depending on desired polyphony level, then selecting randomly a 20-second segment out of each of these Solos (hence the length of each solo-mix is also 20 seconds), normalizing it and mixing these segments together. Each Solo is used only once in a Solo-mix database for a specific polyphonic level.
At classification time, the Solos mixed in the classified mixture are removed from the Learning set for keeping the evaluation independent.

Unfortunately, polyphonic instruments such as the guitar and piano are not appropriate for this labeling method as the employed *f0*-estimation program could not operate completely accurately on polyphonic Solos and "loses" some of the notes. Another problem is that the polyphonic level is required to remain constant for independently testing different voice numbers; the polyphony of the guitar and piano varies constantly throughout their Solo performances. Consequently, the musical instruments mixed are the bassoon, clarinet, flute, violin and cello, excluding the piano and guitar (which are included in the Solo and authentic Duo collections).

The total number of Duo mixes is 34, Trios – 20, Quartets – 14 and Quintets – 10.

See Section 15.4 for several other solutions to the evaluation problem.

CHAPTER 6 FEATURE DESCRIPTORS

In order to perform AMIR of musical instrument sounds a collection of various statistics is computed on each sound sample; these statistics are called "Features". To understand where feature computation is integrated into the complete AMIR process, see the AMIR process overview (Section 1.1), shape #4.

Each feature may have one or more values which I shall refer to as "Descriptors". Rather than working with raw sound data, the classification algorithm deals with feature descriptor vectors of the sound samples. While saving space and computation time, most importantly, feature descriptors allow using knowledge and heuristics for intelligently distinguishing between the sounds rather than performing a dumb comparison of the sound waveforms.

In order to encapsulate various characteristics of the sound signals an extensive set of features is used, consisting of 62 different feature types. Some of these features are comprised of several values while some other feature types are computed several times using different parameter types; this results in a total of 513 different feature descriptors. For example, Spectral Kurtosis "flavors" include Kurtosis computed on the linear spectrum, the log-spectrum, the harmonics envelope, etc., while the MFCC feature is a vector of 12 coefficients.

Most of the feature descriptors are "frame based", meaning that they are first computed on each frame of a Short-Time Fourier Transform (STFT) of the signal (Allen 1977; Allen and Rabiner 1977), using a sliding window of 60 ms with a 66% overlap, and then

the average and standard deviation over all these frames are used as the feature Descriptors.

After computation, all feature Descriptors are normalized to the range of [0 - 1] using Min-Max Normalization.

The feature computation routines were designed and written by Geoffroy Peeters (Peeters 2004) as part of project Cuidado using Matlab (Matlab) and utilizing functionality from several sound toolboxes, such as the VoiceBox toolbox (Brookes 1998).

6.1 FEATURE TYPES

6.1.1 TEMPORAL FEATURES

These features are computed directly on the signal

6.1.2 ENERGY FEATURES

Features referring to various energy content of the signal

6.1.3 SPECTRAL FEATURES

Features computed from the Short Time Fourier Transform (STFT) of the signal

6.1.4 HARMONIC FEATURES

Features computed from the Sinusoidal Harmonic modeling of the signal (harmonic series)

6.1.5 PERCEPTUAL FEATURES

Features computed using a model of the human hearing process. See (Stevens, Volkmann and Newman 1937) for the Mel Scale and (Zwicker and Terhardt 1980) for the Bark Scale

6.2 FEATURE LIST

Note that as the feature set was enlarged with time, in several sections of the thesis an older, smaller, more compact set of features is used. As it makes no difference whether the Compact set or the Full set is used in these sections, the experiments were not repeated with the Full set. The **Compact Feature Descriptor set** has 45 features with a total of 162 descriptors while the **Full Feature Descriptor set** has 62 features with 512 descriptors as noted above. The Compact set includes only the mean of the frame-based features while the Full set has also the standard deviation.

Table 6-A presents a full list of the feature descriptors. For full explanation of all the features and their computations see (Peeters 2004).

The "Related References" column provides references for the features[4]. Features without references are mostly variations of other features and lack individual references, for example Harmonic Spectral Skeweness is actually Spectral Skewness computed using the Harmonic Series. References marked "In instrument recognition" show sources which used a particular feature specifically for AMIR.

The "#Desc" column lists the number of descriptors per feature in the form 'X / Y'. On the left of the slash symbol ('X') is the number of descriptors in the Full Feature Descriptor set, while on the right of the slash, underlined ('Y'), is the number of descriptors in the Compact Feature Descriptor set. Cells with a single number show descriptors present only in the Full set.

The "STFT" column indicates whether a feature is computed on the STFT of the signal; if it is, then the average and the standard deviation of this feature computed over the STFT frames are included in the Full Feature Descriptor set, thus doubling the number of descriptors for this feature.

Several of the feature descriptors are part of the MPEG-7 standard for audio. MPEG-7 is an ISO/IEC standard (ISO/IEC JTC1/SC29/WG11) developed by MPEG (Moving Picture Experts Group), formally named "Multimedia Content Description Interface", which provides a rich set of standardized tools to describe multimedia content. The reference (MPEG-7 2004) applies to all the MPEG-7 features in the table; the formal "MPEG-7" names of these features are listed in the "Related References" column.

[4] Many of the references in the table complement those which appear in (peeters 2004), mentioned above.

Feature	STFT	#Desc.	Related References
Temporal Features			
Global Temporal Features			
Log Attack Time	N	1 / 1	**Attack time importance – (Eagleson and Eagleson 1947; Saldanha and Corso 1964; Elliot 1975)** **Log Attack Time - (Krimphoff, McAdams and Winsberg 1994)** **(Peeters, McAdams and Herrera 2000)** **"LogAttackTime" in MPEG-7**
Temporal Increase	N	1	
Temporal Decrease	N	1 / 1	
Temporal Centroid	N	1 / 1	**(Peeters, McAdams and Herrera 2000)** **"TemporalCentroid" in MPEG-7**
Effective Duration	N	1 / 1	
Instantaneous Temporal Features			
Signal Auto-correlation function	Y	12 / 12	**In instrument recognition – (Brown 1998)**
Zero-crossing rate	Y	1	**In instrument recognition – (Smith, Murase and Kashino 1998)**
Energy Features			
Total energy	Y	1	**(Kaminskyj and Materka 1995)** **"AudioPower" in MPEG-7**
Total energy Modulation (frequency, amplitude)	N	2 / 2	**In instrument recognition – (Martin and Kim 1998)**
Total harmonic energy	Y	1	
Additive harmonic energy	Y	1	
Total noise energy	Y	1	
Additive noise energy	Y	1	
Spectral Features			
Spectral Shape			
Spectral centroid	Y	6 / 1	**As "Brightness" - (Grey and Gordon 1978)** **As "Spectral Centroid" - (Beauchamp 1982)** **In instrument recognition – (Fujinaga 1998)** **"AudioSpectrumCentroid" and "SpectralCentroid" in MPEG-7**
Spectral spread	Y	6 / 1	**In instrument recognition – (Fujinaga 1998)** **"AudioSpectrumSpread" in MPEG-7**
Spectral skewness	Y	6 / 1	**In instrument recognition – (Fujinaga 1998)**
Spectral kurtosis	Y	6 / 1	**(Dwyer 1983)** **In instrument recognition – (Fujinaga 1998)**
Spectral slope	Y	6 / 1	
Spectral decrease	Y	1 / 1	
Spectral rolloff	Y	1 / 1	
Spectral variation	Y	3 / 1	
Global spectral shape description			
MFCC	Y	12 / 12	**Cepstrum - (Bogert, Healy and Tukey 1963)** **MFCC - (Davis and Mermelstein 1980)** **In instrument recognition – (De Poli and Prandoni 1997)**
Delta MFCC	Y	12 / 12	**(Soong 1988), (Rabiner 1993)**
Delta Delta MFCC	Y	12 / 12	
Harmonic Features			**Additive Synthesis - (Risset 1985)** **Partial Tracking – (Depalle, Garcia and Rodet 1993)**
Fundamental frequency	Y	1 / 1	**Maximum likelihood algorithm – (Doval and Rodet 1993)** **"AudioFundamentalFrequency" in MPEG-7**

Fundamental fr. Modulation (frequency, amplitude)	Y	2	**In instrument recognition – (Martin and Kim 1998)**
Noisiness	Y	1 / 1	**"AudioHarmonicity" in MPEG-7**
Inharmonicity	N	1 / 1	**Piano - (Fletcher, Blackham and Stratton 1962)** **Saxophone - (Freedman 1967)** **In instrument recognition – (Martin 1999)**
Harmonic Spectral Deviation	Y	3 / 1	**(Peeters, McAdams and Herrera 2000)** **"HarmonicSpectralDeviation" in MPEG-7**
Odd to Even Harmonic Ratio	Y	3 / 1	**In instrument recognition – (Martin and Kim 1998)**
Harmonic Tristimulus	Y	9 / 3	**(Pollard and Jansson 1982)**
Harmonic Spectral Shape			**The computations of these features are equivalent to corresponding spectral features, but computed on the harmonic series.**
Harmonic Spectral centroid	Y	6 / 1	**(Peeters, McAdams and Herrera 2000)** **"HarmonicSpectralCentroid" in MPEG-7**
Harmonic Spectral spread	Y	6 / 1	**(Peeters, McAdams and Herrera 2000)** **"HarmonicSpectralSpread" in MPEG-7**
Harmonic Spectral skewness	Y	6 / 1	
Harmonic Spectral kurtosis	Y	6 / 1	
Harmonic Spectral slope	Y	6 / 1	
Harmonic Spectral decrease	Y	1 / 1	
Harmonic Spectral rolloff	Y	1 / 1	
Harmonic Spectral variation	Y	3 / 1	**(Peeters, McAdams and Herrera 2000)** **"HarmonicSpectralVariation" in MPEG-7**
Perceptual Features			**Mel Scale - (Stevens, Volkmann and Newman 1937)** **Bark Scale - (Zwicker and Terhardt 1980)**
Loudness	Y	1 / 1	**(Zwicker 1990)** **(Moore, Glasberg and Baer 1997)**
Relative Specific Loudness	Y	24 / 20	**(Zwicker 1990)**
Fluctuation strength	N	24 / 24	**As "spectral irregularity" - (Krimphoff, McAdams and Winsberg 1994)** **As "spectral flux" - (Krumhansl 1989)**
Mean Fluctuation strength	N	1 / 1	**As "spectral irregularity" - (Krimphoff, McAdams and Winsberg 1994)** **As "spectral flux" - (Krumhansl 1989)**
Roughness	N	24 / 24	**(Von Békésy 1960), (Terhardt 1974)**
Mean Roughness	N	1 / 1	**(Von Békésy 1960), (Terhardt 1974)**
Sharpness	Y	1 / 1	**(Aures 1984)**
Spread	Y	1 / 1	
Perceptual Spectral Envelope Shape			**The computations of these features are equivalent to corresponding spectral or harmonic features, but computed on perceptual bands.**
Perceptual Spectral centroid	Y	6	
Perceptual Spectral spread	Y	6	
Perceptual Spectral skewness	Y	6	
Perceptual Spectral kurtosis	Y	6	
Perceptual Spectral Slope	Y	6	

Perceptual Spectral Decrease	Y	1	
Perceptual Spectral Rolloff	Y	1	
Perceptual Spectral Variation	Y	3	
Odd to Even Band Ratio	Y	3	
Band Spectral Deviation	Y	3	
Band Tristimulus	Y	9	
Various features			
Spectral flatness	Y	4 / 4	**(Jayant and Noll 1984)** **In instrument recognition - (Herre, Allamanche and Hellmuth, 2001)** **"AudioSpectrumFlatness" in MPEG-7**
Spectral crest	Y	4 / 4	**(Jayant and Noll 1984)**

Table 6-A. List of feature descriptors and references

CHAPTER 7 FEATURE WEIGHTING AND SELECTION

Not all feature descriptors provide the same amount of distinguishing AMIR information; some may be simply redundant while others could even confuse the classification algorithm, diminishing the recognition rate, if not handled properly.

Feature Weighting
In a feature weighting process, new "super-features" are created from weighted combinations of the original feature descriptors in order to maximize classification performance. In many cases the number of these "super-features" is smaller than the number of original features, thus requiring less memory, storage and classification time.

Feature Selection
Sometimes it is desirable to choose only a subset of the features. Knowing which feature descriptors are the most important ones for class separation has many uses:

- Find out which feature descriptors are required and which ones are redundant
- Discover which qualities distinguish best among the various sound sources
- Save descriptor calculation time
- Save descriptor storage space
- Reduce memory requirements
- Reduce classification time

To understand where feature weighting and selection are integrated into the complete AMIR process, see the AMIR process overview (Section 1.1), shape #6.

Unfortunately finding the ideal feature set is impossible without testing all feature combinations, which is an NP-hard problem (Cover and Van Campenhout 1977); this is one reason why different feature selection algorithms (with non-exponential complexity) may select a somewhat different "best feature set" using the same data. In this thesis two feature selection algorithms are used – GDE and CFS. My GDE algorithm, being LDA based, has the advantage of producing slightly higher recognition rates when the selected descriptors are used for classification with LDA+KNN (discussed in Section 8.3) – the classification algorithm used in the great majority of experiments in this thesis. CFS, however, has the advantage of dealing "harsher" with correlated variables and thus producing somewhat more "interesting" variable lists than the seemingly more erratic GDE results, thus providing the user with a better insight into the AMIR distinguishing sound qualities, especially when the Complete Feature Descriptor set is used which has many more descriptor "flavors" than the Compact Feature set (see Section 6.2).
For a discussion of relevance vs. usefulness and definitions of the various notions of relevance, see the review articles of Kohavi and John (1997) and Blum and Langley (1997).

Disclaimer - this chapter does not attempt to cover the Pattern-Recognition fields Feature Selection and Feature weighting which are very considerable, but rather sparingly discuss the techniques used in this thesis. For a comprehensive review of variable and feature selection, see (Guyon and Elisseeff 2003).

7.1 LINEAR DISCRIMINANT ANALYSIS

The objective of Linear Discriminant Analysis (LDA) (McLachlan 1992) feature weighting algorithm is to perform dimensionality reduction while seeking to find discriminating directions along which the classes are best separated.

Methodology

- Suppose there are C classes
- Let μ_i be the mean vector of class i, $i = 1, 2, \ldots, C$
- Let M_i be the number of samples y within class i, $i = 1, 2, \ldots, C$,
- Let $M = \sum_{i=0}^{C} Mi$ be the total number of samples. and

Within-class scatter matrix:

$$S_w = \sum_{i=1}^{C} \sum_{j=1}^{M_i} (y_j - \mu_j)(y_j - \mu_j)^T$$

Between-class scatter matrix:

$$S_b = \sum_{i=1}^{C} (\mu_i - \mu)(\mu_i - \mu)^T$$

where

$$\mu = \frac{1}{C} \sum_{i=1}^{C} \mu_i \quad \text{(mean of entire data set)}$$

- LDA computes a transformation that maximizes the between-class scatter while minimizing the within-class scatter (called *Fisher criterion*):

$$maximize \frac{\det(S_b)}{\det(S_w)}$$

Linear transformation implied by LDA

The linear transformation is given by a matrix U which columns are the eigenvectors of $S_w^{-1} S_b$ (called *Fisherfaces*).

$$\begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_k \end{bmatrix} = \begin{bmatrix} u_1^T \\ u_2^T \\ \dots \\ u_k^T \end{bmatrix} (x - \mu) = U^T (x - \mu)$$

- The eigenvectors are solutions of the *generalized eigenvector problem*:

$$S_b u_k = \lambda_k S_w u_k$$

There are at most $C - 1$ non-zero generalized eigenvectors (i.e.,$K < C$)

Existence of S_w^{-1}

- If S_w is non-singular, a conventional eigenvalue problem is obtained by writing:

$$S_w^{-1} S_b u_k = \lambda_k u_k$$

- In practice, S_w is often singular since some variables could be partly dependent. To alleviate this problem, two projections can be performed:

  (1) Principal Component Analysis (PCA) (Pearson 1901) is first applied to the data set to reduce its dimensionality.

$$\begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} --> PCA --> \begin{bmatrix} p_1 \\ p_2 \\ \dots \\ p_k \end{bmatrix}$$

  (2) LDA is then applied to further reduce the dimensionality to $C - 1$.

$$\begin{bmatrix} p_1 \\ p_2 \\ \dots \\ p_k \end{bmatrix} --> LDA --> \begin{bmatrix} z_1 \\ z_2 \\ \dots \\ z_{C-1} \end{bmatrix}$$

In most of the experiments in this thesis LDA is used in conjunction with KNN as the classification method. See Section 8.3.

7.2 GRADUAL DESCRIPTOR ELIMINATION (GDE) USING DISCRIMINANT ANALYSIS

My Gradual Descriptor Elimination feature selection algorithm provides the dependencies between the number of descriptors and the average success of the classifications, estimating which are the best n descriptors to keep for a desired number of descriptors or a desired recognition rate[5].

7.2.1 THE GDE ALGORITHM

- Suppose there are C classes. Let $K = C - 1$
- Suppose there are F feature descriptors of S samples in feature descriptor matrix D_{SF} of the sample database.

1. A "classification success" percentage for D_{SF} is estimated, using for example Leave-One-Out (LOO) or Self-Classification (see Chapter 9 for Evaluation Techniques), and recorded along with the current list of descriptors.

2. Let $U_{FK} = LDA(normalize\ (D_{SF}))$

   Each **column** *f* in D_{SF} (containing the values of descriptor *f*), is normalized to [0 - 1]. A linear discriminant analysis transformation matrix is calculated on the normalized D_{SF}.

3. An 'Importance Coefficient' P_f is calculated for each descriptor:

   - Let $V_{FK} = abs \begin{bmatrix} u_{1k} \cdot \lambda_k \\ u_{2k} \cdot \lambda_k \\ \dots \\ u_{Fk} \cdot \lambda_k \end{bmatrix}$

     Where: λ_k is the eigenvalue corresponding to the eigenvector in column k of U_{FK}, $1 \le k \le K$

     The K columns of U_{FK} (the *Fisherfaces*), are multiplied by the corresponding eigenvalues (their *characteristic roots*) and converted to distances. Note that the ratio of the eigenvalues indicates the relative

[5] Obviously as long as this desired rate is smaller than the maximum recognition rate available

discriminating power of the discriminant functions. If the ratio of two eigenvalues is 1.4, for instance, then the first discriminant function accounts for 40% more between-class variance in the dependent categories than does the second discriminant function. See Section 7.1 for computation of LDA.

$$\text{Let } P_f = \sum_{k=1}^{K} v_{fk}$$

Where: $1 \leq f \leq F$

Each row of V_{FK} is summed, providing the 'Importance Coefficient' P_f for descriptor f.

4. The descriptor with the lowest P_f is removed.

5. Steps 1 - 4 are repeated until no descriptors are left.

7.2.2 EXAMPLE EVALUATION

7.2.2.1 Sound Database

Evaluation is performed using an excerpt from the extensive IRCAM Studio OnLine a.k.a. "SOL" (Ballet 1998) separate tone database (see Section 5.1). This excerpt contains 1325 sound samples of 20 musical "instruments" - guitar, harp, violin (pizzicato and sustained), viola (pizzicato and sustained), cello (pizzicato and sustained), contrabass (pizzicato and sustained), flute, clarinet, oboe, bassoon, alto sax, accordion, trumpet, trombone, French horn and tuba.

These sounds are classified using the three taxonomies in Figure 4-A: pizzicato/sustained (2 classes), instrument families – plucked strings, bowed strings, flute/reeds and brass (4 classes) and instrument names (20 classes). All the samples are two-seconds long, monophonic and sampled in 44.1 KHz with 16 bit resolution.

7.2.2.2 Evaluation

GDE was applied to the SOL database excerpt using the 3 different taxonomies. The Compact Feature Descriptor set is used (see Section 6.2), containing 162 descriptors. The following graphs depict the Leave-One-Out recognition rate (see Section 9.7.3) against the retained number of descriptors in the different taxonomies. Note that sometimes the results actually improve after removing a misleading descriptor.

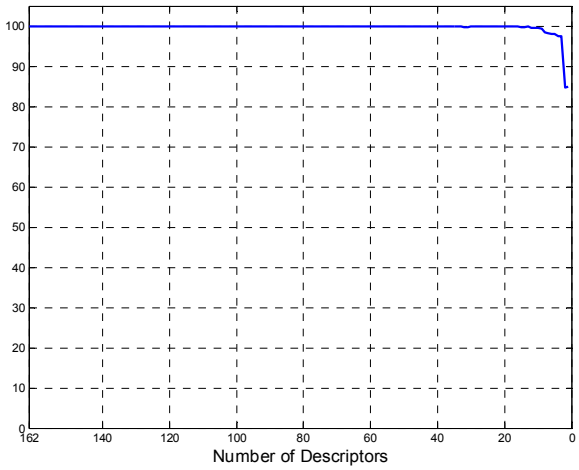

Figure 7-A. GDE using the Pizzicato/Sustain taxonomy

The results in Figure 7-A show it is possible to decrease the number of descriptors from 162 down to 3 and still get an LOO recognition rate of 97.43% for the Pizzicato/Sustain taxonomy.

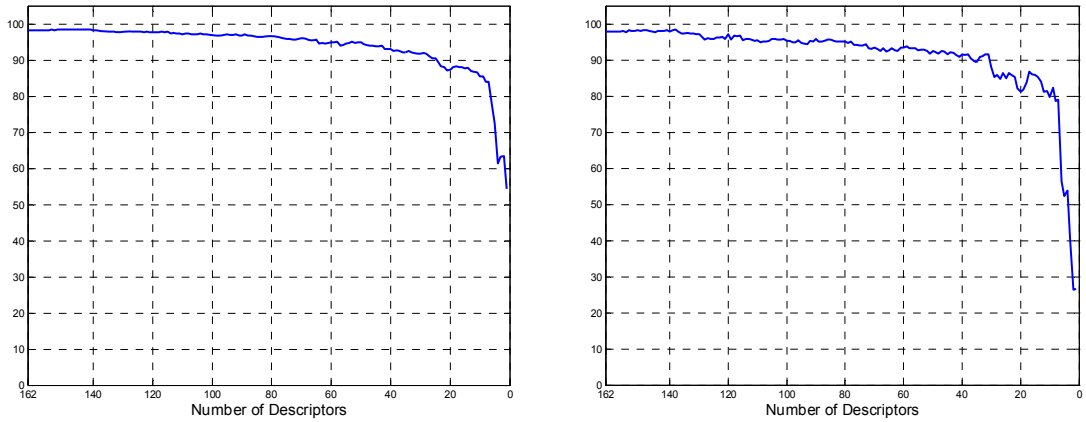

Figure 7-B. LOO recognition rates using GDE-selected descriptors with the Instrument Families (on the left) and the Specific Instruments taxonomies

For the Instrument Families and the Specific Instrument taxonomies, LOO results have decreased below 90% around 30 descriptors.

LOO does not use random choices, therefore it has the advantage here over other database self-consistence evaluation methods which do use random numbers, such as Self-Classification (see Section 9.4.1), by giving a constant and consistent recognition rate not influenced by the random choices. Nevertheless, in order to show that LOO results correspond closely to the results of the more common Self-Classification method, Figure 7-C depicts the average results of Self-Classification using the same descriptors as

above (selected by the GDE Algorithm), performing 20 Self-Classification rounds, each time randomly selecting a Learning set consisting of 66% of the samples.

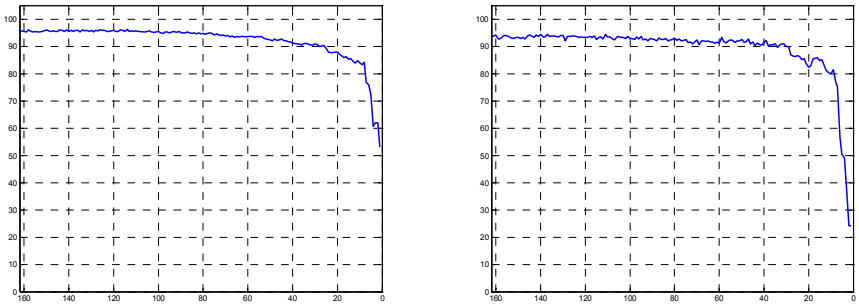

Figure 7-C. Self-Classification rates using GDE descriptors with the Instrument Families (on the left) and the Specific Instruments taxonomies

It is easy to see that the result charts for using Self-Classification and LOO are very similar.

