From Heuristics-Based to Data-Driven Audio Melody Extraction

Juan José Bosch Vicente

TESI DOCTORAL UPF / 2017

Thesis Director:

Dr. Emilia Gómez Gutiérrez
Music Technology Group
Dept. of Information and Communication Technologies
Universitat Pompeu Fabra, Barcelona, Spain

Dissertation submitted to the Department of Information and Communication Technologies of Universitat Pompeu Fabra in partial fulfillment of the requirements for the degree of

DOCTOR PER LA UNIVERSITAT POMPEU FABRA

Music Technology Group (http://mtg.upf.edu), Department of Information and Communication Technologies (http://www.upf.edu/dtic), Universitat Pompeu Fabra (http://www.upf.edu), Barcelona, Spain.

The doctoral defense was held on ..... at the Universitat Pompeu Fabra and scored as .....

Dr. Emilia Gómez Gutiérrez
(Thesis Supervisor)
Universitat Pompeu Fabra (UPF), Barcelona, Spain

Dr. Juan Pablo Bello
(Thesis Committee Member)
New York University (NYU), New York, USA

Dr. Xavier Serra
(Thesis Committee Member)
Universitat Pompeu Fabra (UPF), Barcelona, Spain

Dr. Maarten Grachten
(Thesis Committee Member)
Austrian Research Institute for Artificial Intelligence (OFAI); Vienna, Austria

...to my family, friends, and music

This thesis has been carried out at the Music Technology Group (MTG) of Universitat Pompeu Fabra in Barcelona, Spain, from Oct. 2012 to Apr. 2017, including a stay at the Centre for Digital Music (C4DM) of Queen Mary University of London from Sept. 2015 to Dec. 2015. It is supervised by Dr. Emilia Gómez. Part of the Work in Chapter 3 has been conducted in collaboration with Dr. Emmanouil Benetos, during a research stay at C4DM (QMUL). Work in some parts of this thesis has also been carried out in collaboration with the PHENICX team at the MTG, other institutions, and several collaborators including Rachel Bittner, Ricard Marxer, Marius Miron, Justin Salamon, Julio Carabías and Jordi Janer.

Our work has been supported by the Department of Information and Communication Technologies (DTIC) PhD fellowship (2012-16), Universitat Pompeu Fabra, the European Union under the PHENICX project (FP7-ICT-601166), the Spanish Ministry of Economy and Competitiveness under CASAS project (TIN2015-70816-R), and Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).

Acknowledgments

Firstly, I would like to express my sincere gratitude to my advisor Emilia Gómez for giving me the opportunity to join the MTG as a PhD student, and for the continuous support, patience, motivation and advice during this period. I would also like to thank Xavier Serra for previously accepting me for the SMC master studies, and for the teaching coordination during the PhD.

It was a special pleasure to work closely with Justin Salamon, Ricard Marxer, Perfecto Herrera, Jordi Janer, Álvaro Sarasúa, Julio Carabías, Marius Miron, Jordi Bonada, Oscar Mayor and Ferdinand Fuhrmann, and the complete PHENICX team at the UPF (and other institutions). I also had the great chance to collaborate with Rachel Bittner (NYU), whose work has been very important for this dissertation. I would also like to thank Karin Dressler for executing her methods on our dataset.

I got the amazing opportunity to spend three months in Queen Mary University of London, and I would like to thank my friends and colleagues in C4DM for making the research stay such a great experience, and particularly Emmanouil Benetos for all his advise and support. Also, I'm really thankful to Katerina, Maria and Jan for helping me out so much with accommodation issues.

I would then like to thank my fellow labmates at the MTG for the stimulating discussions, help, and for all the fun we have had in the last years, especially Frederic, Sergio(s), Zacharias, Alastair, Martí, Ajay, Gopala, Rafael, Esteban, Alfonso, Dmitry, Ángel, Carthach, Sergi, Daniel, Moha, Sebas, Joan, Sergi, Hendrik, Eduardo, Xavier, Jordi(s), Sertan, Hector, Jose, Georgi, Oriol, Nadine, Iwona, Olga, Rafa, and many others. Special thanks to Julián Urbano, Agustín Martorell and Panos Papiotis for their guidance and help in many aspects. Also many thanks to the university staff, especially Cristina, Vanessa, Lydia, Jana, Sonia and Alba. I am very grateful as well to everyone that participated in the recordings, and to the former SMC master colleagues, from whom I have learnt (and will learn) a lot.

I would finally like to thank the people that have been close and have taken care of me during this period, including friends, bandmates, flatmates, and especially my family, for supporting me so much throughout writing this thesis and in my life in general.