7.3 CORRELATION-BASED FEATURE SELECTION (CFS)

The entropy-based CFS algorithm scores and ranks the "worth" of subsets of features by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred. As the feature space is very large and checking all the feature combinations is not practical, CFS starts with an empty set and adds features using a stepwise forward search method, searching the space of feature subsets by greedy hillclimbing augmented with a backtracking facility. For further reading on CFS see (Hall 1998). In this thesis, the WEKA data-mining software (Witten and Frank 2005) implementation of the CFS algorithm is used.

CHAPTER 8 CLASSIFICATION ALGORITHMS

After the feature descriptors are computed on the sound samples, each sample is classified as belonging to a musical instrument by the classification algorithm. To understand where the classification algorithm is integrated into the complete AMIR process, see the AMIR process overview (Section 1.1), shape #10.

A classification algorithm is an algorithm which receives as input two sets of data samples – the Learning Set and the Test Set. The Learning Set contains along with each data sample its corresponding class. The purpose of the classification algorithm is to classify the data samples in the Test Set by comparing them with the samples in the Learning Set.

There are many different types of classification algorithms and covering them is out of scope of this work. There are many books on classification algorithms; for further reading on the subject, see (James 1985) for example.
The purpose of this chapter is to introduce only the algorithms used in this thesis and explain briefly why these were chosen.

8.1 NEURAL NETWORKS

Disclaimer - Artificial Neural-Networks are a considerable research field by its own rights and way beyond the scope of this simplistic Section. For comprehensive information on Neural Networks and the backpropagation algorithm read (Aleksander and Morton 1990).

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

The fundamental building block in an Artificial Neural Network is the mathematical model of a neuron. The three basic components of the (artificial) neuron are:

1. The synapses or connecting links that provide weights, w_j, to the input values, x_j for $j = 1, \ldots m$;
2. An adder that sums the weighted input values to compute the input to the activation function $v = w_0 + \sum_{j=1}^{m} w_j x_j$, where w_0 which is called the *bias* is a numerical value associated with the neuron. It is convenient to think of the bias as the weight for an input x_0 whose value is always equal to one, so that $v = \sum_{j=0}^{m} w_j x_j$;
3. An activation function g (also called a squashing function) that maps v to $g(v)$ the output value of the neuron. This function is a monotone function.

While there are numerous different artificial neural network architectures that have been studied by researchers, the most successful applications of neural networks have been multilayer feedforward networks. These are networks in which there is an input layer consisting of nodes that simply accept the input values and successive layers of nodes that are neurons. The outputs of neurons in a layer are inputs to neurons in the next layer. The last layer is called the output layer. Layers between the input and output layers are known as hidden layers.

8.1.1 BACKPROPAGATION (BP)

A backpropagation neural network is a feedforward network which uses the backpropagation algorithm for its training process.

In order to train a neural network to perform some task, the weights of each unit must be adjusted in such a way that the error between the desired output and the actual output is reduced. This process requires that the neural network compute the error derivative of the weights (**EW**). In other words, it must calculate how the error changes as each weight is increased or decreased slightly. The backpropagation algorithm is the most widely used method for determining the **EW**.

The backpropagation algorithm is easiest to understand if all the units in the network are linear. The algorithm computes each **EW** by first computing the **EA**, the rate at which the error changes as the activity level of a unit is changed. For output units, the **EA** is simply the difference between the actual and the desired output. To compute the **EA** for a hidden unit in the layer just before the output layer, all the weights between that hidden unit and the output units to which it is connected are first identified. Those weights are then multiplied by the **EA**s of those output units and the products are added. This sum equals the **EA** for the chosen hidden unit. After calculating all the **EA**s in the hidden layer just before the output layer, the **EA**s can be computed in similar fashion for other layers, moving from layer to layer in a direction opposite to the way activities propagate through the network. This is what gives backpropagation its name. Once the **EA** has been computed for a unit, it is straightforward to compute the **EW** for each incoming connection of the unit. The **EW** is the product of the EA and the activity through the incoming connection.

Note that for non-linear units such as tansig, which is used in this work, the backpropagation algorithm includes an extra step. Before backpropagating, the **EA** must be converted into the **EI**, the rate at which the error changes as the total input received by a unit is changed.

The Backpropagation Neural network is used in Chapter 9 and has a single hidden layer of 80 neurons, using the tan-sigmoid transfer function in the input and hidden layers. It is trained using the Conjugate Gradient with Powell/Beale Restarts algorithm (Powell 1977), until a Mean Square Error of 0.004 is reached.

Advantages and disadvantages of BP[6]
Advantages: can classify non-linearly separable data, does not require normal distribution of the classes, deals well with redundant or dependent variables, rapid classifications.

[6] These advantages and disadvantages apply also to Support Vector Machines (SVM) (Boser, Guyon and Vapnik 1992) and to many other types of Neural Networks in addition to backpropagation.

Disadvantages: very slow learning process and relatively complicated to predict, selecting good starting parameters and a stopping condition which do not cause under/over-fitting to input data is somewhat a matter of trial and error.

8.2 K-NEAREST NEIGHBORS (KNN)

The classification experiments in this work are performed mostly using the KNN classification algorithm preceded by LDA. The K-nearest neighbor (Fix and Hodges 1951) is a supervised non-linear classification algorithm where the classifier is based on a majority voting scheme and does not use a model to fit the Learning set.

Given an unknown feature vector **x** and a distance measure, then:

- Out of the N vectors in the Learning set, identify the k vectors which are the "nearest neighbors" of **x**, i.e., have the minimal distance to **x**.
- Out of these k vectors, identify the number of vectors k_i, that belong to each class ω_i, $i = 1, 2, \ldots, C$. Obviously, $\sum_i k_i = k$.
- Assign **x** to the class ω_i with the maximum number of vectors k_i.

The most frequently used distance measure for KNN and the one used in this work is the Euclidean distance, defined for two n dimensional points P and Q, as: $d = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$

8.2.1 SELECTION OF "K"

In this work, the value of K (number of neighbors) is chosen from a range of values, usually 1 – 80, in order to find the one which produces the best results.

One of these two methods is used here for best-K selection:

- LOO or one of the "Minus-1" cross-validation methods is performed on the Learning set with a range of K values, and the K which produces the best results is selected; this method imitates a "real-world" situation where the Learning set is provided and the Testing sets are completely unknown a-priori.
- One of the "Minus-1" evaluation methods is performed with different K values and the K leading the best average score on the complete data set is selected; this method is useful when the evaluation data originates from different sources, such as in the case of "Minus-1-Solo". This method produces K values which are suited for the general, or "concept" classification, as the Learning and Test sets in the "Minus-1" methods are independent in each evaluation round and thus the selected K suits best the largest number of independent classifications. See Chapter 9 for more details on evaluation methods.

8.3 CHOSEN CLASSIFICATION METHOD - "LDA+KNN"

In most classifications performed in this work the data is first transformed using LDA and then classified with KNN.

There are several advantages of using this sequence of methods:
LDA weighs out the descriptors, thus eliminating the problem of redundant or dependent variables which are the main weakness of the KNN algorithm, which treats all variables as having equal importance. LDA projects the samples into the linear plane which best clusters together the samples of each class (minimizing "within-class scatter") while distancing the different classes from each other (maximize "between-class scatter"); this helps KNN which requires samples from the same class to be close together and away from samples of other classes.

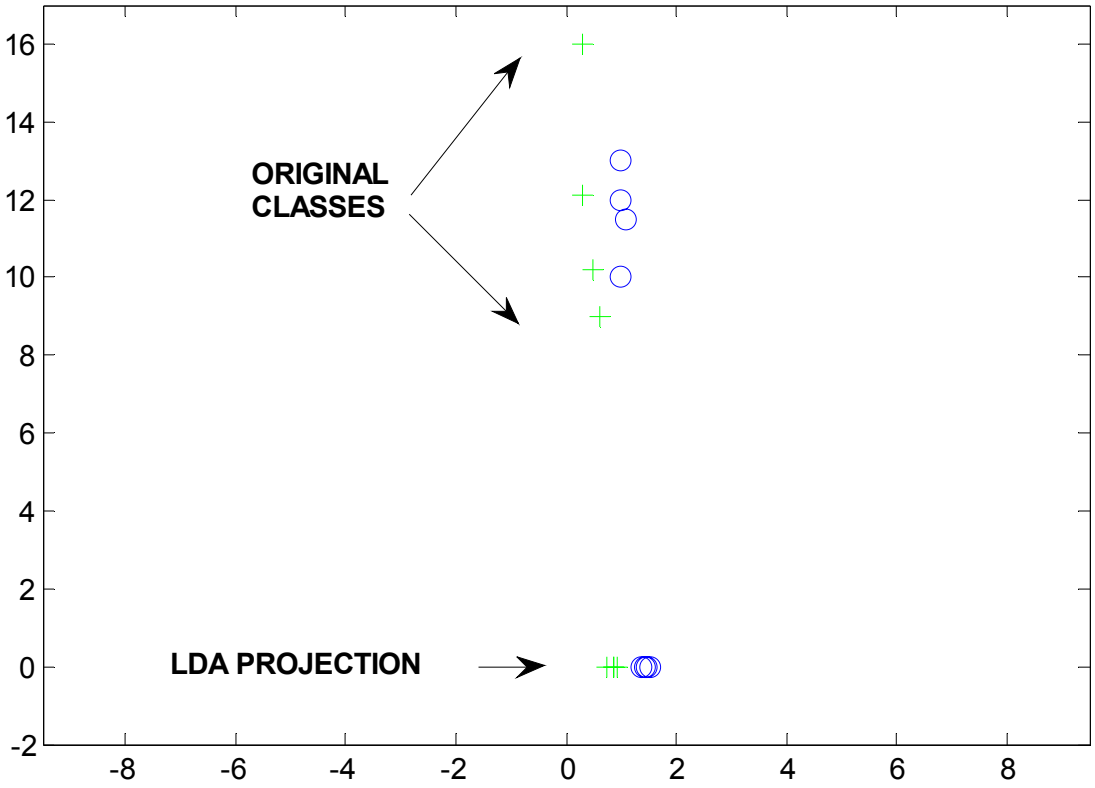

**Figure 8-A. Two classes ('o' and '+') and their LDA projection.
Example of LDA "saving" KNN by diminishing the influence of problematic descriptors and clustering classes together**

Figure 8-A exemplifies two classes and the resulting LDA projection - class #1 is marked with '+' signs and class #2 is marked with '**O**'s. The points marked "ORIGINAL CLASSES" are the samples of the original two-dimensional classes, while the points marked "LDA PROJECTION" (on location 0 of the Y axis) are these original samples transformed by LDA into one-dimensional space.

Figure 8-A clearly shows how all the samples of the original '+' class would have been classified incorrectly by the distance-based KNN because the closest neighbors of all these samples are of the 'O' class. After the LDA transform, however, both the '+' and 'O' classes are clustered together and "ready" for KNN.

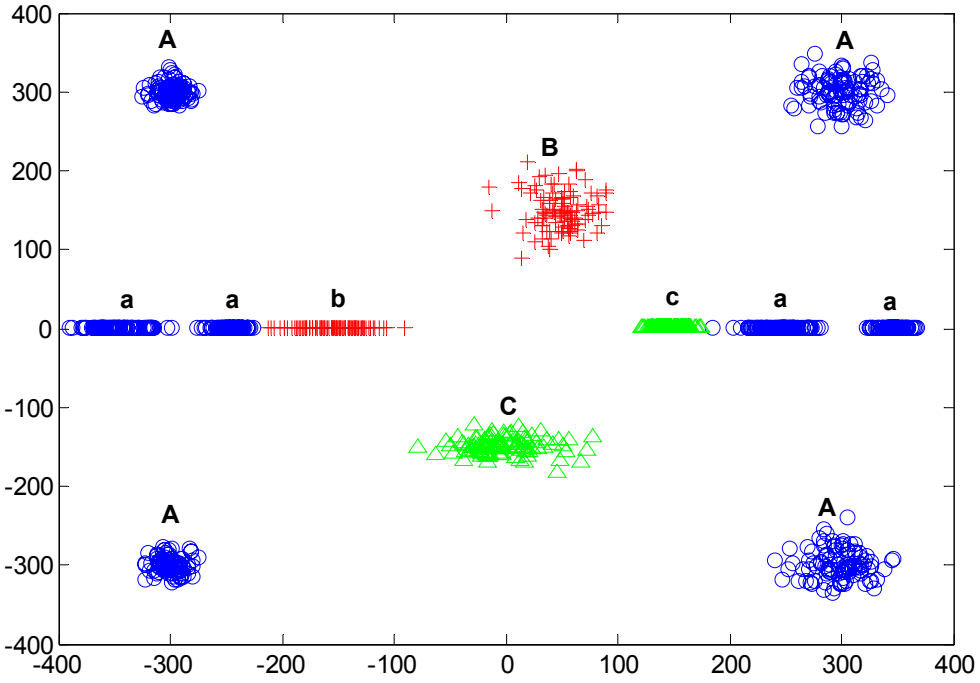

Figure 8-B. Three classes (A - '+', B - 'O' and C - 'Δ') and their LDA projection (a, b and c). Example of KNN (distance-based) handling non-linearly separable classes

While LDA cannot separate well non-linearly separable data[7] (see Section 7.1 for more details), KNN does not require specific distribution or linearity and performs the classification well even with non-linearly separable data as long as the classes are distributed in "chunks".

Figure 8-B exemplifies three classes and the resulting LDA projection - class 'A' is marked with 'O' signs, class 'B' is marked with '+'s and class 'C' is marked with 'Δ's.

[7] Note that my AMIR experiments with various non-linear discriminant analysis methods (quadratic discriminant analysis, kernel discriminant analysis, etc.) have not shown any advantages over using LDA+KNN while producing different extra problems depending upon the specific DA method tested.

The LDA projection of the classes to one-dimensional space[8] is labeled with 'a', 'b', and 'c' respectively (these are positioned in location 0 of the Y axis). Notice class A which is clustered in four separate chunks. Although LDA cannot manage linearly clustering A into one chunk, yet because after the linear projection the samples 'a' remain closely-distanced, KNN can still classify this data properly.

Although unlike classification methods which fit a model to the data, KNN has somewhat large memory requirements at classification time as it needs all data to be present (*O(n)*, where *n* is the number of samples), LDA reduces the data dimensionality considerably (down to C-1 dimensions or less, where C is the number of classification classes), practically eliminating memory problems. This LDA dimension reduction is done by computing a transformation matrix using the Learning set (computation required only once for a system with a constant Learning set) and multiplying it by the Learning set and by any future Test sets reducing their dimensions.
For example, in Chapter 12 when classifying Solos of 7 musical instruments using the Full Feature Descriptor Set (see Chapter 6), the number of dimensions of the feature matrix is reduced by LDA from 512 down to 6, reducing storage and memory requirements by 98.83%.

Compared to the Bayesian Gaussian Classifier (see Gibbs 1998) for example, KNN does not require the data to have a normal distribution which the Gaussian Classifier is based upon. Neural networks such as Backpropagation and non-linear Support Vector Machines (SVM) (see Boser, Guyon and Vapnik 1992), while being able to classify rapidly, have extremely slow learning ("training") processes such that are impractical for my "Minus-1" cross-validation evaluation methods which require multiple classification rounds (see Chapter 9).

LDA+KNN is relatively simple, a learning process is not required at all while classification is fast, and the classification, being distance-based, is performed well even with non-linearly separable and non-Normal distributed data. The LDA requirement for having several times more samples than dimensions is met throughout this document except in Chapter 9, where very small databases are used on purpose in order to artificially produce much higher classification results with a Backpropagation network than with LDA+KNN and prove the point of that chapter.
Comparative experimental motivation for LDA+KNN was given in (Livshin, Peeters and Rodet 2003), where an average of 20 Self-Classification experiments using 16 instruments from the IRCAM Studio OnLine database and the Compact Feature set (Section 6.2), produced slightly higher recognition rates with LDA+KNN than with LDA+Gaussian Classifier (the Gaussian classifier without LDA producing much lower results) and Learning Vector Quantization (LVQ) neural network (Kofidis et al. 1996) in all three taxonomies compared – Pizzicato/Sustain, Instrument Families and Musical Instruments (Figure 4-A).

[8] Obviously in this example figure LDA does not project the data unto the X axis. The LDA transformation appears on location 0 of the Y axis only for displaying the data conveniently in the same graph.

CHAPTER 9 DIFFERENT EVALUATION TECHNIQUES AND THE IMPORTANCE OF CROSS-DATABASE EVALUATION

Many papers, (Martin and Kim 1998, Fraser and Fujinaga 1999, Kaminskyj 2001, Agostini et al. 2001, Peeters and Rodet 2002) and others dealing with AMIR have used sounds taken from a single sound database package for evaluation of their proposed AMIR algorithms, using an evaluation method which is referred to here as "Self-Classification".

This chapter first performs an experiment which demonstrates the problems with Self-Classification and then introduces several better evaluation techniques that use data from different sources in the Learning and Test sets.

The experiment demonstrates the following claims:
Evaluation results using a single sound database are not necessarily a good statistic for the ability of a classification algorithm to learn, generalize or classify well. They also do not demonstrate the generalization ability of the trained classifier, after it has learned the evaluation database, to perform in "real-world" applications - classifying sounds recorded in diverse recording conditions. Furthermore, Self-Classification evaluation results do not indicate how well the evaluation database is representative of the possible sound variations of the classified instruments.
A feature selection algorithm might choose different features for classification of the same instrument types, depending on the sound database it is activated on. This also means that an evaluation of features using a single database will not necessarily demonstrate their suitability for universal classification of the instruments.
The "Minus-1-DB" evaluation method is presented. This method uses several databases, each one being classified by the others joined together. It is demonstrated that this method does not have the shortcomings detailed above.

After it is shown that cross-database evaluation is necessary, two other new cross-validation evaluation techniques are presented - “Minus-1-Solo” and “Minus-1-Instance”; both use sounds from independent sources in the Learning and Test sets. These evaluation techniques are used later in Chapter 11, Chapter 12 and Chapter 13.

To see how AMIR evaluation is integrated into the AMIR process, see the AMIR process overview (Section 1.1), shapes #7 - #9.

9.1 INTRODUCTION

An ultimate goal of research dealing with classification of sounds of musical instruments is to create what I call a "Concept Classifier" - such classifier could recognize which instruments are playing regardless of specific recording conditions, a specific performer or a specific instrument. But what are the qualities that distinguish between the sounds of one Concept instrument and the other, e.g., which features really characterize the sound of the Concept Violin and not just a specific violin recording?

Various AMIR papers suggest collections of specific descriptors to be computed on the sound samples. These descriptors are supposed to encapsulate the differences between different musical instruments. Another approach is to compute a large collection of descriptors and then use a feature selection algorithm which attempts to choose the best ones, e.g., (Peeters and Rodet 2002). Various classification algorithms are compared, with the goal of finding the one which can make the best use of the descriptors and reach the highest AMIR recognition rate for the classified musical instruments.
Until recently, right up to the time (Livshin and Rodet 2003) was published and dealt with issues of database-independent evaluation (upon which part of this chapter is based),

the common evaluation method for evaluating instrument-recognition techniques, which is called here "Self-Classification", was to use a relatively large and well known sound database (usually McGill (Opoloko and Wapnick 1991)), choose out of every instrument class a random Learning set of a desired size, e.g., 66% of the samples, and use this Learning set to classify the remaining samples. The number of papers using a single sound database for evaluation of their proposed sound classification processes, as mentioned above, is quite large - (Martin and Kim 1998, Fraser and Fujinaga 1999, Kaminskyj 2001, Agostini et al. 2001, Peeters and Rodet 2002) and others, while using cross validation with samples out of several independent databases (Eronen 2001) was uncommon.

The problem with an evaluation which uses a single sound database as a source for both the Learning and Test sets, is that normally the samples in the Learning set are quite similar to the ones in the Test set - the same recording conditions, same specific musical instruments used for the recording, same performer, etc. Does such an evaluation process really reflect the classification process's ability to recognize the Concept Instruments, or does it just demonstrate the specific qualities of the samples and the self consistency of the evaluation database? How much does this database reflect the variety of possible sounds of the classified instruments and is it fit for training a Concept Classifier?

This chapter shows that evaluation using a single database indeed does not necessarily represent the generalization abilities of the classification process, the selected feature descriptors or the Learning database. An alternative evaluation method is presented, "Minus-1-DB", which offers better evaluation of the generalization capabilities and the suitability of the classification process for "realistic" tasks. As mentioned, "Minus-1-DB" uses several databases for evaluation, where every one of them is classified by the rest joined together. The assumption behind this method is that it is reasonable to expect samples from different databases to be recorded in different conditions.
The intuitive claim that joining several sound databases in a Learning set helps the classifier to deal better with new samples and get nearer to being a Concept Classifier is also demonstrated.

Following the experiment results, besides Minus-1-DB, two other new cross-validation evaluation techniques are presented - “Minus-1-Solo” and “Minus-1-Instance”, which also use sounds from independent sources in the Learning and Test sets.

Finally - is it really possible to reach the theoretical goal of a Concept Classifier, one which could deal with considerable success with the diverse sound possibilities of Concept Instruments? In Section 11.8.1.1 a Minus-1-Instance (cross-database evaluation technique) recognition rate of over 95% for 10 instruments is achieved.

9.2 THE TESTING SET

9.2.1 THE SOUNDS

Excerpts out of 5 sound databases are used in this experiment: Ircam Studio Online ("SOL") (Ballet 1998), University of Iowa Musical Instrument Samples ("IOWA") (Fritz 1997), McGill University Master Samples ("McGill") (Opoloko and Wapnick 1991) and the sound collections Pro and Vi. These databases have been recorded in various acoustic conditions and with different recording equipment.

There are seven instruments common to all these sound databases: bassoon, contrabass, clarinet, French horn, flute, oboe, and cello; these instruments were extracted out of every database for the experiment. The selected sounds were those played with a "regular" playing technique (not pizzicato, martellato, etc).

The number of samples extracted out of each database, is:
SOL (581), IOWA (1289), McGill (85), Pro (158), Vi (249)[9]

All the samples were resampled in mono, 44.1 KHZ sampling rate, 16bit, and clipped to two seconds.

9.2.2 FEATURE DESCRIPTORS

The Compact Feature Descriptor set (see Chapter 6) is used in this chapter. Before each classification round, the feature descriptors of the databases used in this round are normalized together to the range [0 – 1] using the Min-Max normalization method (STA 2001). Throughout this section, when there is a reference to a sample, it means its corresponding vector of feature descriptor values.

[9] As already mentioned, the LDA requirement for having several times more samples than dimensions is met throughout the thesis except here, where very small databases are used on purpose in order to artificially produce much higher classification results with a Backpropagation network than with LDA+KNN and prove one of the points of the chapter.

9.3 CLASSIFICATION ALGORITHMS

Two classification processes are used in the experiments in this chapter:

9.3.1 "LDA+KNN"

See Sections 7.1, 8.2 and 8.3 for full explanation of Linear Discriminant Analysis (LDA), the K-Nearest Neighbor classifier (KNN) and the LDA+KNN combination.

The value of K used with KNN is selected here from the range of [1 - 20] using LOO on the Learning set (Section 8.2.1).

9.3.2 "BP80"

The backpropagation neural network used in this chapter has a single hidden layer of 80 neurons, uses "tansig" functions in all the layers and is trained on the Learning set using the Conjugate Gradient with Powell/Beale Restarts algorithm until a Mean Square Error of 0.004 is reached[10]. The trained network is then activated on the Test set; this classification process is called "BP80" throughout this section.
To read about Neural-networks see Section 8.1.

Note that an advantage of BP over LDA+KNN is that it can fit models to comparatively small collections of samples while LDA needs the number of samples to be larger than the number of descriptors to operate well.

9.4 EVALUATION METHODS

This section presents the popular Self-Classification evaluation method, Mutual-Classification and Minus-1-DB, which are to be used in the following experiment for "denouncing" the Self-Classification method. Following the conclusions of the experiment, two more evaluation methods are presented later.

9.4.1 SELF-CLASSIFICATION EVALUATION METHOD

In this common evaluation method, a single database of sound samples is split into a Learning set and a Test set, thus the name "Self-Classification". X% of the samples of each class are randomly selected for the Learning set and the rest become the Test set. In

[10] Class information of each sample is represented in the learning set by a vector of 7 numbers (7 is the number of classes), with '1' in the index of the correct class and '0's in the rest. The mean square error is computed using these vectors and the neural network outputs.

order to eliminate the dependency of the resulting recognition rate on a specific random split into Learning and Test sets, this process is repeated N times and the average (and possibly the standard deviation and confidence interval) of the recognition rates are reported.

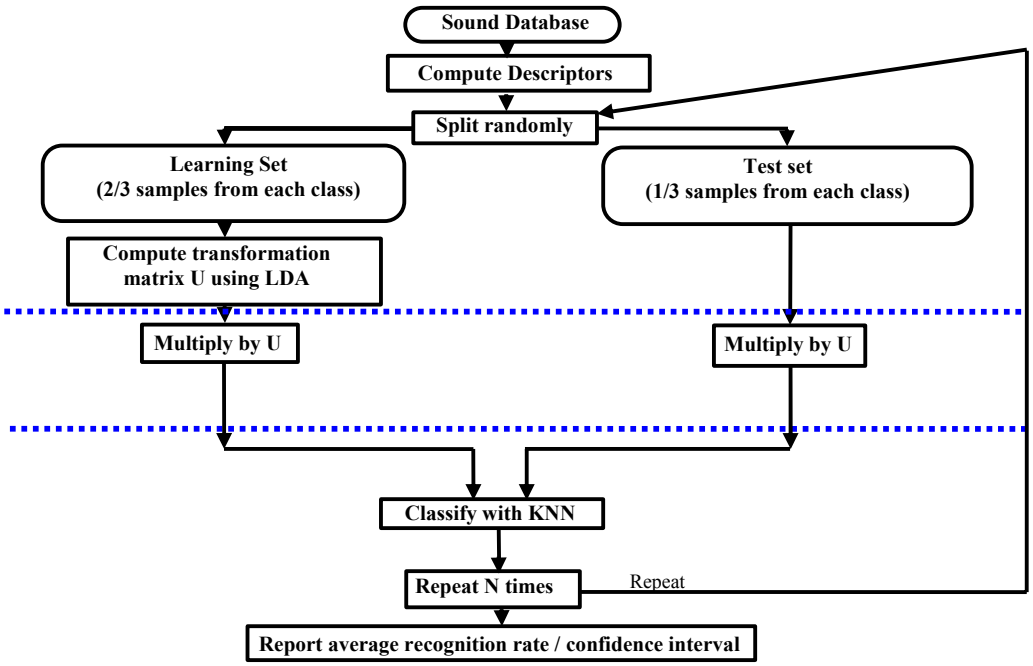

Figure 9-A. "Self-Classification" Evaluation process using LDA+KNN

In this chapter, in experiments where Self-Classification is used, 66% of the samples of each class are randomly selected for the Learning set and 33% for the Test set. Each reported result is the mean of 50 classification rounds with randomly selected sets.

9.4.2 MUTUAL-CLASSIFICATION EVALUATION METHOD

In this proposed classification method a single complete sound database is used as the Learning set, classifying a different single complete database - the Test set. The method is repeated for each pair of sound databases. The average recognition rate is reported along with individual results of all database pairs.

9.4.3 MINUS-1-DB[11] EVALUATION METHOD

This proposed algorithm is an alternative evaluation method to Self-Classification. The Minus-1-DB evaluation method uses several sound databases recorded in different recording conditions, classifying each one by the rest joined together.
If some of the instruments appear only in a single database, they should be removed, as obviously these samples could not be classified correctly. In the experiment in this chapter there are no such samples, because as already mentioned, the excerpts selected out of the five databases contain the same seven instruments.

The advantages of this method over Self-Classification are demonstrated in the following sections.

9.5 DISADVANTAGES OF SELF-CLASSIFICATION

In this section several **points** are demonstrated:

1. Evaluation results using a single database are not necessarily an indication of the generalization abilities of the classification algorithm being used, and its suitability for practical applications of AMIR (as already mentioned, the ultimate goal is to have a Concept Classifier).

2. Self-Classification results do not reflect the classifier ability, after learning this specific database, to deal with new sounds, and thus its performance as a Concept Classifier.

3. Evaluation with Self-Classification of an AMIR process using specific feature descriptors does not necessarily reflect the suitability of these feature descriptors for classification of the tested instruments in a Concept Classifier.

4. The intuitive claim that enriching the Learning database with diverse samples from other databases improves the generalization power of the classifier and makes it more suitable for classification of new sounds, is also demonstrated.

The diagonal in Table 9-A (the numbers in parenthesis) shows the average recognition rate of 50 Self-Classification rounds for each sound database, using a Learning set consisting of 2/3 of its samples.
The “Minus-1-DB” column shows the recognition rates obtained by classifying a database by all the other databases joined together.

[11] The name "Minus-1" was taken from Jazz training records with the same name. These records contain recordings of "Standards" where the whole band plays together excluding one instrument (minus one). The part of the missing instrument should be played by the practicing musician (resembling Kareoke).

The rest of the table shows the Mutual-Classification recognition rates of classifying each of the sound databases using every other sound database as the Learning set.

The classification process used in Table 9-A is LDA+KNN.

Separate-Tone Database	SOL	IOWA	McGill	Pro	Vi	Minus-1 DB
SOL (581) classified by	(98.24)	39.93	20.14	21.51	58.17	68.5
IOWA (1289) classified by	51.43	(97.75)	35.22	29.17	58.42	65.79
McGill (85) classified by	51.76	51.76	(60.78)	23.53	48.23	77.65
Pro (158) classified by	54.43	41.77	26.58	(48.04)	58.86	75.32
Vi (249) classified by	63.45	48.59	30.12	20.88	(64.42)	75.9

Table 9-A. Self-Classification, Mutual Classification and Minus-1-DB results using LDA+KNN

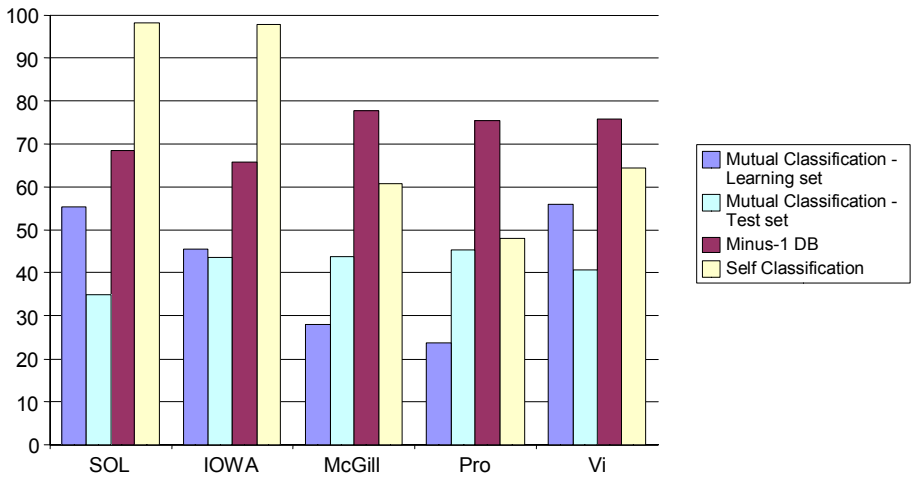

Figure 9-B. Different evaluation grades per sound database using LDA+KNN

The graph in Figure 9-B shows for each database the Minus-1-DB result, self-classification result, the average recognition rate when the database is used as the Learning set for classifying all other databases, and average recognition rate when the database is used as the Test set, being classified by every other database separately.

Some text below is colored in order to point out to which bar in the graph it refers to.

By examining Table 9-A and the SELF CLASSIFICATION *vs.* LEARNING GROUP bars in the graph we see that the results of Self-Classification of a database are not consistent with the results when other databases are being classified by it. This shows that Self-Classification results do not predict how a classifier which is trained on one database will classify new samples. **Point #2 demonstrated**.

We can see that the MINUS-1 DB results are always higher than when classifying a database by a single other database SEPARATELY. Thus it is demonstrated that

enriching the Learning database by samples out of other databases helps the classifier to generalize better and get closer to recognizing the Concept Instruments. Even relatively small databases, which do not even contain enough samples for good Self-Classification using LDA+KNN (McGill, Pro and Vi), when they are added to the Learning set, considerably improve the generalization ability of the classifier. For example, even when SOL (a relatively large database) is classified by IOWA (the largest one), the results are still considerably improved, from 39.93% to 68.5%, after the three small databases are added to IOWA (Minus-1-DB classification of SOL). **Point #4 demonstrated**.

Note that by comparing Self-Classification and Mutual Classification results, it is possible to evaluate for each database its SELF CONTAINMENT *vs.* its DIVERSITY, thus concluding how well the database is suited for generalized classification (which is the important thing for a Concept Classifier). For example, when examining Table 9-A, we can see that Vi, while not appearing to be very self contained, seems to be diverse enough and comparatively suited for classification of the other databases.

Instruments classification	**Self Classification**	**Minus-1 DB**
SOL	(97.93)	87.78
IOWA	(99.35)	74.71
McGill	(77.86)	80
Pro	(87.55)	84.18
Vi	(92.84)	89.16

Table 9-B. Self-Classification and Minus-1-DB results using BP80

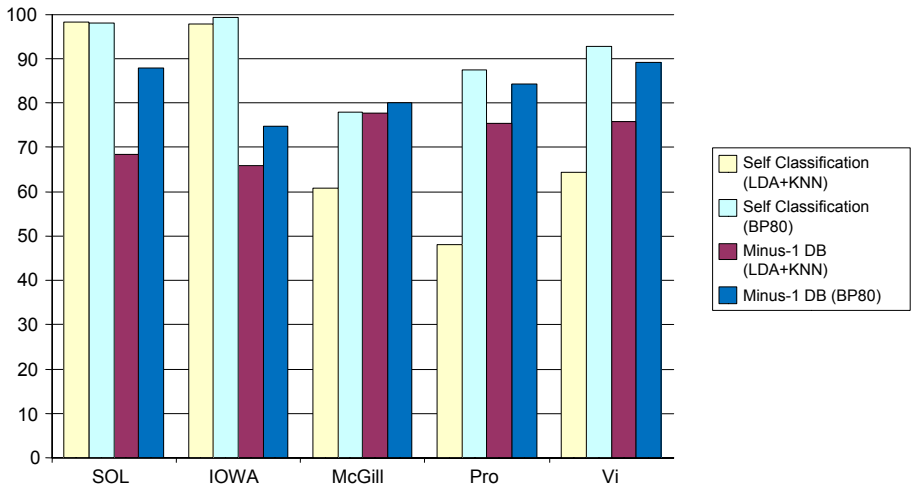

Figure 9-C. Self-Classification and Minus-1-DB evaluations of LDA+KNN vs. BP80 classification algorithms

Examining the graph in Figure 9-C and comparing the MINUS-1 DB results in Table 9-A and Table 9-B, we see that the neural network in this experiment performs much better than LDA+KNN. Yet, if we compare the SELF CLASSIFICATION results of the SOL and IOWA databases in Table 9-A and Table 9-B, we see that the results are very similar.
Following from that, if we would compare LDA+KNN and BP80 just by using Self-Classification of a single large database (like already mentioned, this is a common practice in many articles), we could conclude that there is no considerable difference in the capabilities of LDA+KNN and BP80 and that both perform very well. **Point #1 demonstrated**.

Remark: it is interesting to see in Table 9-A and Table 9-B that although the three smaller databases are too small to reach good Self-Classification results using LDA+KNN (LDA requires having more samples than features), they do contain enough class separation information for BP80 to perform Self-Classification with much higher success.

Demonstrating Point #3:
Using the GDE algorithm (see Section 7.2), the apparently best eight feature descriptors were chosen for each sound database (out of the total 162 descriptors).
Table 9-C shows which features were selected using each database. The upper row contains the indices of all the feature descriptors that were chosen. The Xs indicate the selected features.

Desc #	18	19	42	44	45	46	47	48	49	50	51	52	73	134	135	136	137	138	140
SOL	X	X						X					X	X		X	X	X	
IOWA	X			X	X	X	X							X				X	X
McGill			X			X	X				X	X				X	X	X	
Pro	X					X	X		X					X		X	X	X	
Vi	X					X	X			X				X	X	X	X		
Total	4	1	1	1	1	4	4	1	1	1	1	1	1	4	1	4	4	4	1
All DBs Merged				X		X	X	X						X		X	X	X	

Table 9-C. Eight best feature descriptors provided by GDE

The table shows that different "best" features are selected for each database, thus evaluation of features using a single database does not necessarily demonstrate the usefulness of these features for a Concept Classifier. **Point #3 demonstrated.**

Note for the curious – the most popularly selected feature descriptors:
Seven descriptor indices figure four times each - Sharpness, Specific Loudness-19, Specific Loudness-20, Spectral Centroid, Spectral Skewness, Spectral Kurtosis and Spectral Slope. Out of these, six were selected (except Sharpness) when the merged databases were used.

9.6 CONCLUSIONS

In the above experiment the common "Self-Classification" evaluation method, which uses a single sound database for evaluation of AMIR, was criticized.

The following claims were shown:

1. Evaluation results using a single sound database are not necessarily an indication of the generalization capabilities of the classification process and thus its suitability for realistic classification tasks, where the ultimate goal is to have a Concept Classifier - a classifier which could classify instrument sounds regardless of their specific recording conditions.

2. Self-Classification results do not demonstrate the ability of a classifier which was trained on a single database to classify new sounds, and thus its performance as a Concept Classifier. This also means that Self-Classification results do not demonstrate the diversity or self containment of the sound database being used.

3. A feature selection algorithm might choose different features for classification of the same musical instrument types, depending on the sound database being used. This also means that evaluating features using a single database will not necessarily help to choose the right features for a Concept Classifier.

4. By enriching the Learning database with diverse sound samples from other databases, we help the classifier to generalize better and make it more suitable for classification of new sounds.

To deal with the shortcomings of Self-Classification, the "Minus-1-DB" evaluation method was introduced. With Minus-1-DB, the evaluation is performed using several sound databases, each classified by the rest joined together. Minus-1-DB results do provide an indication as to the generalization abilities of the evaluated classification algorithm and feature descriptors, as the classification algorithm never learned the classified database and is not adapted to the specific features of its samples (as happens in Self-Classification). Although the generalization ability still depends on the sound databases being used (because getting all the sounds in the world is difficult), it could be reasonably assumed that the recording conditions of the samples in the various databases are different, which allows evaluation of the generalization abilities of the AMIR process concerning at least these databases, while Self-Classification does not practically evaluate generalization at all.

9.7 MORE EVALUATION ALGORITHMS

9.7.1 MINUS-1 INSTRUMENT INSTANCE EVALUATION METHOD

This proposed AMIR evaluation method is an alternative to Minus-1-DB.
At each classification round, a single instrument is removed from a sound database and then classified by all the databases joined together, including its own. This process is repeated until all instrument instances are classified. The average recognition rate per instrument is reported.

9.7.1.1 Comparison with Minus-1-DB

Both these methods are intended for databases of separate notes. Both do not use samples from the same source in the Learning and Test sets.
An advantage of Minus-1 Instrument Instance is that an average 'Instrument Grade' is calculated, which is much more informative than an average sound-database grade as the arbitrary division into databases and the instruments present or missing from certain databases do not affect it.
On the other hand, Minus-1 Instrument Instance is much more time consuming as each instrument Instance requires a whole classification learning/classifying cycle while in Minus-1-DB several Instances of different instruments may be integrated into a single database and classified together in one classification round.

9.7.2 MINUS-1-SOLO EVALUATION METHOD

This is a proposed method for evaluation of AMIR in Solos.

Algorithm:
- All Solos are divided into chunks – either notes or other Solo segments
- For every Solo:
  - Remove its chunks from the Solo database
  - Classify them by the rest of the Solos in the database together
  - Compute an average recognition rate for the chunks
- Report average recognition rate of Solos for each musical instrument

This method is more informative than computing the average recognition rate per Solo or per chunk, as the number of Solos by each instrument and their lengths could be different. Like the proposed methods above, this is a cross-validation method which uses truly independent data in the Learning and Test sets.

9.7.3 LEAVE-ONE-OUT CROSS VALIDATION METHOD

This is a comparatively old evaluation method. In order to avoid the possible bias introduced in Self-Classification Evaluation by relying on any random division of the evaluation database into Test and Learning sets, in this method, each sample of the evaluation database is classified by the rest joined together; finally, the average recognition rate per sample is reported.

This method is called leave-one-out (LOO) cross-validation (Kohavi 1995), because one sample is always left out of the Learning set.

LOO could be considered the father-method for most of my suggested evaluation methods in this chapter. If at each classification round, instead of removing one sample and classifying it by the rest, a whole sound database is removed, the resulting algorithm is 'Minus-1-DB'. If at each round an Instrument Instance is removed, the result is Minus-1-Instance. Removing one Solo at each stage produces Minus-1-Solo.

CHAPTER 10 IMPROVING THE CONSISTENCY OF SOUND DATABASES

In AMIR, like in other machine learning fields, the Learning (classifying) database might contain samples which could disturb the classification process:

- **Attribute Noise**
  Badly sampled sounds or garbled data

- **Class Noise**
  Samples mislabeled as belonging to the wrong class

- **Distance Outliers**
  Samples correctly recorded and labeled but still mislead the classification process by differing too much from other samples in their class (see 10.5 for discussion)

The presence of such samples can lead to inflated error rates and substantial distortions of parameter and statistic estimates when using either parametric or nonparametric tests (Zimmerman 1998).

For a thorough historical summary on Outliers see (Jason and Overbay 2004).

This chapter presents two new algorithms for removing outliers and compares them with the common IQR method. To understand where database purging is integrated into the complete AMIR process, see the AMIR process overview (Section 1.1), shape #5.

10.1 ALGORITHMS FOR REMOVING OUTLIERS

10.1.1 INTERQUANTILE RANGE (IQR)

IQR (Draper 1999) is a commonly used outlier removal approach:

For every descriptor, let **P1** be some value bigger than X% of the values of this descriptor, and let **P2** be a value bigger than Y% of the values, X>Y. For example: X=99, Y=1.

Remove samples where the descriptor has values that are larger than

P1+(P1-P2)*C

or smaller than

P2-(P1-P2)*C

where C is some scalar (e.g., 1).

Instead of using percentages, a common modification of IQR[12] is to calculate the mean and standard deviation (STD) of every descriptor, and then remove the samples where that descriptor has distances which are several times bigger than the STD.

Notice that IQR is not a supervised method, meaning that it does not utilize classification information of the Learning database, and thus is appropriate for usage with non-labeled databases.

10.1.2 MODIFIED IQR (MIQR)

This is a new, proposed supervised variant of Interquantile Range.

- Modification 1: Perform IQR on each class separately instead of all the database samples together.
- Modification 2: When a sample with an outlier descriptor is found, do not remove it automatically, but rather count for every sample descriptor-vector its number of outlying descriptors. At the end of the process remove the samples which have more outlying descriptors than a specified threshold.

[12] See for example the subsection "Removing Outliers" in section "Data Analysis" in the Matlab R2006b documentation.

10.1.3 SELF-CLASSIFICATION OUTLIER REMOVAL (SCO)

This proposed outlier removal method is a kind of a "wrapper method" in the sense that it utilizes the specific classification algorithm itself for its purpose.
Like in Self-Classification evaluation (see Section 9.4.1), the Learning set consists of a certain percentage of the samples from each class (66% in the experiment in this chapter) which are selected randomly, while the Test set is made of the rest (reminder - the Learning set is used to classify the Test set). The classification process repeats N number of times (50 times here). After each classification round the indices of the misclassified samples are recorded.

At the end of the process, samples which were misclassified more than a certain number of times are removed.

Note that this method differs significantly from my older LOO outlier removal method presented in (Livshin, Peeters and Rodet 2003) which has classified each sample in the database using Leave-One-Out and removed the misclassified ones. SCO uses partial, randomly selected groups of samples (66%/34%) at each classification step, creating a kind of "bagging" effect and thus lowering the distortion in classifications caused by outliers in the Learning set (François et al. 2003), which the older LOO method suffers from.

10.2 CONTAMINATED DATABASE

The current evaluation was performed using an excerpt of the extensive IRCAM Studio OnLine (SOL) separate tone database (see Section 5.1). This excerpt contains 1325 sound samples of 20 musical "instruments" - guitar, harp, violin (pizzicato and sustained), viola (pizzicato and sustained), cello (pizzicato and sustained), contrabass (pizzicato and sustained), flute, clarinet, oboe, bassoon, alto sax, accordion, trumpet, trombone, French horn and tuba. All the samples are two seconds long, monophonic and sampled in 44.1 KHz with 16 bit resolution.
The Compact Feature Descriptor set, consisting of 162 Feature Descriptors (See Chapter 6) is computed on each sample.

As the SOL database was professionally recorded, it is very self-consistent as is evident from its average self-classification grade - 95.7% for 50 Self-Classification experiments of 66% / 34% split.

In order to compare the effectiveness of the three different outlier removal methods, the SOL database was "contaminated" with four kinds of outlying samples:

- **"Class Noise"**: The class labels of random 5% of the database samples was changed to a different, randomly selected, class.

- **"Random256 Samples"**: samples with descriptors selected randomly from the range of [0 256] for each descriptor were added to the database with random classes. The quantity of these samples is 5% of the original database size.
- **"Random Bound Samples"**: the minimum and maximum of each descriptor in the original database were found. 5% of random samples were added to the database, each random descriptor in these samples is bound by its respective minimum and maximum values.
- **"Random Samples Class Bound"**: 5% of random samples were added to each class, with descriptors bound by their respective minimum and maximum values in this class.

10.3 EXPERIMENT

Each of the outlier removing algorithms was performed on the contaminated database. As there is a tradeoff between the number of good and bad samples removed by the algorithms, each algorithm was evaluated twice; first allowing up to 1% of "good" samples to be removed (Table 6.1) and second time with up to 10% good samples removed (Table 6.2).

10.4 RESULTS

Reading Table 10-A and Table 10-B:
All the results are in percentages.

The columns

- "Clean": this column shows the average Self-Classification result of classifying only the good samples in the contaminated database, i.e., the contaminated database with all bad samples removed. Note that this "Clean" database is 5% smaller than the original SOL database because of the removal of the distorted Class Noise samples.
- "Contaminated": the average result with the contaminated database.
- IQR, MIQR, SCO – the classification results of the contaminated database after it was purged with each of these algorithms.

The rows

- "Grade" – the average grade of 50 66%/34% self-classification rounds. Numbers in parenthesis are the 95% confidence intervals.
- "Class noise", "Random256, "Random Bound", "Random Bound Class" – the percentage of each type of bad samples removed by the algorithm. For example, in Table 10-A, MIQR has removed 53% of the Class Noise.

- "Bad Removed" – the total percentage of bad samples removed.
- "Good Removed" - the total percentage of good samples removed.

	Clean	Contaminated	IQR	MIQR	SCO
Grade	92.7 **(92.0– 93.4)**	79 **(78.6 – 79.5)**	88.1 **(87.7 – 88.4)**	91.6 **(91.3 – 91.8)**	86.2 **(85.8 – 86.6)**
Class Noise	NA	0	0	53	51.5
Random256	NA	0	100	100	39
Random Bound	NA	0	81.8	100	43.9
Random Bound Class	NA	0	12.5	31.8	7.6
Bad Removed	NA	0	49.6	70.1	35.5
Good Removed	NA	0	0.9	0.9	0.95

Table 10-A. Outlier removal results with up to 1% of good samples removed

	Clean	Contaminated	IQR	MIQR	SCO
Grade	92.7 **(92.0– 93.4)**	79.2 **(78.6 – 79.5)**	89.8 **(89.5 – 90.2)**	92.3 **(91.5 – 93.1)**	96.8 **(96.4 – 97.1)**
Class Noise	NA	0	18.2	75.7	100
Random256	NA	0	100	100	100
Random Bound	NA	0	100	100	98.5
Random Bound Class	NA	0	50	86.4	51.6
Bad Removed	NA	0	67.2	90.4	87.8
Good Removed	NA	0	9.9	8.8	9.5

Table 10-B. Outlier removal results with up to 10% of good samples removed

Looking at the types of bad samples removed by each algorithm we can see:

IQR - As could be expected from its non-supervised nature, IQR has dealt badly with Class Noise, being unable to detect it. Random256 and Random Bound outliers were removed well as the probability of getting at least a single descriptor out of 162 with an "edge" value is high with these contamination types, and a single outlying descriptor is enough for IQR to remove a sample. Samples from the Random Bound Class are much more difficult for IQR to detect – many descriptors in many classes do not have edge values compared to the Min/Max values of these descriptors over the entire database. For example in class X, the minimum and maximum values of descriptor Y could be [-10, 10], while the minimum and maximum values of descriptor Y over the entire database are [-50, 50]. And so, Random Bound Class samples from class X will never have an outlying descriptor Y. As IQR does not use class information, it cannot detect such descriptors even if they do have a "local" edge value in their class.

MIQR - We can see that the MIQR method has outperformed the other two, removing higher percentages of bad samples. As it uses class information, it did not have the disadvantages of IQR regarding Class Noise and Random Bound Class samples. Another reason for its higher "data-to-noise" ratios is that it did not remove every sample with a single outlying descriptor, but rather removed samples which had at least N outlying descriptors. Naturally, in systems where a single sensor may go wrong and produce sometimes random values, N could be simply set to 1.

SCO – As the SCO algorithm does not attempt guessing whether samples should be removed by examining their values, its behavior is the same with all types of outliers as long as they are misclassified; however, as the Random Bound Class samples had the highest probability of being actually classified as their appointed class (while possibly having outlying values which could be detected by MIQR) SCO had the least success removing them. Random Bound Class samples were the toughest samples to handle for all the outlier removing algorithms.
In Table 10-B we see that SCO has produced the purged database with the highest mean self-classification grade – 96.8%, which is even noticeably higher than the grade the Non-Contaminated database produced – 92.7%. This high self-classification grade was achieved while removing only 87.8% of the contaminated samples vs. 9.5% of good samples (worse than MIQR). This is actually not surprising – allowing the SCO algorithm to remove as much as 10% of the good samples, we let it tailor the remaining database for itself, SCO being a wrapper method; nevertheless, this does not mean that SCO outperformed the other algorithms in this case. Our goal was not to get the highest self-classification grade but rather to get rid of the most "bad" samples for the price of a certain percentage of good samples removed as well. See the next section, "Conclusions", for some more discussion of this topic.

10.5 CONCLUSIONS

For non-labeled data, out of the three tested algorithms, IQR is the "only way to go" as the other two algorithms require class information. For getting rid of contaminated data in labeled databases, MIQR seems to be the best. If maximally high classification results are required, specifically tailored wrapper-type methods may well be the answer, such as SCO.

Note that not all outliers should be always removed – there are many arguments about the desirability of the whole business of removing outliers, as diversity in a database is not necessarily bad and may actually model a special, interesting, population rather than indicate sampling errors. The general rule is to "know your data" and being able to "intelligently guess" which percentage of erroneous samples could be expected thus providing the outlier removing algorithms with appropriate limiting parameters, such as the percentage of samples to remove, the number of descriptors which are likely to go

wrong, or even tailor special outlier removing algorithms for specific data types such as the one in (Adam, Rivlin and Shimshoni 2001) for removing outliers from different views (graphical images) of the same scenery.

CHAPTER 11 AMIR OF SEPARATE TONES AND THE SIGNIFICANCE OF NON-HARMONIC □NOISE□ VS. THE HARMONIC SERIES

Sound produced by Musical instruments with definite pitch consists of the Harmonic Series and the non-harmonic Residual. It is common to treat the Harmonic Series as the main characteristic of the timbre of pitched musical instruments. But does the Harmonic Series indeed contain the complete information required for discriminating among different musical instruments? Could the non-harmonic Residual, the "noise", be used all by itself for instrument recognition?

This chapter begins by performing musical instrument recognition with an extensive sound collection using a large set of feature descriptors, achieving a high instrument recognition rate. For history of AMIR performed on isolated tones see Section 3.1.

Next, using Additive Analysis/Synthesis, each sound sample is resynthesized using solely its Harmonic Series. These "Harmonic" samples are then subtracted from the original samples to retrieve the non-harmonic Residuals. Instrument recognition is performed on the resynthesized and the "Residual" sound sets. The chapter shows that the Harmonic Series by itself is indeed enough for achieving a high instrument recognition rate; however, the non-harmonic Residuals by themselves can also be used for distinguishing among musical instruments, although with lesser success. Using feature selection, the best 10 feature descriptors for instrument recognition out of our extensive feature set are presented for the Original, Harmonic and Residual sound sets.

11.1 INTRODUCTION

Musical instruments with definite pitch ("pitched instruments") are usually based on a periodic oscillator such as a string or a column of air with non-linear excitation. In consequence, their sound is mostly composed of a Harmonic Series of sinusoidal partials, i.e., frequencies which are integer multiples of the fundamental frequency (*f0*); see Figure 11-A for several examples of harmonic series of different instruments.

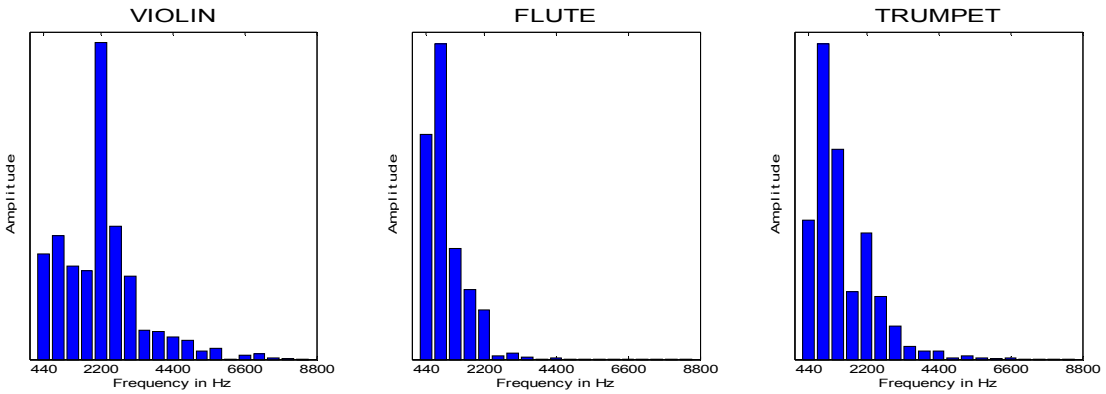

Figure 11-A. First 20 harmonics of an A4 note played by different instruments; DFT analysis window is 200ms long and taken from the sustained part of the signal

While the relation between the energy levels of the different harmonics is widely considered as the main characteristic of pitched instruments' timbre, (e.g., (Eggink and Brown 2004), (Kitahara et al. 2007)), if we subtract this Harmonic Series from the original sound there is a non-harmonic Residual left. This Residual is far from being 'white noise'; it is heavily filtered by the nature of the instrument itself as well as the playing technique, and may contain inharmonic sinusoidal partials as well as non-

sinusoidal 'noise', such as the breathing sounds in the flute or the scraping noises in the guitar.
Does the Harmonic Series indeed encapsulate all the distinguishing information of the sounds of pitched musical instruments? If so, about the same instrument recognition rates should be achieved by using only the Harmonic Series as by using all the information in the signal, with the same feature descriptor set used for classification. This is a practical question for the field of instrument recognition; when performing instrument recognition in MIP music, it is difficult as well as computationally expensive to perform full source separation (Vincent and Rodet 2004) and restore the original sounds out of the polyphonic mixture in order to recognize each source separately[13]. On the other hand, estimating the Harmonic Series of the different notes in the mixture is a relatively easier task (Yeh, Röbel and Rodet 2005). For example, in (Livshin and Rodet 2004b) Harmonic Series estimation is used for performing "Source-Reduction", reducing the volume of all instruments except one and then recognizing it. In (Eggink and Brown 2004), instrument recognition is performed using only features based on the Harmonic Series, estimated using a-priori *f0* information.

Another interesting question comes from the opposite direction: is the non-harmonic Residual, the "noise" a musical instrument produces, so distinct as to allow distinguishing between different instrument types, e.g., can we actually distinguish between different wind instruments just by the sound of their airflow hiss?

In order to answer these questions, this chapter explores how instrument recognition rates using signals resynthesized solely from the Harmonic Series of the sound, and signals containing solely the non-harmonic Residuals, compare with the recognition rates when using the complete signals. In order to perform this comparison as directly as possible, the first step is to achieve a high instrument recognition rate. This is accomplished here by computing an extensive set of feature descriptors on a large and diverse set of pitched musical instrument sound samples, reducing the feature dimensions with Linear Discriminant Analysis (LDA) and then classifying the sounds with K-nearest neighbours (KNN).

Next, the Harmonic Series of each sample in the sound set is estimated, including the *f0*s, harmonic partials and corresponding energy levels, and using Additive Synthesis all the signals are resynthesized using only their Harmonic Series, thus creating synthesized 'images' of the original signals which lack any non-harmonic information; these resynthesized sounds are referred to in the chapter as "Harmonic" signals, while the original sounds from the sound set are called, the "Original" signals.

[13] There is also research attempting to perform instrument recognition by recognizing directly instrument mixtures instead of trying to separate them into individual instruments, see for example (Essid, Richard and David 2006).

As the phase information of the Original signals is kept in the Harmonic signals, by subtracting the Harmonic signals from the Original signals we remain with the non-harmonic, "noisy", part of the signals, referred to shortly as the "Residuals".

After that, the same set of feature descriptors is computed on each sample group: the Original, Harmonic and Residual Signals. These three groups are then divided separately into training and Test sets and instrument recognition is performed on each group independently. The instrument recognition results are presented and compared in Section 11.8.

Using the Correlation-based Feature Selection (CFS) algorithm with a greedy stepwise forward search method, the 10 most important feature descriptors for each of the three groups of samples are estimated and presented.

11.2 ORIGINAL SOUND SET

This sound set consists of 3223 samples of single notes of 10 "musical instruments": bassoon, clarinet, flute, trombone, trumpet, contrabass, contrabass pizzicato, violin, violin pizzicato and piano. As the violin and bass pizzicato sounds are very different from the bowed sounds they are treated here as separate instruments.

The sound samples were collected from 12 different commercial and research sound databases (see Section 5.1). The databases contain sounds recorded in different recording environments, using different individual instruments (e.g., using different violins in each sound database). The sound set spans the entire pitch range of each of the 10 instrument types and includes vibrato and non-vibrato sounds where applicable.

The collection of all the samples of a specific instrument taken from a single database (e.g., all the violin samples from database #1), is referred to here as an "instrument Instance". The total number of instrument Instances in the sound set is 67.

Preprocessing:

All sounds are sampled in 44 KHz, 16 bit, mono.

11.3 NOISE REMOVAL

Some of the noise in the Original sound signals falls inside the harmonic grid of the tones. This causes the Additive analysis/synthesis program to attempt modeling such noise with sinusoids, along with the true harmonic-series of the notes. Therefore, in order to get purely "harmonic-series notes" it is important to get rid of the noise prior to performing Additive analysis/synthesis.

The noise removal is done here with an algorithm which performs classification of sinusoidal and noise peaks in the signal and then removes the noise. The classification is based on descriptors derived from properties related to time-frequency distributions: mean time, duration, instantaneous frequency and normalized bandwidth. The descriptors are designed to properly deal with non-stationary sinusoids, which enables this algorithm to produce superior classification results compared to older methods, such as the standard correlation based approach. A 10% error threshold was selected for peaks classification. For full details see (Zivanovic, Röbel and Rodet 2004).

11.4 HARMONIC SOUNDS AND RESIDUALS

Additive analysis/synthesis is based on Fourier's theorem, which states that any physical function that varies periodically with time with a frequency *f* can be expressed as a superposition of sinusoidal components of frequencies: *f*, *2f*, *3f*, *4f,* etc. Additive synthesis applies this theorem to the synthesis of sound (Risset 1985). For a review of supplementary Additive Synthesis techniques see (Serra and Smith 1990).

In order to separate the sound samples into their harmonic and non-harmonic components, after noise removal, the samples are analyzed and then selectively resynthesized using the Additive analysis/synthesis program - "Additive" (Rodet 1997; Röbel 2006), which considers also inharmonic deviations of the partials (e.g., found in the piano sounds). Very precise Additive analysis was performed by supplying the Additive program with specifically tailored parameters for each sound sample using its note name and octave, known in advance, for estimating its *f0*. For example, the Additive analysis/synthesis window size was set to 4*(1/*f0*), FFT size to 4*nextpow2[14](sampleRate * windowSize), etc.

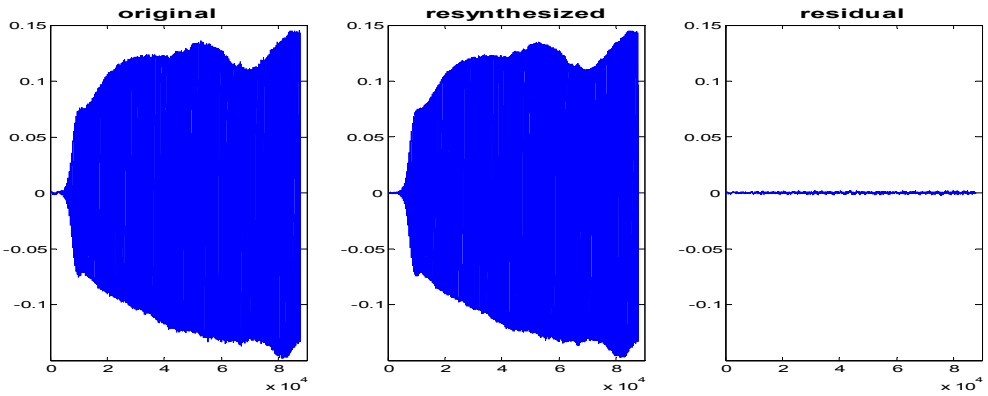

Figure 11-B. Left to right: original Clarinet sample (A3), the sample resynthesized from the Harmonic Series, the Residual (subtraction).

[14] nextpow2(N) is the first P such that 2^P >= abs(N)

Figure 11-B shows an example of an Original clarinet sample (the note A3), the sound resynthesized from the Harmonic Series of the Original, and the non-harmonic Residual. We can see that the Original and Harmonic sound envelopes are similar and that the Residual energy, resulting from subtracting the resynthesized Harmonic sound from the Original, is comparatively very low.

While the sounds resynthesized from the Harmonic Series sound very similar to the Original samples, the non-harmonic Residuals sound very differently from them while sounding quite similar to each other for the same instrument. For example, the clarinet Residual of the note A3 sounds like a steady airflow while the trombone Residual of the same pitch sounds mellower and with addition of "static-electricity" crackle. The bass pizzicato Residual of A3 sounds like a wooden barrel being hit with a hammer, while the Residual of the violin pizzicato of exactly the same note sounds much higher "pitched" due to the considerably smaller size of its wooden resonator, and includes some tremolo. To learn how the physical structure of musical instruments shapes the sound, see (Fletcher and Rossing 1998). Note that the Attacks are far from being the only parts of the Residuals influencing the descriptors; The Sustained parts of the Residuals of the clarinet, flute, trombone and trumpet contain energy levels as high as or higher than their Attack Transients.

11.5 FEATURE DESCRIPTORS

The Full Feature Descriptor set with 513 descriptors is computed on the Original samples, the Harmonic samples and the Residuals. See Chapter 6 for full details.

11.6 FEATURE SELECTION

In order to provide the 10 best features out of the Full Feature Descriptor set for each group of samples (the Original samples, the Harmonic samples and the Residuals) the Correlation-based Feature Selection (CFS) evaluator is used with a greedy stepwise forward search method (see Section 7.3). CFS was chosen here over GDE (Section 7.2) as the purpose is to present a list of non-dependent descriptors as intuitively meaningful as possible, showing the different distinguishing features of the different sets, rather than to find an overly specific set which produces the highest recognition rate with LDA+KNN and the current data (producing around 1% higher recognition rates than CFS).

11.7 CLASSIFICATION AND EVALUATION

AMIR is performed on the Original, Resynthesized and Residual sets of samples separately. In order to get meaningful instrument recognition results it is necessary not to use sounds recorded by the same instrument and the same recording conditions both in the Learning and Test sets (Livshin and Rodet 2003). For this purpose, 'Minus-1 Instrument Instance' cross-validation evaluation method is used (see Section 9.7.1).

Each classification phase of the Minus-1-Instance Evaluation is performed using the LDA+KNN method (See Section 8.3). K values in the range of [1 - 80] are tested at each classification phase. After the Minus-1-Instance evaluation process completes and all the Instances are classified, the best K for the whole classification process is reported.

11.8 RESULTS

11.8.1 INSTRUMENT RECOGNITION

The confusion matrices in this section show the Minus-1-Instance recognition rates for the Original samples, the Harmonic samples and the Residuals. These matrices show the percentage[15] of samples (rounded to integers) of the instruments in the first column which were classified as the instruments in the first row. For example in Table 11-A, 6% of the clarinet samples are misclassified as flute. The instrument abbreviations are: bsn = Bassoon, cl = Clarinet, fl = Flute, tbn = Trombone, tr = Trumpet, cb = Contrabass, cbp = Contrabass Pizzicato, vl = Violin, vlp = Violin Pizzicato, pno = Piano.

11.8.1.1 Original Samples

	bsn	Cl	fl	tbn	tr	cb	cbp	vl	vlp	pno
bsn	**96**	2	1	1	0	0	0	1	0	0
cl	0	**92**	6	0	1	1	0	1	0	0
fl	0	3	**96**	0	1	0	0	1	0	0
tbn	2	0	0	**93**	5	0	0	0	0	0
tr	0	1	1	3	**96**	0	0	0	0	0
cb	0	0	0	0	0	**99**	0	1	0	0
cbp	0	0	0	0	0	1	**96**	0	0	3
vl	1	0	2	0	0	0	0	**96**	0	0
vlp	0	0	0	0	0	0	0	0	**99**	0
pno	2	0	0	0	0	2	0	0	2	**94**

Table 11-A. Confusion matrix of the Original samples

[15] Due to rounding, the total percentage in each row/column does not always add up to 100%.

The average Minus-1-Instance recognition rate per instrument for the Original samples is 95.58% (using K=8 with KNN). It is rather hard to compare recognition rates with other papers as each paper attempts to recognize its own instrument set, uses different sound databases and different evaluation techniques. In addition, most papers on instrument recognition of separate tones are unfortunately using sounds from the same instrument Instances both in the Learning and Test sets, a fact which raises a strong doubt regarding the applicability of their results, which are often unrealistically high (Livshin and Rodet 2003; see Chapter 9). Even so, while the current results are obtained by Minus-1-Instance evaluation, they are still higher or comparable to most instrument recognition rates reported by papers on instrument recognition of separate tones, regardless of their evaluation techniques. In (Livshin and Rodet 2003) for example, an average Minus-1-DB recognition rate of 83.17% for seven instruments is achieved.

It is interesting to note, that the main difference between the classification performed in this chapter and the one we used in (Livshin and Rodet 2003) is that our current sound set is much larger and more diverse. This exemplifies well an intuitive claim from (Livshin and Rodet 2003), which states that enriching a Learning set with sound samples from different databases improves its generalization power.

11.8.1.2 Harmonic Samples

	bsn	cl	fl	tbn	tr	cb	cbp	Vl	vlp	pno
bsn	**97**	1	0	0	0	0	0	1	0	0
cl	0	**88**	5	0	4	2	0	0	0	0
fl	0	4	**94**	0	0	0	0	1	0	1
tbn	3	0	0	**89**	8	0	0	0	0	0
tr	0	9	5	4	**82**	0	0	1	0	0
cb	1	0	0	0	0	**95**	0	4	0	0
cbp	0	0	0	0	0	0	**94**	0	2	4
vl	0	1	5	0	0	1	0	**92**	0	0
vlp	0	0	0	0	0	0	6	0	**90**	4
pno	2	0	0	0	0	1	1	0	2	**94**

Table 11-B. Confusion matrix of the Harmonic samples

The average Minus-1-Instance recognition rate per instrument for the resynthesized samples is 91.51% (using K=10 with KNN). This recognition rate is only 4.07% lower than the rate achieved using the Original samples, and is still quite high. This rate shows that the information in the Harmonic Series of the signal is quite enough for achieving a high average instrument recognition rate which is rather close to the rate obtained using the complete signals. Comparing the confusion matrices of the Harmonic samples (Table 11-B) to the Originals (Table 11-A), we can see that the recognition rate of almost

all the instruments has worsened somewhat, which consistently indicates that some instrument-discriminating information was lost. The most noticeable declines are the trumpet (-13.68%) and the violin pizzicato (-9.69%).

11.8.1.3 The Residuals

	bsn	cl	fl	tbn	tr	cb	cbp	vl	vlp	pno
bsn	**87**	2	3	0	0	3	0	1	3	0
cl	3	**76**	11	2	2	0	0	5	0	1
fl	0	12	**75**	1	7	0	0	6	0	0
tbn	11	4	3	**62**	6	0	6	5	0	3
tr	0	6	15	3	**74**	0	0	2	0	0
cb	2	1	0	0	0	**97**	0	0	0	0
cbp	0	0	0	0	0	0	**97**	0	0	3
vl	1	10	7	0	2	5	0	**74**	1	1
vlp	2	3	0	2	0	0	0	2	**90**	2
pno	1	0	0	1	0	0	4	0	5	**89**

Table 11-C. Confusion matrix of the Residuals

The average Minus-1-Instance recognition rate for the Residuals is 81.99% (using K=6 with KNN), which is 13.59% lower than the rate achieved with the Original samples. While this is quite a considerable difference, these results do indicate, perhaps surprisingly, that the Residuals by themselves (yes, these "airflow" and "click" sounds) contain considerable distinguishing instrument information. As this experiment did not involve any descriptors "tailored" specifically for the Residuals, it seems reasonable to expect that the recognition rate could be improved further.
The instrument with the mostly reduced recognition rate compared to the Original samples is the trombone (-30.55%), which is now confused mainly with the bassoon.

11.8.2 BEST 10 FEATURE DESCRIPTORS

Using CFS with a greedy stepwise forward search method, the best 10 feature descriptors were selected for each of the three sample groups out of the total 513 different feature descriptors in the Full Feature Descriptor set.

Feature Type	Descriptor Flavor	ST	O	H	R
Relative Specific Loudness	Mel-Band #2	M	1	1	5
Temporal Increase			2	4	X
Spectral Kurtosis	log frequency, normalized db amplitude	M	3	X	X
MFCC	Coefficient #2	M	4	X	X
Temporal Decrease			5	2	X
Roughness	ERB filter #8		6	5	3
Spectral Spread	linear frequency, normalized db amplitude	M	7	10	10
Bark-Band Tristimulus	linear amplitude, bands(2+3+4)/sum(all)	M	8	X	X
Bark-Band Tristimulus	linear amplitude, band(1)/sum(all)	S	9	X	X
Temporal Centroid			10	8	4
Spectral Skewness	linear frequency, linear amplitude	M	X	3	X
Bark-Band Tristimulus	normalized db amplitude, band(1)/sum(all)	M	X	6	X
Inharmonicity		M	X	7	X
Harmonic Spectral Roll-Off		M	X	9	X
Bark-Band Tristimulus	normalized db amplitude, bands(2+3+4)/sum(all)	M	X	X	1
Fluctuation Strength	mean (ERBs)		X	X	2
MFCC	Coefficient #4	M	X	X	6
Perceptual Spectral Centroid	linear frequency, power amplitude	M	X	X	7
Roughness	ERB filter #5		X	X	8
MFCC	Coefficient #3	M	X	X	9

Table 11-D. The 10 best features for the Original, Harmonic and Residual sample groups, selected using CFS

The "Feature Type" column shows the feature type, while the "Descriptor Flavor" column shows the parameter types used with each feature. For more information on the features, see Chapter 6.

Most features are computed on each STFT frame of the signal separately and then either the mean ('M') or the standard deviation ('S') of these frames is used. For such features,

the "Frames" column specifies which of these statistics was used. The "O", "H" and "R" columns indicate the Original sample group, the Harmonic samples and the Residuals, and show which feature descriptors were selected for these sample groups and in which order of importance, from 1 to 10. An X indicates that a feature was not selected.

Out of the 10 "best" descriptors, the Original samples "share" six descriptors with the Harmonic samples; out of these, four are shared also with the Residuals. The other descriptors are unique. Although Table 11-B shows that the Harmonic signals contain enough distinguishing instrument information for getting a rather high recognition rate (resulting in a recognition loss of only 4.07% compared with the Original samples), Table 11-D shows that removing the non-harmonic residuals has caused a somewhat different set of 10 features to be selected by the CFS feature selection algorithm. This indicates that for AMIR purposes the Harmonic set is not equivalent to the Original samples as it requires somewhat different features.

The recognition rates using only these sets of 10 selected feature descriptors are 76.11% for the Original samples, 65.24% for the Harmonic samples and 62.69% for the Residuals. The fact that with the 10 "best" descriptors, the recognition rate of the Harmonic samples is 10.87% lower than the Original samples seems to indicate that although the Harmonic samples get a high recognition rate with the Full Feature descriptor set (see Chapter 6), yet when the number of descriptors is reduced to a few selected ones, the Harmonic set requires more descriptors than the Original set to get similar recognition rates.

11.9 CONCLUSIONS

This chapter shows that using only information present in the Harmonic Series of the signal is enough for achieving a high average musical instrument recognition rate – 91.51% for 10 instruments using Minus-1-Instance evaluation. This is only 4.07% less than the recognition rate obtained by using the complete, Original signals.

On the other hand, Table 11-C shows that there is much distinguishing instrument information present in the non-harmonic Residuals which by themselves produced an average instrument recognition rate of 82%. It was also shown that the information present in the non-harmonic Residuals is not completely redundant to the information present in the Harmonic Series; Table 11-B shows that although the average recognition rate of the Harmonic signals is high, some of the instruments have suffered noticeably from removing the non-harmonic Residuals, especially the trumpet and violin pizzicato, which is an interesting result. In addition, Table 11-D shows that the 10 best feature descriptors selected for the Original sample set differ from the ones selected for the Harmonic samples. These results indicate that the sound of pitched musical instruments should not be treated as containing only the Harmonic Series, although most of the energy and distinguishing instrument information of the signal is indeed present in the Harmonic Series.

It was shown that using only the harmonic series does not considerably lower the average instrument recognition rate although some instruments "suffer" more than others. This means that instrument recognition in MIP music could indeed be performed with rather high results without performing full source-separation; Using multiple *f0* estimation algorithms, such as (Yeh, Röbel and Rodet 2005), estimated harmonic partials could be used solely to classify musical instruments without lowering too much the recognition rates compared with the full signal - see Section 13.1.3 for the Harmonic-Resynthesis technique.

11.10 FUTURE WORK

It might be possible to increase the instrument recognition rate of the Residuals by specifically tailoring special feature descriptors for them. Instrument recognition of pitched instruments could then be improved by splitting the classified sounds into harmonic and non-harmonic components (when applicable) and computing special feature descriptors on the Residuals in addition to the feature descriptors computed on the original signal. Splitting the signal makes it easier to deal with the non-harmonic Residuals, due to their relatively low energy.
Using the current Full Feature Descriptor set, experiments where the descriptors of all three sample sets were merged together (Original + Harmonic + Residuals) have not yielded higher recognition rates than using the Feature descriptor set of the Original samples by itself.

CHAPTER 12 AMIR IN SOLOS

Instrument recognition in Solo performances (monophonic or polyphonic musical phrases performed by a single instrument) is different and more complicated than dealing with separate note databases, as the time evolution of each sound (attack, decay, sustain, release) is not well defined, the notes may not be separated, there are superpositions of concurrent sounds and room echo, different combinations of playing techniques, etc.
For history of AMIR in Solos see Section 3.2.

12.1 MOTIVATION

Although it seems that there are not many applications which actually require Solo recognition, yet as shall be demonstrated in the next chapter, knowledge of performing AMIR in Solos can help in recognition of MIP music. There are also some applications for Solo recognition per se, for example, labeling tracks in multi-track recordings, in real-time or offline.

12.2 DATA SET

For evaluation of AMIR in Solos, a large and very diverse Solo database (See Section 5.2 for more details) is used in order to encompass the different sound possibilities of each instrument and evaluate the generalization abilities of the classification process.

Reminder - this collection includes 108 authentic Solo recordings of 7 instruments: bassoon, clarinet, flute, classical guitar, piano, cello and violin. In order to evaluate

classification generalization (See Chapter 9), I made sure that all Solos are performed by different musicians, recorded in different concerts (thus different recording conditions) and the same Solo is never used fully/partly in both the Learning and the Test sets in the experiment.

Preprocessing:

A two-minute piece is taken from each Solo recording and cut into one-second cuts with 50% overlap (resulting in instrument segmentation resolution of half a second) – a total of 240 cuts out of each Solo. The feature descriptors are computed on each one-second Solo-cut separately.

12.3 CLASSIFICATION

The classification method is LDA+KNN (see Section 8.3).
The evaluation method is "Minus-1-Solo" (See Section 9.7.2).

12.4 REALTIME SOLO RECOGNITION

'Realtime' recognition of Solo performance means here that while the Solo is recorded or played the features of each one-second piece of the music are computed and classified immediately after it was performed.

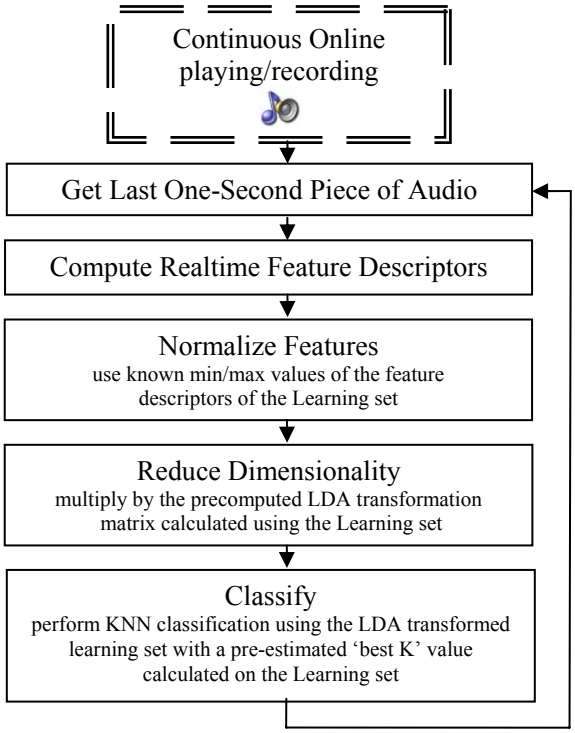

Figure 12-A. Realtime Solo recognition process

As the Full Feature Set of 62 features (see Chapter 6) takes a rather long time to compute, in order to achieve Solo recognition in realtime a smaller feature set is required, which is quick to compute but compromises the recognition rate as little as possible. To achieve that, the most time-consuming features were removed and GDE was used (See Section 7.2) to reduce the number of features from 62 (with 512 descriptors) to 20 (with 250 descriptors), while still maintaining a high recognition rate. Using these features we have actually implemented a realtime Solo AMIR program which works on a 1.6Mhz Intel Processor and is written in plain Matlab code (without compilation or integration with machine language boost routines). Naturally, this program uses a precomputed LDA matrix and pre-estimated 'best K' for the KNN classification, as the Learning set remains constant and should not depend on the Solo input.

We can see in Figure 12-A that the classification process uses at each round the last one-second of the recording, which makes the recognition 'resolution' increase in direct relation to the hardware speed and efficiency of the sub-algorithms being used.

12.5 MINUS-1-SOLO RESULTS

	One-second pieces		Instrument Detection	
	"Realtime" 20 features	**"Complete Set" 62 features**	**"Realtime" 20 features**	**"Complete Set" 62 features**
Bassoon	86.3	90.2	100	100
Clarinet	79.3	86.9	90.5	100
Flute	83.3	80.9	100	100
Guitar	86.3	87.8	93.8	93.8
Piano	91.0	93.9	100	100
Cello	82.2	88.7	94.1	100
Violin	88.3	88.5	92.9	92.9
Average	**85.2 %**	**88.1 %**	**95.9 %**	**98.1 %**

Table 12-A. Minus-1-Solo and Solo-Instrument Detection results using the Complete and Realtime feature sets

Table 12-A shows on the left the Minus-1 Solo recognition results for the "Complete" and "Realtime" feature sets. On the right side of the table, the Solo-Instrument Detection grade shows the percentage of Solos where the performing instrument gets the highest number of one-second pieces classified into its class, indicating whether the classification results could be used to determine which instrument is playing each Solo.

We can see that the average recognition rate per one-second piece using the "Realtime" set is rather close to the "Complete Set" - only 2.9% difference. The difference in average Solo-Instrument Detection grade is also small – 2.2%.

12.5.1 REALTIME FEATURE SET

Table 12-B shows the resulting 20 feature list for realtime classification of Solos, sorted by importance from the most important feature down to the least.

1. Perceptual Spectral Slope	2. Perceptual Spectral Centroid
3. Spectral Slope	4. Spectral Spread
5. Spectral Centroid	6. Perceptual Spectral Skewness
7. Perceptual Spectral Spread	8. Perceptual Spectral Kurtosis
9. Spectral Skewness	10. Spectral Kurtosis
11. Spread	12. Perceptual Deviation
13. Perceptual Tristimulus	14. MFCC
15. Loudness	16. Auto-correlation
17. Relative Specific Loudness	18. Sharpness
19. Perceptual Spectral rolloff	20. Spectral rolloff

Table 12-B. A sorted list of the most important features for realtime AMIR Solo recognition (for the seven musical instruments in this chapter)

We can see in Table 12-B that the 10 most important features are the first four statistical moments and the Spectral Slope, computed both using the perceptual and spectral models (see Chapter 6).

Note - in Section 13.1.3.2, Solo recognition is performed in a different way than in this chapter, using notes resynthesized from their harmonic series. As this method makes no sense outside the context of Chapter 13, where later-on classification of resynthesized notes from multi-instrumental music is performed, I chose to present this "Solo recognition" technique in Chapter 13 rather than here.

CHAPTER 13 AMIR IN MULTI-INSTRUMENTAL, POLYPHONIC MUSIC

For most practical applications AMIR needs to be performed on music in which several musical instruments are playing concurrently – Multi-Instrumental, Polyphonic (MIP) music. Recognition of musical instruments in MIP music is a difficult challenge due to the fact that the musical signals of the different instruments are mixed together and cannot be simply separated.

13.1 AMIR METHODS FOR MIP MUSIC

13.1.1 "NAÏVE" SOLO CLASSIFIER

What happens when a classifier trained to recognize Solos, such as the one used in Chapter 12, is activated directly on MIP music? Will it recognize one of the playing instruments or get "completely confused" by the multi-instrumental mix?

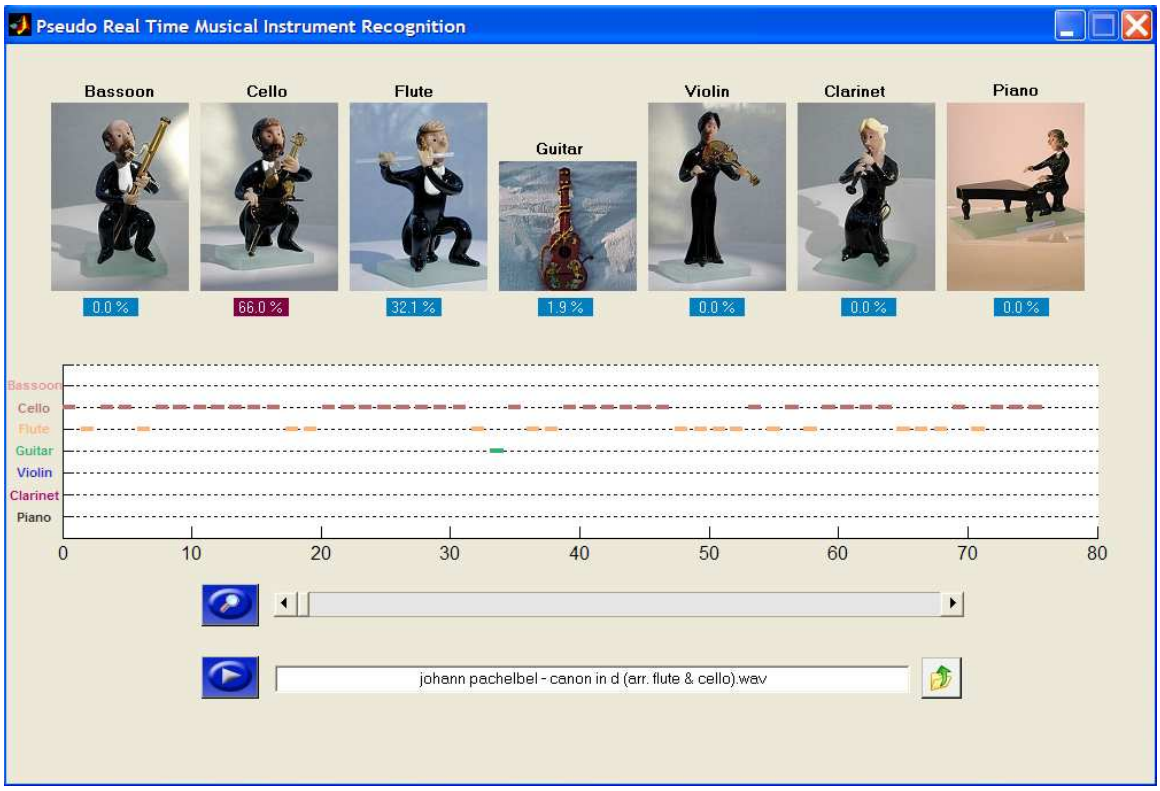

Figure 13-A. An example of the Naïve Solo recognizer ran on a flute and cello Duo

An obvious disadvantage of using a Solo classifier directly on MIP music is that it attempts to recognize only one instrument in each classified musical segment, regardless of the number of instruments actually playing concurrently.

The Naïve Solo classifier tested in this chapter performs AMIR on each consecutive one-second segment of the classified music, same as in Chapter 12; the length of musical segments in the Learning set is also one second.

13.1.2 SOURCE-REDUCTION (SR)

This new technique, like the Naïve Solo Classifier, uses an AMIR Classifier "trained" on a Learning set consisting of Solo segments. Nevertheless, in order to make this classifier "multi-dimensional", i.e., perform AMIR on each note separately even if these notes sound at the same time, each musical note in the classified MIP piece, in its turn, is made easier to recognize by reducing the volume level of all the other notes playing concurrently with it. This way, metaphorically, the AMIR "spotlight" is turned to each note individually. Next, as mentioned above, this filtered note is classified using an AMIR classifier of Solos such as the one presented in Chapter 12.

In an unpublished report (Livshin et al. 2005), we have integrated a Source-Reduction AMIR module with a multiple-*f0* estimation module to create a system which

automatically produces partituras out of recorded MIP music, by first finding the different notes in the musical piece and then arranging them into staves by their recognized musical instruments.

13.1.2.1 SR Algorithm

- A Multiple-*f0* detection program is used in order to estimate the note boundaries (where each note begins and ends) in a MIP musical piece, the fundamental frequencies of each note and its harmonic series. In the experiments in this chapter, the *f0*-estimation program of (Yeh, Röbel and Rodet 2005) is used with 93ms analysis window and automatic estimation of the polyphonicity, i.e. number of notes played concurrently, in each analysis window.
  In order to reduce the influence of accidental *f0*-estimation errors, notes shorter than a "minimum" threshold are discarded – the reason for this is that when the same note pitch (*f0*) is detected in several consecutive frames (vibrato/glissando are allowed), the probability of this note being an accidental estimation error is diminished. Recognition rates of two minimum-length note-thresholds are presented in the following section - 0.25second and 0.5second.

- Filter and classify each detected note:
  - Perform STFT of the music segment where the note is present
  - For each STFT frame:
    - Normalize DFT bin amplitudes to [0 - 1]
    - Find all peaks in the frame above a minimal threshold. In the experiments in this chapter the minimal normalized peak threshold is 0.02
    - In all DFT bins belonging to peaks which do not contain partials of the detected note, put 0
      Currently, peaks containing overlapping partials, i.e., partials of the detected (classified) note along with partials of other notes, are left unmodified[16]. The expectation that notes "dirty" with some overlapping partials from other notes could still be classified correctly is based upon the experimental results in Section 13.2 using the Naïve Solo Classifier; these results show that when a Solo classifier is activated on a MIP musical segment, it usually succeeds to recognize one of the playing instruments. Thus, while some overlapping partials of an interfering note may remain, if most of the interfering note is removed then the chances of the musical segment to be classified as the current, detected (non-filtered) note, are increased considerably.
      Note - various, more sophisticated approaches to the problem of overlapping partials exist. Eggink and Brown (2003) for example, use GMMs with the missing feature model to exclude overlapping partials

[16] That is the reason why the correct term for this technique is "Source-Reduction" and not "Source Separation" – full separation of the sound sources is not attempted here, but rather, as noted above, a "reduction" of all sounds except a selected note in order to make it more easily distinguishable by an AMIR Solo classifier.

from the classification process. Virtanen (2002) computes linear models for the overtone series and uses these in order to estimate the amplitudes of overlapping partials. While producing interesting results, these methods still need to be perfected.

- Classify the filtered note using a Solo Classifier. In this chapter the same Solo classifier as presented in Chapter 12 is used; this classifier is trained on one-second Solo pieces with 50% overlap, from the Solo database in Section 5.2.

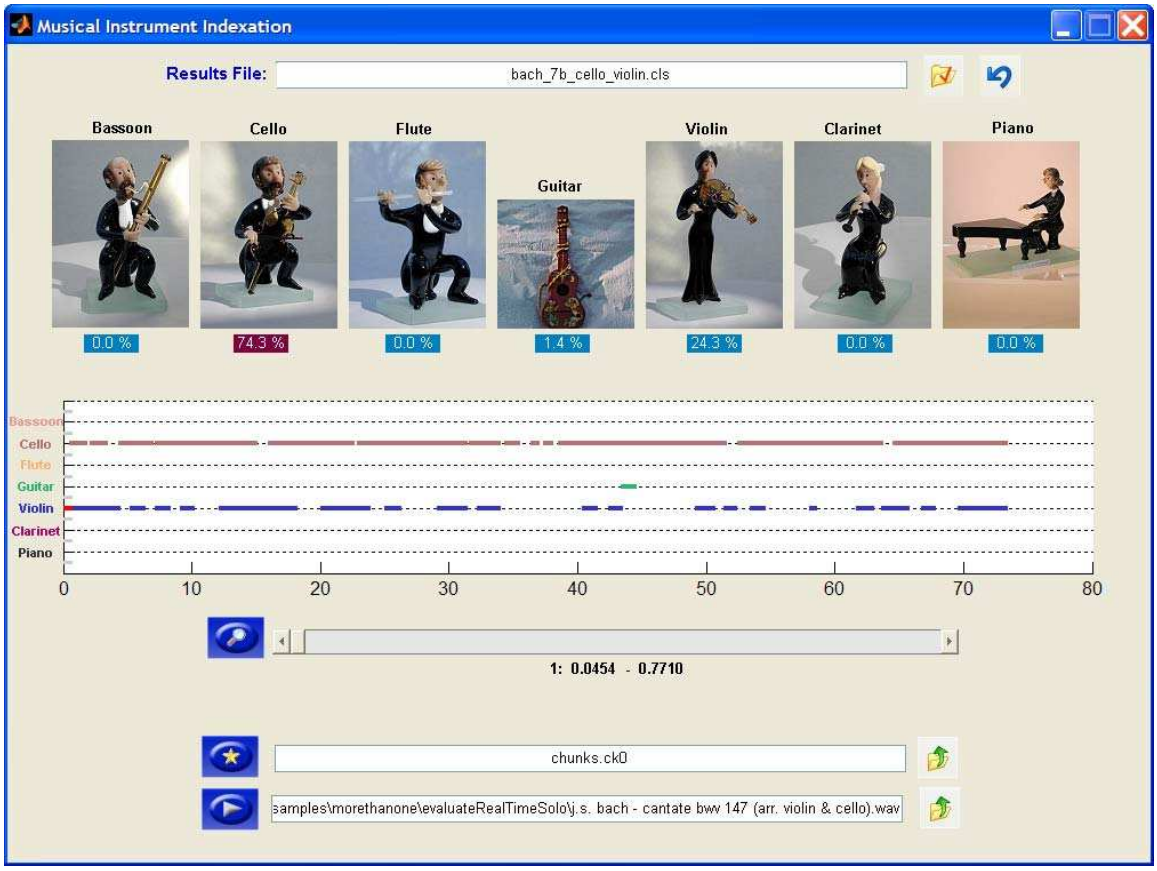

Figure 13-B. An AMIR example using Source-Reduction

Note that when performing AMIR on Duos of two monophonic instruments, it is possible to use an "Anti-Note" strategy in order to classify also some of the notes that are shorter than the minimum note-threshold mentioned above:
It was shown in Chapter 12 that in order to classify correctly an instrument using a Solo classifier there is no need for each of the played notes to be classified separately and even Solo segments containing polyphonic music, such as played by piano and guitar, could be recognized successfully. The main reason, in Source-Reduction, for filtering out all notes playing concurrently except a single one is that the remaining note is assumed to be performed by a single instrument as notes played in complete unison by several instruments are very rare.

In Duos performed by monophonic instruments it could be quite safely assumed that when one instrument is playing a note then all the other notes, of various lengths, sounding at the same time with that note, are performed by the other instrument. Therefore, it is possible, when detecting a note of at least a minimal length, besides filtering everything else out and leaving that note to be classified, to do the complete opposite and filter out the note itself, classifying all that remains, as these "Anti-Notes" could be assumed to be performed by a single instrument[17].

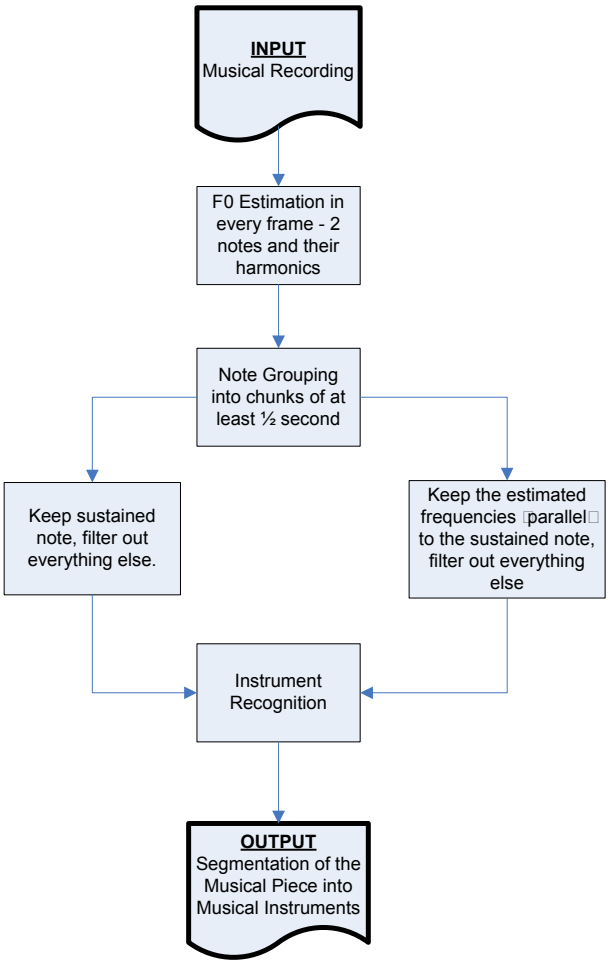

Figure 13-C. An example of the Source-Reduction process - a special Duo version including the "Anti-Note" strategy

[17] Classifying automatically the Anti-Notes as belonging to the other instrument is unadvisable, as a recognition error of the detected note will automatically cause recognition error of the Anti-Notes.

13.1.3 HARMONIC-RESYNTHESIS (HR)

It was shown in Chapter 11 that when performing AMIR on separate tones, using solely information present in the harmonic series is enough for achieving high instrument recognition rates.
Automatic estimation of fundamental frequencies in MIP music, such as presented in (Yeh, Röbel and Rodet 2005), is a popular research topic producing constantly improving algorithms; using such automatic multiple-*f0* estimation results allows approximate estimation of the harmonic series of the playing notes.

Therefore, it is possible to use the results of an automatic multiple-*f0* estimation program to estimate the harmonic series of the musical notes in MIP music, and then use these harmonic series for performing AMIR, thus creating a completely automatic MIP AMIR system.

13.1.3.1 HR Algorithm

- A Multiple-*f0* detection program is used in order to estimate the note boundaries in a musical piece, the fundamental frequencies of each note and its harmonic series. As in Source-Reduction, in this chapter the *f0*-estimation program of (Yeh, Röbel and Rodet 2005) is used with 93ms analysis window and automatic estimation of the polyphonicity, i.e. number of notes played concurrently, in each analysis window. In order to discard accidental *f0*-estimation errors, notes shorter than a "minimum" threshold are discarded – the reason for this is that when the same note pitch (*f0*) is detected in several consecutive frames, the probability of this note being an accidental estimation error is diminished. Recognition rates of two minimum-length note-thresholds are presented below - 0.25second and 0.5second.

- For each detected note (vibrato/glissando are allowed):
  - Resynthesize[18] it using its estimated harmonic series (frequencies and amplitudes)
  - Classify it using a Learning set consisting of notes from Solos, resynthesized from their estimated harmonic series.

13.1.3.2 Resynthesized-Solos recognition

In order to have the Learning and Test sets as similar as possible, HR uses a classifier trained on notes resynthesized from estimated harmonic series computed from Solos, to perform AMIR on notes resynthesized from estimated harmonic series from MIP music.
Before classifying resynthesized notes from MIP music, it is important to check that this resynthesized Solo classifier can indeed classify resynthesized Solo notes well. The Solo collection is the same as used in the Solo recognition in Chapter 12 (for details see

[18] Resynthesis is not compulsory; the harmonic series could be used directly for descriptor computation. Here, as many of the descriptor calculation routines are 3'rd party and require a waveform as input, the notes are resynthesized "back" from the harmonic series without loss of generality (WLOG).

Section 5.2), and includes Solos of seven instruments: bassoon, clarinet, flute, cello, violin, guitar and piano.

Resynthesized-Solo classifier evaluation process:

- An *f0*-estimation program is run on the Solos
- All notes in the estimation results which are at least 0.25second long (to prevent accidental estimation errors) are detected
- The detected notes are resynthesized using their estimated harmonic series using Additive Synthesis (see Section 11.4 for explanation of Additive Synthesis)
- The descriptors of the resynthesized notes are computed
- AMIR classification evaluation is performed using the "Minus-1-Solo" method (Section 9.7.2): the resynthesized notes of each Solo (actually, their descriptor vectors) are classified using a Learning set consisting of the notes of all the other Solos together

Results

	Bassoon	Clarinet	Flute	Cello	Violin	Guitar	Piano	**Average**
Resynthesized Notes	88.7	72.9	83.2	73.6	80.5	82.4	90.3	**81.7**
Solo Instrument Detection	90.9	100	100	100	100	94.1	90.0	**96.4**

Table 13-A. "Minus-1-Solo" average recognition rates using resynthesized Solo notes

The "Resynthesized notes" row in Table 13-A shows the Minus-1-Solo recognition rates for the resynthesized Solo notes. The "Solo Instrument Detection" row of the table shows the percentage of Solos where the performing instrument gets the highest number of notes classified into its class, implying whether the classification results could be used to determine which instrument is playing each Solo.

Note that while these results seem somewhat lower than the recognition rates for the same Solo database (Table 12-A) using non-resynthesized, original Solo-pieces, it should be reminded that these results cannot be directly compared.
In Chapter 12, the samples used for Solo recognition (also used as the Learning set with the SR method) are sequentially cut, half-overlapping, one-second pieces out of the Solos. Here, the current method looks for continuous notes (of at least 0.25second) in the *f0*-estimation results and then resynthesizes them. The resynthesized Solo database consists of 19769 *notes*, while the non-resynthesized, "original" Solo database in Chapter 12 has 22464 one-second *samples*.

To conclude - the results in Table 13-A show that this classifier can indeed classify resynthesized Solo notes rather well, which means that the Resynthesis method is indeed working for Solos, while the recognition rates with the current *f0*-estimation program are somewhat lower than with non-resynthesized Solo pieces.

13.1.3.3 HR vs. SR

- Both systems in this chapter use estimated harmonic-series computed by a multiple-*f0* estimation program - (Yeh, Röbel and Rodet 2005), and thus do not require pre-given *f0*-information and can be directly applied to authentic musical recordings, unlike (Kitahara et al. 2007) and (Eggink and Brown 2003) for example.

- Both HR and SR methods first detect note boundaries in the *f0* estimation data and then perform AMIR on each detected note, thus, unlike (Essid, Richard and David 2006) for example, can be directly utilized for automatic transcription, ordering notes into staves according to the playing instrument and can be directly applied to score-based annotation systems such as MusicXML (Good 2001).

- Both methods do not use heuristics based on musical context, such as proposed by (Kitahara et al. 2007), and thus can be applied indiscriminately to any musical genre, including classical music, modern atonal music and even random sound mixtures.

- The HR method is simpler somewhat than SR as it does not require filtering. It can be concluded from Chapter 11 that if the multiple-*f0* detection algorithm provides reasonable results, the HR method should work well. SR, on the other hand, depends also on the filtering method.

- The SR method has less strict demands from the *f0*-estimation program than HR; it requires less precision in the partial frequencies estimation and does not require partial amplitude estimation at all:
  The filtering process in the SR method leaves the DFT peaks intact – it does not attempt to surgically remove all the partials of undesired notes, but rather to remove whole peaks which do not contain partials of the "desired" note inside them (the note we wish to keep). This method, while somewhat rough, is also somewhat "safe", as it leaves partials of the desired note untouched and only removes "undesired" information. This means that the Reduction method does not require the *f0*-estimation program to supply energy levels of the partials, which are very hard to estimate in cases of partial collisions, but only frequencies, while the Resynthesis method requires the amplitudes of partials. This also means that the *f0*-estimation program has some freedom of error regarding the partial frequencies, as long as they still "land" inside the partial peaks while HR resynthesizes exactly the estimated frequencies.
  Note however, that the *f0*-detection program used in this chapter (Yeh, Roebel and Rodet 2005) does not attempt to estimate the amplitudes of overlapping partials but simply reports the amplitudes in the DFT bins corresponding to the partials and therefore the amplitude requirement of the HR method does not greatly differ in this case from the inherent amplitude-ignorance of the SR.

13.2 EVALUATION RESULTS

13.2.1 AUTHENTIC DUO RECORDINGS

Table 13-B shows the average recognition rates of the three methods described above for AMIR in MIP music, Naïve Solo Classifier, SR and HR, performed on 18 authentic Duo recordings (see Section 5.3); these Duos are performed by different pairs of seven musical instruments – bassoon, flute, clarinet, violin, cello and the polyphonic instruments - guitar and piano.

The "Recognition Method" column specifies the AMIR technique.
The SR and HR methods are evaluated on all detected musical notes in a Duo which are longer than a selected minimal note length. Different minimal note lengths were preliminary tested and two lengths were selected to be shown in this table – 0.25second and 0.5second. Note that limiting the note sizes to lengths above 0.5second has decreased the number of classified notes down to 45.4% compared with using a minimum threshold of 0.25second. Minimal note length thresholds higher than 0.5second leave very few notes for classification, while notes significantly shorter than 0.25second are mostly too short to be classified correctly.

As exact instrument labeling for the tested Duos is unfortunately not available, the recognition rates are computed using the average "Mutual Grade" method, which indicates the percentage of notes which are classified correctly as either one of the playing instruments.

The "Instrument Detection" columns indicate whether both correct instruments received the highest number of classifications, i.e. whether the classification results could be used to determine exactly which instruments are playing in the Duo.
This grade is important; if the maximum number of instruments in a musical piece is known and the classifier can be relied upon to correctly determine them (e.g., the two correct instruments in a Duo), then it is possible afterwards to limit the Learning set to contain only the participating instruments and perform the recognition process again, this time getting a much more precise instrument-segmentation of the musical piece exclusively into the correct playing instruments. Obtaining the list of participating instruments is also useful for performing other MIR tasks, such as *f0*-estimation and score alignment, which could utilize instrument models of the correct participating instruments.

Instrument Detection levels:

- The "Both" sub-column indicates the percentage of Duos where both instruments were identified.
- The "Only One" sub-column shows the percentage of Duos where only one correct instrument got the majority vote of classifications while the other instrument "lost" the vote to another, incorrect, instrument.

- The "Neither" sub-column shows the percentage of Duos where none of the correct instruments got a majority classification.

Recognition Method	"Mutual" Grade	Instrument Detection		
		Both	Only One	Neither
Naïve Solo-Classifier	87.5 %	77.8 %	22.2 %	0 %
Source-Reduction 0.25	78.6 %	88.9 %	5.6 %	5.6 %
Source-Reduction 0.5	83.0 %	77.8 %	22.2 %	0 %
Harmonic-Resynthesis 0.25	80.9 %	94.4 %	5.6 %	0 %
Harmonic-Resynthesis 0.5	83.2 %	100 %	0 %	0 %

Table 13-B. AMIR in authentic Duo recordings of seven instruments, using the Naïve Solo classifier, Source-Reduction and Harmonic-Resynthesis methods

Discussion – Authentic Duo Results

Naïve Solo Classifier

Looking at Table 13-B, it is interesting to note that the Naïve Solo classifier is not much "confused" by the MIP mixture (Livshin and Rodet 2004a); it classifies 87.5% of the one-second samples correctly as either one of the playing instruments and succeeds to determine correctly both participating instruments in 77.8% of the Duos.

One should keep in mind, however, that the Naïve recognition is only one-dimensional, i.e., each one-second piece of the music is classified only once, as if it was performed by a single musical instrument. Another disadvantage is that the Naïve method classifies here constant-length musical segments, ignoring note boundaries. For this reason, the Naïve method cannot be directly compared with the multi-dimensional, single-note classifiers - Source-Reduction and Harmonic-Resynthesis, which classify each note separately and can classify notes that overlap in time.

The 77.8% Instrument Detection grade seems to suggest that while the Naïve recognizer may recognize only the "domineering", higher-volume, instrument in each one-second musical segment, usually there is no complete "domination" of one of the instruments throughout the musical piece as both participating instruments could be detected.

The ability to recognize the "domineering" instrument in signals that are comprised of two concurrent sounds seems to imply that sounds which contain a domineering note with some extra, low-level "leftovers" from other notes, such as may remain in filtered notes produced by the Source-Reduction method, may still be correctly classified in most cases.

SR and HR

Comparing the Source-Reduction (SR) and Harmonic-Resynthesis (HR) results, the table shows that HR performs better than SR – it produces a slightly higher mutual grade than SR for both minimal note sizes – 2.3% better with 0.25second note threshold and 0.2% better for notes over minimal length of 0.5second. The HR method also detects better

than SR the participating instruments and when using a minimal note length of 0.5second, it actually succeeds to determine correctly the playing instruments in all 18 Duos.

While both SR and HR methods produce higher grades using minimal note length of 0.5second, it is important to keep in mind that the number of classified notes is actually reduced by 54.7% when limiting classification to notes starting from 0.5second compared with 0.25second threshold. While this is not important if the goal is to determine the instruments playing in the musical piece, when the requirement is to instrument-segment as much notes as possible, for automatic transcription applications for example, then it is better to use 0.25second minimal note length with a slight compromise on recognition rate than to leave more than half the notes in the musical piece unclassified.

13.3 SOLO MIXTURES

Authentic musical recordings with their precise transcriptions, as required for exact AMIR evaluation, are practically unavailable. Different alternative approaches exist, with the most common one being using music resynthesized or automatically mixed from separate tones according to MIDI files, such as done in (Kitahara et al. 2007). Music resulting from this approach, however, is too different from authentic musical recordings to fairly represent them in AMIR experiments.
Artificially mixing authentic Solo performances is a rather good compromise which produces precisely instrument-labeled MIP music which is quite similar to authentic recordings of MIP music.
For full details of my Solo-Mixing process and a more thorough discussion of the advantages and disadvantages of evaluation using Solo-mixtures, see Section 5.4. For a discussion regarding the requirement for precise AMIR evaluation and proposal of several possible solutions, see Section 15.4 in the "Future Work" chapter.

In order to perform precise instrument labeling of the Solo mixtures, it is first required to detect the precise note boundaries (where each note begins and ends) in the Solos which participate in each mix. While precise boundary detection is relatively easy with Solos performed by monophonic and semi-monophonic instruments, detecting note boundaries in Solos performed by polyphonic instruments, such as the piano and guitar, is as difficult as doing so in MIP music. Therefore, in this section, which is dedicated to providing precise MIP AMIR evaluation, the polyphonic instruments were removed from the Solo database (described in Section 5.2), resulting in Solo-mixes of five monophonic and semi-monophonic instruments – bassoon, flute, clarinet, violin and cello.
Limiting the Solo-mixes to monophonic instruments also allows setting precisely the concurrent number of notes played in the Solo-mixes; mixing Solos of polyphonic instruments, such as the guitar, would have created mixes with non-determined numbers of notes playing concurrently depending upon the guitar passages.

13.3.1 INDEPENDENT EVALUATION

Before classifying each Solo-mixture, in order to keep the evaluation "fair" (see Chapter 9) the Solos which were mixed into it are removed from the Learning database; this causes lowering of the recognition rate but allows truly independent evaluation.

13.3.2 GRADING

Several different grade methods are used in order to show different strengths of the tested recognition methods:

13.3.2.1 Instrument Grade

Computing the Instrument Grade:

- Each note is checked whether it is classified exactly as the instrument playing it, i.e. its instrument-label
- An average grade for each participating instrument is computed for each Solo-mix
- The Instrument Grade is the average of these average instrument-grades over all the Solo-mixes

13.3.2.2 Notes Grade

Computing the Notes Grade:

- Each note is checked whether it is classified correctly as its instrument label
- The average recognition rate per note is computed for each mixture
- The Notes Grade is computed as the average of the average note-grades in all the Solo-mixes

The Notes Grade differs from the Instrument Grade as the number of notes performed by different instruments in a Solo-mix is different. The instrument grade indicates how well each instrument is recognized while the Pieces Grade indicates how well each musical note is classified.

13.3.2.3 Mutual Grade

This is a version of the Notes Grade, where a note is considered to be classified correctly if it is classified as either one of the instruments mixed in the mixture. This grade was used in the previous section where AMIR was performed on authentic Duo recordings.

The Mutual Grade in this section attempts to produce a grading which diminishes the influence of recognition errors between participating instruments, such as may be caused by incorrect partial-amplitude estimation by the *f0*-estimation program or imprecise

filtering; while listening to some resynthesized HR notes or some filtered SR notes it indeed happened sometimes that I heard several of the instruments playing concurrently.

13.3.2.4 "Vote" Grade

This grade shows how well the AMIR algorithm allows precise detection of the instruments participating in the MIP music. A high Vote grade means that the resulting instrument lists are precise enough to be used practically, for example, in order to pass corresponding instrument models to an *f0*-estimation program. While for Duos, in the previous section, a simple Instrument Detection grade was sufficient, for more multi-instrumental music a somewhat complex grade computation method is required.

Computing the Vote Grade:

- Produce a sorted list of the instruments in a Solo-mixture by the number of samples each of the instruments "received" from the classification
- If correct instruments get the highest places in the list, each correct instrument gets a full 100% rate. If some participating instruments get, however, the same number of "votes" as some incorrect instruments, the 100% is divided among all of these instruments with the same number of "votes".
- A number of "Reserved Places" as the number of participating instruments is kept for musical instruments that get the highest number of classifications, regardless of whether these are indeed participating instruments or wrong ones.
- The Vote Grade is the average of the grades of the participating instruments that get Reserved Places

Example:
Let us say that the analyzed musical piece is a clarinet and bassoon Duo. The number of samples classified as different instruments are:

Bassoon: 8 samples, Clarinet: 4, Flute: 4, Violin: 4, Cello: 0

Bassoon, which has the majority of classifications (8 samples) gets 100% and occupies the first of the 2 places reserved for correct instruments (this is a Duo - 2 instruments are playing).
Clarinet, however, has 4 samples, the same as the incorrectly identified flute and violin, and thus it gets the "number of remaining places" divided by the "number of instruments with same number of votes" = 1/3 = 33.3%.
The average Vote grade in this example is therefore (100 + 33.3) / 2 = 66.7%.

13.3.2.5 "Vote All" Grade

This utter simplification of the "Vote" Grade, analogical to the "Both" sub-column of the "Instrument Detection" column in Table 13-B, simply gives 100% to Solo-mixes where all participating instruments get full majority and 0% to all the other mixes, i.e., it

indicates the percentage of mixes where all the participating instruments are identified as such.

13.3.3 RESULTS

Reading Table 13-C:

- "Info." column:
  - Polyphony – number of musical instruments playing concurrently in this experiment
  - Mixes – number of Solo-mixes classified in this experiment
  - Solos – number of different Solos mixed in this experiment
  - Unlabeled – average percentage of notes in the Solo-mixes, over 0.25second long, which were not detected during the instrument-labeling procedure in the Solos participating in the mix and therefore could not be labeled and were exempted from classification (see Section 5.4)
  - Over 0.5s – two minimal sample sizes were tested – 0.25second and 0.5second. This percentage shows how many samples are in the [0.25second - 0.5second] range, i.e. removed during the experiments with a minimal sample size of 0.5second

- "Min. Len." column:
  Indicates the minimal classified sample length - 0.25second or 0.5second. Note that sometimes it happened that there were no notes of a certain instrument over 0.5second long at all (may happen also due to *f0*-estimation errors), resulting in this instrument actually "disappearing" from classification. This is the reason why some of the 0.5second grades may seem bizarre and much lower than expected. For example, the HR grade of 5 voice polyphony is only 60% instead of the normally expected 100%. Yet this is fair – if only notes of 4 instruments are left, classifying a sample as the 5'th instrument, even if this instrument existed in the original mixture, is wrong.

- "Alg." column:
  Indicates the AMIR MIP algorithm used in the experiment, SR (Source-Reduction) or HR (Harmonic-Resynthesis)

- "Instr. G." column: average Instrument Grade
- "Notes G." column: average Notes Grade
- "Mutual G." column: average Mutual Grade
- "Vote G. column: average Vote Grade
- "VoteAll G.": average Vote-All Grade

In columns 4 – 8 the numbers in the brackets indicate the K which was used for KNN in each test type. This K, selected from a range of 1 – 80, has produced the highest results for this kind of test.
It is reasonable to use different K values depending on which type of grade we are most interested in. Nevertheless, in case where we do want to use the same K value for all the polyphonic voice range (2 – 5 voices), the average difference in grade between using "best" K's for each voice number or just a constant K is rather small, around 2%.

- The bottom double-row of the table shows average results for each technique over the different numbers of voices.

Info.	Min. Len.	Alg.	Instr. G.	Notes G.	Mutual G.	Vote G.	VoteAll G.
Polyphony:2 Mixes:34 Solos:68 Unlabeled:9.1% Over 0.5s:54.4%	0.25s	SR	66.4(15)	68.0(15)	79.5(74)	82.8(6)	67.6(6)
		HR	66.3(26)	66.3(30)	77.4(12)	84.6(6)	67.6(6)
	0.5s	SR	67.6(68)	69.2(18)	80.8(74)	89.0(3)	76.5(3)
		HR	**68.9(53)**	66.7(26)	**78.3(53)**	**83.6(21)**	64.7(19)
Polyphony:3 Mixes:20 Solos:60 Unlabeled:7.6% Over 0.5s:51.2%	0.25s	SR	53.3(13)	**56.8(13)**	**84.7(17)**	**82.5(12)**	**40.2**(14)
		HR	57.3(7)	59.0(20)	86.2(7)	89.2(4)	65.0(4)
	0.5s	SR	54.6(3)	54.7(2)	86.0(17)	84.3(6)	50.0(6)
		HR	60.5(11)	60.2(11)	85.4(43)	86.2(34)	60.0(3)
Polyphony:4 Mixes:14 Solos:56 Unlabeled:5.4% Over 0.5s:49%	0.25s	SR	**49.6(9)**	**55.4(9)**	**94.8(70)**	**98.2(4)**	92.9(4)
		HR	56.9(44)	58.9(4)	92.8(2)	91.7(6)	64.3(3)
	0.5s	SR	51.1(5)	57.0(9)	94.0(68)	95.8(17)	85.7(17)
		HR	58.2(8)	60.1(68)	91.6(12)	94.0(12)	78.6(12)
Polyphony:5 Mixes:10 Solos:50 Unlabeled:2.1% Over 0.5s:41.6%	0.25s	SR	38.8(62)	41.7(64)	100.0(1)	100.0(1)	100.0(1)
		HR	**43.3(27)**	**45.5(27)**	100.0(1)	100.0(1)	100.0(1)
	0.5s	SR	43.0(62)	41.9(62)	99.4(1)	98.3(1)	90.0(1)
		HR	**40.1(24)**	**38.0(24)**	**86.3(1)**	85.8(5)	60.0(4)
AVERAGE Unlabeled:6.0% Over 0.5s:49%	0.25s	SR	52.0	55.5	89.7	90.9	75.2
		HR	55.9	57.4	89.1	91.4	74.2
	0.5s	SR	54.1	55.7	90.0	91.8	75.5
		HR	56.9	56.2	85.4	87.4	65.8

Table 13-C. Source-Reduction and Harmonic-Resynthesis AMIR results on Solo-Mixtures

Looking at the Average double-row (bottom one), we can see that HR produces slightly higher Instrument Grades and Notes Grades than SR, while SR leads somewhat with the Mutual and Vote-All Grades.

Both methods produce better average grades using notes of at least 0.5second long than with 0.25second minimal threshold, however, on average, using 0.5second minimal note length leaves only 49% of the notes to be classified, i.e. there is twice the recognition resolution using a minimal note length of 0.25second. As the difference in recognition rate is small and the resolution difference is considerable, a 0.25second minimum length threshold should be preferred.

The average Vote-All grade reaches 75.5% (the Vote grade 91.8%), which means that caution should be taken if using the current program configuration for determining the participating instruments in MIP music for practical applications, as on average, the program fails to produce the complete list of participating instruments in 24.5% of the mixes.

Side Note – an experiment was performed where MIQR was used in order to remove 10% of the most "suspicious classifications" (corresponding to "results purging" depicted in shape #11 of Figure 1-A). This experiment has resulted in slightly over 1% increase in all recognition rates over the results summarized in Table 13-C. As this increase in recognition rates does not seem very significant, the full details of this experiment are omitted.

13.4 CONCLUSIONS

It was shown that both methods, Harmonic-Resynthesis and Source-Reduction, are relevant and produce much higher recognition rates than random instrument labeling.
In a way, with the Harmonic-Resynthesis method, the problem of AMIR in MIP music is reduced to correct multiple *f0*-estimation (including the harmonic series); this is because in HR there is no fundamental difference between instrument recognition with resynthesized notes from Solos and resynthesized notes from MIP music.

In the general case, the preferred recognition method depends on the precision qualities of the *f*0-estimation program. It was shown in (Livshin and Rodet 2006b) and Chapter 11, that when using the HR method, the instrument recognition rate depends almost entirely on estimation quality of the frequencies and amplitudes of the partials of the different notes. If the *f*0-estimation is very good, then the HR is a better method as it does not require filtering, is comparatively simple and straight-forward, and can lead to very high recognition rates.
Nevertheless, if the *f*0-estimation program is less precise, SR may be preferred as it allows the frequency estimation to be less exact as long as the estimated partials fall inside the correct spectral peaks.

CHAPTER 14 SUMMARY

The subject of this thesis is "Automatic Musical Instrument Recognition" (AMIR) which means, intuitively speaking, that given a musical recording, the computer attempts to identify which parts of the recording are performed by which musical instruments.
The thesis deals with the different stages of AMIR as presented in Figure 1-A.

Chapter Summary

Chapter 1 is an overview of the thesis. It presented a flowchart of the AMIR process and described its different stages. It is important to understand the flowchart in order to comprehend how each thesis chapter fits in the complete AMIR process. After discussing the AMIR process, a short overview of the goals in each thesis chapter was given.

Chapter 2 presented an introduction to AMIR. After providing a formal definition of the task, Section 2.1 explained why AMIR is an important research area and for which practical applications it is useful; to summarize, it could be said that AMIR is mostly applicable not as an end product, i.e., providing instrument recognition results to the user, but rather as a software module integrated into various applications and algorithms such as intelligent music searches over the Internet, automatic music transcription, software for composers and other applications. Section 2.2 presented some of the different challenges a researcher in AMIR has to deal with, including classification accuracy and generality, erroneous data, overlapping sounds in polyphonic music, pattern-recognition issues, etc.

Chapter 3 told the history of Instrument Recognition research which mainly evolved in three stages: AMIR in separate tones, Solos and Multi-Instrumental, Polyphonic (MIP) music. AMIR research was almost entirely limited to recognition of isolated note samples for about 30 years until around 1998, when several works on recognition of Solos, or as

frequently called, “musical phrases” have appeared, such as (Dubnov and Rodet 1998). Finally, with the paper of Eggink and Brown (2003), research on AMIR in MIP music began to gain popularity.

Chapter 4 mentioned briefly a few alternative taxonomies and classification structures to the ones used in this thesis - throughout most of the thesis the classification classes are concrete instrument names while the classification structure is a non-hierarchical, flat ‘all-vs.-all’.

Chapter 5 detailed the different sound sets used in the thesis: separate tone databases, authentic Solos, authentic Duos and Solo mixes.

In separate tone databases each sound sample contains an audio recording of a single note. Authentic Solos are recordings of music performed by a single instrument, whether it is a monophonic instrument, such as the flute, or polyphonic such as the piano. Authentic Duos are recordings of music where two instruments are playing concurrently.

Due to the fact that correctly labeled, recorded multi-instrumental music is virtually unavailable, authentic Solos were mixed together to create polyphonic “Solo mixtures” used for evaluation of AMIR of MIP music. This automatically tagged Solo-mix database is a new method for producing MIP instrument-tagged music for AMIR evaluation. This method has many advantages over music resynthesized according to MIDI files out of separate tone collections, as commonly used in other AMIR researches, such as (Kitahara et al. 2007), while having the disadvantage that the music it produces does not comply with any style-dependent composition rules.

Note that several rather large sound databases were created and gathered during this work. As such testing material is quite hard to find, obtaining and collecting these sounds for research purposes is a notable contribution in itself – most of all the large and diverse Solo database, which is available as part of the Scientific project MusicDiscover (MCI 2004).

Chapter 6 presented a list of the feature descriptors used for classification throughout this thesis along with relevant references. The feature descriptors are of different types – “Temporal features”, “Energy features”, “Spectral features”, “Harmonic Features” and “Perceptual features”. A full explanation of each feature is available in (Peeters 2004).

Chapter 7 discussed the feature selection and weighting techniques used in the thesis.
It explained the pros and cons of Linear Discriminant Analysis, which is the feature weighting technique used throughout this document.

The CFS feature selection algorithm was described and the new “GDE” feature selection algorithm was presented. While GDE, due to its LDA ‘engine’, selects the descriptors which produce the highest recognition rates when classification using LDA+KNN is performed, CFS selects a more “meaningful” list of non-dependent descriptors. As a non-

exponential feature selection algorithm which selects the optimal feature list is a theoretical impossibility, choosing the best feature selection algorithm depends on application.

Chapter 8 described the classification algorithms used in this thesis - Backpropagation Neural Network and KNN. Next, the "LDA+KNN" classification method was presented; it was demonstrated why "LDA+KNN" was selected for most of the classification tasks in the thesis.
The Backpropagation Neural Network is used only in Chapter 9 in comparison with "LDA+KNN" in order to demonstrate a specific point.

Chapter 9 has dealt with the issue of evaluation of AMIR techniques; it presented several evaluation methods including the new cross-validation techniques "Minus-1-DB", "Mutual Classification", "Minus-1-Instance" and "Minus-1-Solo". The chapter criticized the "Self-Classification" evaluation method, which used to be common before the publication of our (Livshin and Rodet 2003) paper, and used a single sound database for evaluating AMIR results. While a few researchers in AMIR did claim that it is important to use independent data in the Learning and Test sets, the importance of this data belonging to different sound databases was never directly evaluated before.

This chapter has filled a "theory hole" regarding the evaluation issue and proved the following claims:

- Evaluation results using a single sound database are not necessarily an indication of the generalization capabilities of the classification process and thus its suitability for realistic classification tasks, where the ultimate goal is to have a Concept Classifier - a classifier which could classify instrument sounds regardless of their specific recording conditions.
- Self-Classification results do not demonstrate the ability of a classifier which was trained on one database to classify new sounds, and thus its performance as a Concept Classifier. This also means that Self-Classification results do not demonstrate the diversity of the sound database being used.
- A feature selection algorithm might choose different features for classification of the same instrument types, depending on the sound database being used. This also means that evaluating features using a single database will not necessarily help to choose the right features for a Concept Classifier.
- Enriching the Learning database with diverse sound samples from other databases, helps the classifier to generalize better and makes it more suited for classification of new sounds.

To deal with the shortcomings of Self-Classification, first the "Minus-1-DB" evaluation method was introduced following by other multiple-source cross-evaluation methods. In "Minus-1-DB", the evaluation is performed using several sound databases, each classified by the rest joined together. Minus-1-DB results do provide an indication as to the generalization abilities of the evaluated classification algorithm and feature

descriptors, as the classification algorithm never learns the classified databases and is not adapted to the specific characteristics of their samples (as happens in Self-Classification). Although the generalization ability still depends on the sound databases being used, it could be reasonably assumed that the recording conditions of the samples in the various databases are different, which allows the evaluation of the generalization ability of the databases performed, while Self-Classification does not evaluate generalization at all and over-fits the Learning set.

Chapter 10 dealt with the issue of self-consistency of a Learning database. A sound database may contain samples which are badly recorded or mislabeled, causing classification errors. This chapter presented two new database purging algorithms – "MIQR" and "SCO" and compared them with the common Interquantile Range (IQR) algorithm. The performance was evaluated using a sound database contaminated with four different outlier types.

For non-labeled data, out of the three algorithms, the unsupervised IQR algorithm is the "natural way to go" as the other two algorithms require class information. When the goal was to be rid of contaminated data in labeled databases, MIQR achieved the best results. If the goal is to achieve the highest classification results, then wrapper-type methods, specifically tailored to the data and classification algorithms, such as SCO, may well be the answer.

Chapter 11 performed AMIR of separate tones from 10 instruments reaching a high recognition rate - 95.58% using strict Minus-1-Instance evaluation. At the time of publishing this result in our (Livshin and Rodet 2006b) paper, this recognition rate seemed to be the state of the art although it is somewhat hard to compare results between different AMIR papers as the evaluation methods and instruments vary considerably. Next, the chapter explored the instrument discrimination power of the harmonic series and the non-harmonic residuals. While it is common to treat the Harmonic Series as the main characteristic of the timbre of pitched musical instruments it seems that no direct experiments were performed to prove this assumption before it was done in our (Livshin and Rodet 2006a) paper.

As stated, this chapter has filled a "theory hole", checking the common assumption that distinguishing information is presented mainly/only in the harmonic series; in order to check it, using Additive Analysis/Synthesis, each sound sample was resynthesized using solely its Harmonic Series. These "Harmonic" samples are then subtracted from the original samples to retrieve the non-harmonic Residuals. AMIR is performed on the original samples, the Resynthesized ones and the Residuals and the results are compared and discussed. Using CFS feature selection, the best 10 feature descriptors for instrument recognition are presented for the Original, Harmonic and Residual sound sets.

The chapter shows that using only information present in the Harmonic Series of the signal is enough for achieving a high average musical instrument recognition rate –

91.51% for 10 instruments using Minus-1-Instance evaluation. This is only 4.07% less than the recognition rate obtained by using the complete, Original signals.

On the other hand, it was also shown that there is considerable distinguishing instrument information present in the non-harmonic Residuals, which by themselves produced an average instrument recognition rate of 81.99%. It was revealed that the information present in the non-harmonic Residuals is not completely redundant to the information present in the Harmonic Series as although the average recognition rate of the Harmonic signals is high, some of the instruments have "suffered" noticeably from removing the non-harmonic Residuals, especially the trumpet and violin pizzicato. In addition, it was shown that the 10 best feature descriptors selected for the Original sample set differ from the ones selected for the Harmonic samples. These results show that the sound of pitched musical instruments should not be treated as containing only the Harmonic Series, although most of the energy and distinguishing instrument information of the signal is indeed present in the Harmonic Series.

One conclusion from this chapter is that because it was shown that using only the harmonic series does not considerably lower the average instrument recognition rate, then instrument recognition in MIP music could be done with rather high results without performing full source-separation; Using results obtained from multiple-*f0* estimation algorithms, estimated harmonic partials could be used solely to classify musical instruments without losing too much distinguishing information. This is exactly what the "Harmonic-Resynthesis" method in Chapter 13 is based upon.

Chapter 12 has dealt with Instrument recognition in Solo performances (monophonic or polyphonic musical phrases performed by a single instrument) which is different and more complicated than dealing with separate tone databases, as the time evolution of each sound (attack, decay, sustain, release) is not well defined, the notes are not separated, there are superpositions of concurrent sounds and room echo, different combinations of playing techniques, etc.
AMIR was performed on a large Solo collection of 7 instruments – bassoon, clarinet, flute, piano, guitar, cello and violin. First the Solos were classified using the Full Feature Descriptor set producing a Minus-1-Solo Instrument Grade of 88.13%, and next a specifically reduced set of descriptors was used to create an actual realtime AMIR of Solos application (while the Solo plays/records, each one-second piece is instrument-recognized) with only a slight decrease in recognition rate, producing a Minus-1-Solo recognition rate of 85.24%. Note that at the time of publishing these results (Livshin and Rodet 2004a), both the results for offline and realtime Solo recognition were the state of the art and notably higher than other Solo recognition results obtained before.

Chapter 13 dealt with instrument recognition in multi-instrumental, polyphonic (MIP) music. It presented three techniques – Naïve Solo Classifier, Source-Reduction (SR) and Harmonic-Resynthesis (HR). The Naïve Classifier only classifies each signal frame one-dimensionally, as a single instrument, while SR and HR return polyphonic AMIR results. Both HR and SR utilize a multiple-*f0* estimation program and use Solos in the Learning

set. SR attempts to achieve single-instrument notes by filtering out of every section of MIP music where a note is detected all the other sounds except this note's harmonics, while Harmonic-Resynthesis resynthesizes, using Additive Synthesis, single notes out of harmonic series obtained from *f0*-estimation information. In a way, with the HR method, the problem of AMIR in MIP music is reduced to correct multiple *f0-estimation*; this is because in HR, there is no theoretical difference between instrument recognition with resynthesized notes from Solos and resynthesized notes from MIP music.

As both HR and SR systems in this thesis use estimated harmonic-series computed by a multiple-*f0* estimation program, they can be directly applied to authentic musical recordings, unlike (Kitahara et al. 2007) and (Eggink and Brown 2003) for example, which require manual *f0*-information to be supplied. Both SR and HR perform AMIR of each note separately (using detected note boundaries), thus, unlike (Essid, Richard and David 2006) for example, which performs AMIR of all the polyphony at once, SR and HR can be directly utilized for automatic transcription, ordering notes into staves according to the playing instrument. Both SR and HR do not use heuristics based on musical context, such as proposed by (Kitahara et al. 2007) or assumed in (Essid, Richard and David 2006), and thus can be applied with similar success to any musical genre, including classical music, modern atonal music and even random sound mixtures.

These techniques were evaluated on authentic recordings of Duos with 7 musical instruments and on mixtures of Solo recordings of 5 different instruments with up to 5 instruments playing concurrently. The chapter showed that both MIP methods, HR and SR, produce much higher recognition rates than random instrument labeling. The Mutual AMIR Grade for 18 authentic Duos with 7 musical instruments is over 80% with both techniques, while with HR, an instrument detection level of 100% is reached. With artificial mixes of Authentic Solos of 5 instruments, an average, precise recognition rate of 56.9% was reached for 2 – 5 simultaneously playing instruments.

Unfortunately the recognition rates cannot be currently informatively compared among AMIR MIP papers as the evaluation methods as well as the evaluation data are very different, especially due to lack of commonly available, tagged MIP music. For a discussion of the need for precise evaluation, see Section 15.4.

Chapter 14 presents a summary of the thesis and the various contributions in each chapter.

Chapter 15 recommends further work and research in order to improve AMIR. Utilizing composition rules, creating specialized feature descriptors, precise evaluation, human integration into the AMIR process and moving from theoretical research into practical applications are the various issues discussed in this chapter.

Appendix A lists acronyms and abbreviations used in the thesis.

Appendix B contains my 6 published papers.

Summarizing Final Words:

Automatic Musical Instrument Recognition is a wide, complex multi-disciplinary field. It involves knowledge from various research areas including musical theory, audio physics, musical instrument theory, signal processing and pattern recognition.

While this work describes the different stages and components required to perform instrument recognition in separate tones, Solos and MIP music, research on AMIR is far from being completed. Creating a system which could recognize each instrument participating in a classical concert, for example, at least at the level a human listener can, is yet far from possible.

CHAPTER 15 FUTURE WORK

The problem of performing AMIR in MIP music in a practical way for real-world applications is yet rather far from being solved. Further work should be done on the subject in order to improve recognition rates and other aspects.

15.1 USING COMPOSITION RULES

Instrument recognition as performed in this work did not assume any specific musical structure and treated music as random sequences of sounds performed by Musical Instruments. Moreover, evaluation of MIP music in Chapter 13 using random mixtures of Solos has actually *created* "music" to which virtually no composition rules apply, thus rendering usage of any musical knowledge, such as voice leading rules, common harmony, non-crossing of note streams, etc., practically impossible.

Nevertheless, until perfect recognition could be obtained by musicologically-ignorant Signal Processing and Pattern-Recognition algorithms, taking into consideration the musical style and thus statistically common composition rules from the authentic musical genre, the AMIR task could be simplified and improved. For example, Essid, Richard and David (2006) write that their AMIR system is suited specifically for Jazz quartets. Kitahara et. al (2007) have assumed that note streams of different instruments do not cross each other as indeed rarely happens in classical music arrangements. This assumption allowed Kitahara to assume that sudden changes in instruments in the middle of note streams are classification errors, and "correct" such "misclassifications" by changing the classification of these notes to the same instrument as other notes in their vicinity.

A musical genre recognition system integrated with an AMIR system could allow specifying a set of musical rules for improving the instrument recognition. Otherwise, this input may be provided by a user depending upon application – for example if large collections of classical music are to be processed for AMIR, the user may set the "Classical Music genre" parameter.

15.2 FEATURE DESCRIPTORS

15.2.1 UTILIZING INFORMATION IN THE NON-HARMONIC RESIDUALS

Chapter 11 has shown that the Residual part of pitched musical instruments contains much instrument distinguishing information. When dealing with Solos or single tone databases, Instrument recognition of pitched instruments may be improved by splitting the classified sounds into harmonic and non-harmonic components and then computing special feature descriptors on the non-harmonic Residuals in addition to the feature descriptors computed on the original signals. The splitting of the signal makes it easier to deal with the non-harmonic Residuals due to their relatively low energy levels.

Note that using the current Full Feature descriptor set (see Chapter 6), experiments where the descriptors of the Original samples, the Harmonic and the Residual sets were all used together have not yielded higher recognition rates than when using the Feature descriptor set of the Original samples by itself. The descriptors for utilizing information from the Residuals should therefore be specifically designed for this purpose and take into account the special nature of such "noises".

15.2.2 HEURISTIC DESCRIPTORS

Tailored feature descriptors designed to capture certain attributes specific to one or two instruments could be added, based on knowledge of very particular instrument characteristics (similar to the existing feature descriptor which calculates odd-to-even harmonic ratio in order to detect the clarinet). See (Fletcher and Rossing 1998) for example, for physical properties of different musical instruments and how these affect the sound.

On the opposite side, attempts have been made to create systems for automatic construction of completely non-heuristic descriptors out of prototype signal-processing building blocks in order to distinguish best among given musical sample groups. Such is the Extractor Discovery System (EDS) (Pachet and Zils 2004) for example.

15.2.3 MODELLING SIGNAL EVOLUTION

In the current feature set the evolution over time of the signal is modeled by the Temporal Feature Descriptors (see Chapter 6) and by computing the frame-based features on each STFT frame separately and then using the mean and standard deviation of these frames. Much more precise tracking of the sound evolution over time could be achieved using Hidden Markov Models (HMM) (Baum and Petrie 1966; Eichner, Wolff and Hoffman 2006) of the feature descriptors over the STFT frames, Dynamic Time Warping (Ratanamahatana 2005) and other time-series modeling techniques.

15.3 PRACTICAL APPLICATIONS

The current MIP AMIR systems as defined in Chapter 13 are not ready yet for many AMIR practical applications, such as labeling music on the web, and besides improving the recognition rates, should be scaled and enhanced:

15.3.1 INCREASING THE NUMBER OF INSTRUMENTS

Currently the systems in Chapter 13 recognize only seven different instruments in MIP music. Besides adding specific instruments, compound instruments could be added, where the sounds may consist of several instruments playing concurrently, such as a string section, where it is very hard to distinguish each violin by itself while the whole section performs together as a single instrument. Databases of instruments and sound sets typical of different genres could be used depending on application, such as rock music, pop, jazz, etc.

In order to increase the number of recognized instruments using the HR and SR techniques, the Learning database should be enriched by Solos of new instruments. This was not done in this work as Solos are very hard to find, especially as it was insisted here that they come from different recording sources in order to allow informative evaluation (see Chapter 9). This demand may be slackened when enriching the Learning database in practical applications, as full Learning database evaluation is not the goal there.

Note that non-pitched instruments such as many drums and percussion instruments should be dealt with caution as some techniques presented in this thesis, such as Harmonic-Resynthesis (Section 13.1.3), do not apply.

15.3.2 SPEED IMPROVEMENT

When a large volume of musical pieces needs to be scanned, speed considerations should be applied. Except Section 13.4 which dealt with Solo recognition in realtime, no speed improvement attempts have been made in this thesis. Currently, on a Pentium 1.6 MHZ

computer it may take around four minutes to perform AMIR on a one-minute musical piece with four voices playing concurrently, not counting the multiple-*f0* estimation part, which may require much longer than that. Current Matlab routines should be translated to C or Assembly language and some of the algorithms may be optimized for lower complexity.

15.4 PRECISE EVALUATION

In order to perform exact AMIR evaluation, authentic recordings of MIP music along with its precise transcription are required. Unfortunately, such labeled recordings are practically impossible to obtain unless performed on MIDI instruments.

Different researchers use various techniques in order to partially circumvent this problem: Music resynthesized from MIDI files, manually labeled musical pieces or as used in this thesis, artificial mixtures of authentic Solos.
While Solo-Mixtures seem to have various advantages over resynthesized MIDI files such as including instrument articulations, realistic sounds, etc. (see Section 5.4 for full details), the Solo sounds do not influence each other and do not create real, common room echo. Moreover, as noted above, musical composition rules do not apply to the Solo-mixtures as the mixed Solos have no relevancy to each other.

The precise-evaluation problem is rather hard to tackle. (Vincent et al 2007) have recorded separately, in the same room, instruments playing their different parts in an MIP composition and then created an authentically sounding mix by playing the parts together, in the same room, and recording "from the air" with differently positioned microphones; this method does not seem to suffer from the weaknesses mentioned above. While this seems like a rather good solution, unfortunately, it was done only on a very small scale (a single three-voice MIP piece) and thus wide MIP evaluation database is still not available.

There are some methods which allow synthesis of semi-natural music with articulations, such as Digital Waveguide Synthesis (Smith 1992) which uses computational physical models, however, preparing naturally sounding music with such methods without using specifically designed instruments (which obviously miss the whole point of automatic preparation of an authentic evaluation database), requires too much definition work. Another solution could be to perform Score Alignment (see (Rodet, Escribe and Durigon 2004) and (Cont 2006) for example) in order to align sheet music with audio recordings thus gaining an approximate positioning of each transcribed note (e.g., MIDI files) inside the recordings.

To create an evaluation database, one of the best direct solutions, in my opinion, is while an orchestra (or other musical arrangement) plays, besides recording and mixing the

whole music as usual, to record in addition each instrument separately[19]. Afterwards, *f0*-estimation could be performed on each recorded stream, which is relatively easy with monophonic music, and the results should be stored in symbolic format along with the audio recordings of the full orchestra and each instrument separately. This database could be then used by researchers from many MIR fields, including AMIR, Score Alignment, Source Separation, etc. I believe such a database would be very useful and save much time spent on dealing with certain evaluation problems. Note that polyphonic instruments, such as the harpsichord or the piano should be treated differently.

When recording symphonic orchestras, an interesting experiment may be to record the whole concert as usual (without recording each instrument separately) while sampling the movements of the conductor, afterwards, align the music to the musical partitura according to the rhythm of the conductor's strokes and her other movements. This may be also attempted with recorded videos of musical concerts. While this task may prove rather difficult, I believe it is an interesting research idea.

While theoretically it may be possible to create special musical arrangements of musicians all playing different MIDI instruments, while instrument-specific articulations may be present (depending upon the MIDI instrument sophistication), yet the resulting music, at least for now, is going to differ significantly from music played on musical instruments which produce the sound acoustically (which are still, electrically amplified or not, the most commonly used instruments almost in every musical genre and therefore desirable for AMIR)

15.5 HUMAN INTEGRATION

For different practical applications, such as instrument-indexing large musical collections on the Internet, it may be worthwhile to make the human user "part" of the AMIR system by asking her some questions about the analyzed musical pieces during the AMIR process, which can improve the performance of the AMIR system.

Examples:

- In many cases, the user may know which instruments are playing in a musical piece. Printed labels, for example, usually contain some information regarding the musical arrangement. When given the instrument names, the AMIR program can index the piece with fewer errors using a reduced sound database containing only the participating instruments.
- The user may supply the musical genre and allow the AMIR program to assume appropriate composition rules as explained in Section 15.1 above.

[19] This is being done at Ircam with very directive and close microphones; evaluation databases are not available yet.

- An AMIR confidence level could be defined, thus helping to recognize samples which the AMIR system has problems handling and accordingly update the database or disregard these samples. Some examples for confidence levels:
    - Using several classification algorithms to classify the same samples and giving a low confidence mark to sounds which are differently classified by different algorithms.
    - Using database purging methods such as MIQR for detecting outliers, before or after the classification (see Chapter 10).
    - The classification algorithm may directly supply the confidence level. For example, when using BP Neural Networks or similar for classification (see Section 8.1.1) the confidence level could be defined as the difference between the highest network output and the others; the smaller the difference, the smaller the confidence. When using KNN, if the number of neighbors supporting the winning class is very close to the number of other classes, the confidence is small.

  If a large group of similar samples gains a low confidence level, the user could be prompted to specify which instrument these samples belong to. This could be instruments already present in the Learning database or completely new instruments. Afterwards, the Learning database can be enriched with the new samples accordingly.

Appendix A - Abbreviations and Acronyms

AMIR	Automatic Musical Instrument Recognition
BP	Backpropagation neural network
BP80	A BP with 80 neurons in the hidden level
CFS	Correlation-based Feature Selection
DB	Database
DFT	Discrete Fourier Transform
f0	Fundamental frequency
FFT	Fast Fourier Transform (Cooley and Tukey 1965)
GDE	Gradual Descriptor Elimination algorithm
HR	Harmonic-Resynthesis algorithm
Hz	Hertz
IQR	InterQuantile Range
LDA	Linear Discriminant Analysis
LOO	Leave-One-Out Cross Validation method
KNN	K-Nearest Neighbors classifier
MFCC	Mel Frequency Cepstral Coefficients
MIP	Multi-Instrumental Polyphonic
MIQR	Modified InterQuantile Range
MIR	Music Information Retrieval
PCA	Principal Component Analysis
SOL	Studio en Ligne IRCAM sound database
SCO	Self-Classification Outlier removal method
SOM	Self Organizing Map
SR	Source-Reduction algorithm
STFT	Short Time Fourier Transform (Allen 1977; Allen and Rabiner 1977)

Appendix B - Published Papers

(Papers are present only in the “library CD” version of the thesis)

This appendix contains six published papers of which I am the main author:
(Livshin, Peeters and Rodet 2003), (Livshin and Rodet 2003), (Livshin and Rodet 2004a), (Livshin and Rodet 2004b), (Livshin and Rodet 2006a) and (Livshin and Rodet 2006b).

Most of this material is presented in the thesis as well.

To people who are interested in reading the papers, I recommend reading only four of them - (Livshin, Peeters and Rodet 2003), (Livshin and Rodet 2003), (Livshin and Rodet 2004b) and (Livshin and Rodet 2006b).

(Livshin and Rodet 2004a) and (Livshin and Rodet 2006a) are expanded and updated in my later papers and therefore do not have much individual contribution.

References

Adam, A., Rivlin, E., Shimshoni, I., 2001. “ROR: Rejection of Outliers by Rotations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 23, No 1, January 2001.

Agostini, G. , Longari, M., Pollastri, E. October 2001. "Musical Instrument Timbres Classification with spectral Features", *International Workshop on Multimedia Signal Processing*, IEEE Signal Processing Society, Cannes, France, 3-5.
URL: http://www.guilio.com/pdf/IEEE Signal Processing Society - Cannes.pdf

Aleksander, I., Morton, H., 1990. *An Introduction to Neural Computing*, Chapman and Hall.

Allen J. B. 1977. "Short Time Spectral Analysis, Synthesis and Modification by Discrete Fourier Transform." IEEE Transaction on Acoustics, Speech and Signal Processing 25(3):235-238.

Allen J. B., and L. Rabiner 1977. "A Unified Approach to Short-Time Fourier Analysis and Synthesis." Proceedings of the IEEE vol. 65(11):1558-1564.

Aures, W., 1984. “Berechnungsverfahren für den Wohlklang beliebiger Schallsignale, ein Beitrag zur Gehörbezogenen Schallanalyse”, Dissertation am Lehrstuhl für Elektroakustik der Technischen Universität München, Germany.

Bailer-Jones, C. A. L., Bhadeshia H. K. D. H., MacKay, D. J. C., 1999. “Gaussian Process Modelling of Austenite Formation in Steel”, *Journal of Materials, Science and Technology*. 15, 287–94.

Ballet, G. 1998. "Studio On Ligne". URL: http://forumnet.ircam.fr/402.html

Beauchamp, J. W. 1982. "Synthesis by Spectral Amplitude and 'Brightness' Matching Analyzed Musical Sounds." Journal of Audio Engineering Society 30(6): 396-406.

Baum, L. E., Petrie, T., 1966. “Statistic inference for probabilistic functions of finite state Markov chains”, Ann. Math. Stat., vol. 37, pp. 1554-1563.

Boser, B. E., Guyon, I. M., Vapnik , V. N., 1992. “A training algorithm for optimal margin classifiers”, *Proc. 5th Annual ACM Workshop on COLT*, pp 144-152.

Blum, A., Langley, P., 1997. “Selection of relevant features and examples in machine learning,” Artificial Intelligence, 97(1-2):245–271.

Bogert B. P., Healy M. J. R., Tukey J. W. 1963. "The quefrency alanysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-cepstrum, and saphe cracking". Proceedings of the Symposium on Time Series Analysis (M. Rosenblatt, Ed) Chapter 15, 209-243. New York: Wiley.

Brooks, M. 1998;“VOICEBOX: Speech Processing Toolbox for MATLAB”. URL: http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

Brown, J.C., 1998. ``Musical Instrument Identification using Autocorrelation Coefficients", Proc. *International Symposium on Musical Acoustics*.

Brown, J. C., Houix O., McAdams, O. 2001. “Feature dependence in the automatic identification of musical woodwind instruments,” *Journal of the Acoustical Society of America*, Vol. 109, No. 3, pp 1064-1072.

Conover, W. J., 1980. “Practical Nonparametric Statistics”. Wiley.

Cont, A., 2006. “Realtime Audio to Score Alignment for Polyphonic Music Instruments, using Sparse Non-Negative Constraints and Hierarchical HMMS”, *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006).*

Cooley, J., Tukey J., 1965. "An Algorithm for the Machine Calculation of Complex Fourier series." Mathematics of Computation.

Cover, T. M., Hart, P. 1967. "Nearest neighbor pattern classification". *IEEE Transactions on Information Theory* 13 (1): 21–7.

Cover, T. M., Van Campenhout, J., 1977. “On the Possible Orderings in the Measurement Selection Problem,” *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-7, No. 9.

Davis, S.B., Mermelstein, P., “Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences”, IEEE Trans. on Acoustic, Speech and Signal Processing, 28(4):357–366, 1980.

Depalle, P., Garcia, G., Rodet, X. 1993. “Tracking of partials for additive sound synthesis using hiddenMarkov models”, in *ICASSP 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*.

De Poli, G., Prandoni, P. 1997. “Sonological Models for Timbre Characterization”. *Journal of New Music Research*, Vol 26 (1997), pp. 170-197.

Doval, B., Rodet, X. 1993.” Fundamental frequency estimation and tracking using maximum likelihood harmonic matching and HMMs”. In ICASSP 1993 IEEE *International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 221--224.

Draper, N. 1999. "Statistics 201 lectures". *Online statistics lectures by the Statistics Department of the Wisconsin Madison University*. URL: www.stat.wisc.edu/~jyan/st201/pdf/dis10.pdf

Dubnov, S., Rodet, X. 1998. “Timbre recognition with combined stationary and temporal features.” *In Proc. of the 1998 International Computer Music Conference (ICMC98)*.

Duda, R. O., Hart, P. E. “Pattern Classification and Scene Analysis”, Wiley-Interscience 1973.

Dwyer, R. F., 1983. “Detection of non-Gaussian signals by frequency domain kurtosis estimation,” Int. Conf. on Acoustic, Speech, and Signal Processing, Boston 1983, pp. 607–610.

Eagleson, H., Eagleson W. 1947. "Identification of Musical Instruments when Heard Directly Over a Public Address System." *Journal of Acoustical Society of America* 19(2): 338-342.

Eggink, J., Brown, G. J., 2003. “A MISSING FEATURE APPROACH TO INSTRUMENT IDENTIFICATION IN POLYPHONIC MUSIC,” in *ICASSP 2003 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing Proc.*, 2004.

Eggink, J., Brown, G. J., 2004. “Instrument recognition in accompanied sonatas and concertos,” in *ICASSP 2004 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing Proc.*, 2004, pp. 217-220.

Eichner, M., Wolff, R., Hoffman, R., 2006. “Instrument classification using Hidden Markov Models”, in *Proc. of International Symposium on Music Information Retrieval (ISMIR 2006)*.

Elliot, C. 1975. "Attacks and Releases as Factors in Instrument Identification." *Journal of Research in Music Education* 23: 35-40.

Eronen A. 2001. "Comparison of features for musical instrument recognition". *In Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA 2001*.

Essid, S., Richard, G., David, B., 2006. “Instrument recognition in polyphonic music based on automatic taxonomies,” *IEEE Transactions on Speech and Audio Processing*, Jan. 2006.

Fletcher, N., Rossing, T. D., 1998. *The Physics of Musical Instruments, second edition*. Springer.

Fraser, A., Fujinaga, I., 1999. "Toward real-time recognition of acoustic musical instruments", *Proc. of the ICMC*. 175-177, 1999. URL: citeseer.nj.nec.com/fraser99toward.html

François, J., Grandvalet, Y., Denoeux, T., Roger, J. M., 2003. “Resample and combine: an approach to improving uncertainty representation in evidential pattern classification,” *Information Fusion* 4(2): 75-85 (2003).

Fix, E., Hodges, J. L., 1951. “Discriminatory analysis—nonparametric discrimination: Consistency properties,” Technical Report Number 4, Project Number 21-49-004, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

Fletcher, H., Blackham, E. D., Stratton, R. 1962. “Quality of Piano Tones”. *J. Acoust. Soc. Am.,* **34**(6), 749-761.

Freedman, M. D. 1967. “Analysis of musical instrument tones”. *J. Acoust. Soc. Am.,* **41**, 793-806.

Fritts, L., 1997. "University of Iowa Musical Instrument Samples". URL: http://theremin.music.uiowa.edu/MIS.html

Fujinaga, I. 1998. “Machine recognition of timbre using steady-state tone of acoustic instruments”. *Proc. ICMC 98*, pages 207-210.

Gibbs, M. N., 1998. “Bayesian Gaussian processes for regression and classification”, *PhD Thesis,* University of Cambridge.

Good, M., 2001. “MusicXML: an internet-friendly format for sheet music,” in Proceedings of the XML Conference & Exposition, Orlando, USA, December 2001.

Grey, J. M. 1977. “Multidimensional perceptual scaling of musical timbres,” Journal of the Acoustical Society of America, 61, (5), 1270-1277.

Grey ,J. M., Gordon, J. W. 1978. "Perceptual Effects of Spectral Modifications on Musical Timbres." Journal of Acoustical Society of America 63(5): 1493-1500

Guyon, I., Elisseeff, A., 2003. “An Introduction To Variable and Feature Selection,” Journal of Machine Learning Research 3 (2003), pp. 1157-1182.

Hall, M. A., 1998. *Correlation-based feature selection machine learning*, Ph.D. Thesis, Department of Computer Science, University of Waikato, Hamilton, New Zealand.

Herre, J., Allamanche, E., & Hellmuth, O., 2001. "Robust matching of audio signals using spectral flatness features." 2001 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'01), IEEE.

Herrera, P., Peeters, G., Dubnov, S. 2003. "Automatic Classification of Musical Sounds," *Journal of New Musical Research,* Vol. 32, No. 1, pp 3-21.

James, M. 1985. Classification Algorithms. Wiley&Sons.

Jayant N.S., Noll, P. 1984. "Digital Coding of Waveforms: Principles and Applications to Speech and Video", Prentice-Hall, 1984

Judge, G. G., R. C. Hill, W. E. Griffiths, H. Lutkepohl, and T.-C. Lee. 1988. "Introduction to the Theory and Practice of Econometrics", Wiley.

Kaminskyj, I., Materka, A. 1995. "Automatic source identification of monophonic musical instrument sounds". *Proceedings of the IEEE International Conference On Neural Networks*, 1, (pp. 189-194).

Kaminskyj, I. 2001. "Multi-feature Musical Instrument Sound Classifier". *Australasian Computer Music Conference*.

Kitahara, T., Goto, M., Okuno, H. G., 2004. "Category-Level Identification of Non-Registered Musical Instrument Sounds," In *ICASSP 2004 IEEE Int. Conf. on Acoustics, Speech and Signal Processing Proc.*.

Kitahara, T., Goto, M., Komatani, K., Ogata, T., Okuno, H. G., 2007. "Instrument Identification in PolyphonicMusic: Feature Weighting to Minimize Influence of Sound Overlaps", *EURASIP Journal on Advances in Signal Processing,* Vol. 2007, Article ID 51979.

Kofidis, E., Theodoridis, S., Kotropoulos, C., Pitas, I. 1996. "Nonlinear adaptive filters for speckle suppression in ultrasonic images", *Signal Processing*, 52(3):357-372. URL: citeseer.nj.nec.com/kofidis96nonlinear.html

Kohavi, R. 1995. "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection." *IJCAI* 1137-1145.

Kohavi, R., John, G., 1997. "Wrappers for feature selection," Artificial Intelligence, 97(1-2):273–324.

Kononenko, I. 1994. "Estimating Attributes: Analysis and Extensions of RELIEF". *European Conference on Machine Learning*, 1994. Catania.

Krimphoff, J., McAdams, S., Winsberg, S. 1994. "Caractérisation du timbre des sons complexes. II: Analyses acoustiques et quantification psychophysique". *Journal de Physique*, 4, 625-628.

Krumhansl, C. L. 1989. "Why is musical timbre so hard to understand?" In S. Nielzenand & O. Olsson (Eds.), *Structure and perception of electroacoustic sound and music (pp. 43-53)*. Amsterdam: Elsevier.

Laurikalla, J. , Juhola, M., Kentala, E. 2000. "Informal identification of outliers in medical data." *5th International Workshop on Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP-2000)*. URL: citeseer.nj.nec.com/327617.html

Livshin, A., Dubnov, S., 1998. "netSound – Categorization of musical instruments using neural networks", *unpublished laboratory report*.

Livshin, A., Peeters, G., Rodet, X. 2003. "Studies and Improvements in Automatic Classification of Musical Sound Samples," In *Proceedings of the International Computer Music Conference (ICMC'03)*.

Livshin, A., Rodet, X. 2003. "The Importance of Cross Database Evaluation in Musical Instrument Sound Classification: a critical approach.", In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR'03)*.

Livshin, A., Rodet, X., 2004a. "Instrument Recognition Beyond Separate Notes – Indexing Continuous Recordings," In *Proceedings of the International Computer Music Conference (ICMC'04)*.

Livshin, A., Rodet, X., 2004b. "Musical Instrument Identification in Continuous Recordings," in *DAFx 2004 7'th international conference on Digital Audio Effects Proc.,* pp. 222-227.

Livshin, A., Rodet, X., 2006a. "The importance of the non-harmonic residual for automatic musical instrument recognition of pitched instruments," in *Proceedings of the Audio Engineering Society Convention (AES 2006*).

Livshin, A., Rodet, X., 2006b. "The Significance of the Non-Harmonic "Noise" Versus the Harmonic Series for Musical Instrument Recognition," In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR'06)*.

Livshin, A., Yeh, C., Röbel, A., Rodet, X. 2005. "A waveform to MIDI conversion System," *unpublished report*.

Marques, J., Moreno, P. J. 1999. "A study of musical instrument classification using Gaussian mixture models and support vector machines," *Cambridge Research Laboratory Technical Report Series,* CRL/4.

Martin, K. D., Y. E. Kim. 1998. "Musical instrument identification: A pattern-recognition approach." In *Proc. 136 th meeting of the Acoustical Society of America*.
URL: citeseer.nj.nec.com/martin98musical.html

Martin, K. 1999. "Sound-source recognition: A theory and computational model," *PhD Thesis,* MIT.

Matlab R13, Neural Network Toolbox.

McAdams, S., Winsberg, S., de Soete, G., & Krimphoff, J. 1995. "Perceptual scaling of synthesized musical timbres: common dimensions, specificities, and latent subject classes." *Psychological Research*, 58, 177-192.

MCI 2004. "Action Concertée Incitative, MASSES de DONNEES, Descriptif complet du projet", *classified document*. Project URL: http://recherche.ircam.fr/equipes/analyse-synthese/musicdiscover

McLachlan, G. J. 1992. Book title: "Discriminant Analysis and Statistical Pattern Recognition." New York, NY: Wiley Interscience.

Moore B.C.J., Glasberg B.R., Baer T. 1997. "A model for the prediction of thresholds, loudness, and partial loudness.", J. Audio Eng. Soc. 45, 224-240.

MPEG 1992. Coding of moving pictures and associated audio for digital storage media at up to 1.5 Mbit/s, part 3: Audio. International Standard IS 11172-3, ISO/IEC JTC1/SC29 WG11, 1992.

MPEG-7 2004. "MPEG-7 Overview (version 10)", URL: http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm

Opoloko, F., Wapnick, J., 1991. "McGill University Master Samples" CD-ROM for SampleCell VOLUME 1. 1991.

Osborne, J. W., Overbay A., 2004. "The power of outliers (and why researchers should always check for them)". *Practical Assessment, Research & Evaluation*, 9(6), 2004.

Pachet, F., Zils, A, 2004. "Automatic Extraction of Music Descriptors from Acoustic Signals", *Proc. of Fifth International Conference on Music Information Retrieval (ISMIR04)*.

Pearson, K. 1901. "On Lines and Planes of Closest Fit to Systems of Points in Space". *Philosophical Magazine* 2 (6): 559–572.

Peeters, G., McAdams, S., & Herrera, P. 2000. "Instrument sound description in the context of MPEG-7". *Proceedings of the 2000 International Computer Music Conference*. San Francisco, CA: International Computer Music Association.

Peeters, G., Rodet, X. 2002. "Automatically selecting signal descriptors for Sound Classification." in *Proceedings of the International Computer Music Conference (ICMC 02)*. URL: www.ircam.fr/equipes/analyse-synthese/peeters/ARTICLES/Peeters_2002_ICMC_SoundClassification.pdf

Peeters, G., Rodet, X. 2003. "Hierarchical Gaussian Tree with Inertia Ratio Maximization for the Classification of Large Musical Instrument Databases", in *DAFx 2003 6 th international conference on Digital Audio Effects*.

Peeters, G. 2004. "A large set of audio features for sound description (similarity and classification) in the CUIDADO project". *CUIDADO I.S.T. Project Report 2004*. URL: http://www.ircam.fr/anasyn/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf

Pollard, H., Jansson, E., 1982. "A tristimulus method for the specification of musical timbre", *Acustica*, 51: 162-71.

Powell, M. J. D., 1977. "Restart procedures for the conjugate gradient method," *Mathematical Programming*, vol. 12, pp. 241-254.

Rabiner, J., 1993. "Fundamentals of speech recognition". Prentice-Hall 1993.

Ratanamahatana, C., 2005. "Improving Efficiency and Effectiveness of Dynamic Time Warping in Large Time Series Databases", *PhD Thesis,* University of California, Riverside.

Risset, J. C., 1985. "Computer Music Experiments, 1964-…," *Computer Music Journal*, vol. 9, no. 1, pp. 11-18.

Röbel, A., 2006. "Estimation of partial parameters for non stationary sinusoids," *Proc. Int. Computer Music Conference (ICMC'06).*

Rodet, X. "THE ADDITIVE ANALYSIS-SYNTHESIS PACKAGE", URL:http://recherche.ircam.fr/equipes/analyse-synthese/DOCUMENTATIONS/additive/index-e.html

References

Rodet, X. 1997. "Musical Sound Signals Analysis/Synthesis: Sinusoidal+Residual and Elementary Waveform Models", *Applied Signal Processing* (1997) 4:131-141.

Rodet, X., Escribe, J., Durigon S. 2004. "Improving score to audio alignment: Percussion alignment and Precise Onset Estimation" , In *Proceedings of the International Computer Music Conference (ICMC 04)*.

Sabin, T. J., Bailer-Jones, C.A.L. 2000. "Accelerated learning using Gaussian process models to predict static recrystalisation in an Al-Mg alloy. Modelling and Simulation in Materials Science and Engineering, 8(5):687-706. URL: www.mpia-hd.mpg.de/homes/calj/gpcryst.pdf

Saldanha, E., Corso, J. 1964. "Timbre cues and the identification of musical instruments." *Journal of Acoustical Society of America* 36: 2021-2026.

Serra, X., Smith, J. O., 1990. "Spectral Modeling Synthesis: A Sound Analysis Synthesis System Based on a Deterministic plus Stochastic Decomposition," *Computer Music Journal*, vol. 14, no. 4, pp. 12-24.

Smith, G., Murase, H. Kashino, K. 1998. "Quick audio retrieval using active search". *In Proceedings of the 1998 ICASSP*.

Smith, J., 1992. "Physical Modeling using Digital Waveguides", *Computer Music Journal*, vol. 16, no. 4, pp. 74-91.

Soong, R., 1988. "On the Use of Instantaneous and Transitional Spectral Information in Speaker Recognition". *IEEE Trans. Acoustics, Speech and Signal Proc,* Vol. 36, No. 6, pp. 871-879.

STA. 2001. "STA 6938 Range and Distribution Normalization." *Online Data-Mining lectures by the Department of Statistics at the University of Central Florida (UCF)*.
URL: dms.stat.ucf.edu/sta6938notes/Lecture/Lecture2/ STA6938_Lecture2.pdf

Stevens, S.S., Volkmann, J., Newman, E. B., 1937. "A Scale for the Measurement of the Psychological Magnitude Pitch", *The Journal of the Acoustical Society of America,* -- January 1937, vol. 8, Issue 3, pp. 185-190 .

Terhardt, E. 1974. "On the perception of periodic sound fluctuations (roughness)," Acustica 30(4): 201-213.

Vincent, E. 2006. "Musical Source Separation Using Time-Frequency Source Priors," *IEEE Trans. on Audio, Speech and Language Processing,* 14(1), pp. 91-98

Vincent, E., Rodet, X., 2004. "Instrument identification in solo and ensemble music using Independent Subspace Analysis," in *ISMIR 2004 Fifth Int. Conf. on Music Inf. Retr. Proc*.

Vincent, E., Sawada, H., Bofill, P., Makino, S., Rosca J. P. 2007. "First stereo audio source separation evaluation campaign: data, algorithms and results," *In Proc. Int. Conf. on Independent Component Analysis and Blind Source Separation (ICA)* pp. 552-559.

Virtanen, T., 2003. "Algorithm for the separation of harmonic sounds with time-frequency smoothness constraint", *In Proc. of the 6th international conference on digital audio effects (Dafx-03).*

Von Békésy, G., 1960. Experiments in Hearing. New York: Acoustical Society of America Press (1989).

Wettschereck, D., Dietterich, T. G. 1995. "An Experimental Comparison of the Nearest-Neighbor and Nearest-Hyperrectangle Algorithms." *Machine Learning* 19(1):5-27.
URL: citeseer.nj.nec.com/wettschereck95experimental .html

Witten, I. H., Frank, E. 2005. *Data Mining Practical Machine Learning Tools and Techniques, second edition*. Morgan Kaufmann Publishers (imprint of Elsevier).

Yeh, C., Röbel, A., Rodet, X., 2005. "Multiple fundamental frequency estimation of polyphonic music signals," In *ICASSP 2005 IEEE Int. Conf. on Acoustics, Speech and Signal Processing Proc.*.

Zimmerman, D. W., 1998. "Invalidation of parametric and nonparametric statistical tests by concurrent violation of two assumptions". *Journal of Experimental Education*, 67(1), 55-68.

Zivanovic, M., Röbel, A., Rodet, X., 2004. "A new approach to spectral peak classification," *Proc. of the 12th European Signal Processing Conference (EUSIPCO'04)*, pp. 1277-1280.

Zwicker, E., 1990. *Psychoacoustics*, Springer-Verlag, Berlin, Germany.

Zwicker, E., Terhardt, E. 1980. "Analytical expressions for critical-band rate and critical bandwidth as a function of frequency", in *The Journal of the Acoustical Society of America,* November 1980.Vol. 68, Issue 5, pp. 1523-1525.