Abstract

The identification of the melody from a music recording is a relatively easy task for humans, but very challenging for computational systems. This task is known as "audio melody extraction", more formally defined as the automatic estimation of the pitch sequence of the melody directly from the audio signal of a polyphonic music recording. This thesis investigates the benefits of exploiting knowledge automatically derived from data for audio melody extraction, by combining digital signal processing and machine learning methods. We extend the scope of melody extraction research by working with a varied and realistic set of data, and considering multiple definitions of melody. We first present an extensive overview of the state of the art, and perform an evaluation on a novel symphonic music melody extraction dataset. Results show that most approaches are not able to generalise well to the characteristics of such data, which presents a high pitch range and a low energetic predominance of the melody over the accompaniment. A pitch salience function based on source-filter modelling is found to be specially useful in such context. We then propose its integration with melody tracking methods based on pitch contour characterisation, and evaluate them on a wide range of music genres. Firstly, this salience function is adapted for pitch contour creation by combining it with another one based on harmonic summation. This combination increases the salience of melody pitches and improves melody extraction accuracy over previous approaches, with two different contour-based melody tracking methods: pitch contour selection based on heuristic rules, and supervised pitch contour classification. Secondly, the latter approach is further improved by using novel timbre, tonal and spatial features, which are helpful to discriminate melodic from non-melodic pitch contours. Finally, we also propose a method for the estimation of multiple melodic lines based on pitch contour classification, which exploits continuity within melodic lines. The combination of supervised and unsupervised approaches leads to advancements on melody extraction and shows a promising path for future research and applications.

Resum

La identificació de la melodia en un enregistrament musical és una tasca relativament fàcil per a éssers humans, però molt difícil per a sistemes computacionals. Aquesta tasca es coneix com " extracció de melodia ", més formalment definida com l'estimació automàtica de la seqüència d'altures corresponents a la melodia d'un enregistrament de música polifònica. Aquesta tesi investiga els beneficis de l'ús de coneixement derivat automàticament de dades per a extracció de melodia, combinant processament digital del senyal i mètodes d'aprenentatge automàtic. Ampliem l'abast de la recerca en aquest camp, en traballar amb un conjunt variat i realista de dades, i considerar múltiples definicions de melodia. En primer lloc presentem una extensa anàlisi comparativa de l'estat de la qüestió, en les tasques relacionades amb extracció de melodia, i realitzem una avaluació en un context de música simfònica. Segons els resultats obtinguts, la majoria dels mètodes no són capaços de funcionar adequadament amb d'aquestes dades, que presenten un alt rang melòdic i un baix predomini energètic de la melodia sobre l'acompanyament. Un dels descobriments és que l'ús d'una funció de saliència tonal basada en un model font-filtre és especialment útil en aquest context, i proposem la seva integració amb mètodes de seguiment de melodia basats en la caracterització de contorns tonals. En primer lloc, adaptem aquesta funció de saliència per a la creació de contorns tonals, combinant-la amb una altra basada en la suma d'harmònics. Aquesta combinació augmenta la saliència de la melodia i ajuda a millorar la seva extracció amb dos mètodes de seguiment de melodia diferents: selecció de contorns basant-se en regles heurístiques, i classificació supervisada de contorns. En segon lloc, milloren aquest últim enfocament amb l'estimació de característiques de contorns tonals relacionades amb el seu timbre, tonalitat i posició espacials, que ajuden a diferenciar els contorns tonals que corresponen a la melodia dels que no. Finalment, proposem un mètode d'estimació de múltiples línies melòdiques basat en classificació de contorns tonals, i que promou la continuïtat dins de les línies melòdiques. La combinació de mètodes supervisats i no supervisats porta a millores en l'extracció de melodia i mostra un camí prometedor per a futures investigacions i aplicacions.

Resumen

La identificación de la melodía en una grabación musical es una tarea relativamente fácil para seres humanos, pero muy difícil para sistemas computacionales. Esta tarea se conoce como "extracción de melodía", más formalmente definida como la estimación automática de la secuencia de alturas correspondientes a la melodía de una grabación de música polifónica. Esta tesis investiga los beneficios del uso de conocimiento derivado automáticamente de datos para extracción de melodía, combinando procesado digital de la señal y métodos de aprendizaje automático. Ampliamos el alcance de la investigación en este campo, al trabajar con un conjunto variado y realista de datos, y considerar múltiples definiciones de melodía. En primer lugar presentamos un extenso análisis comparativo del estado de la cuestión en tareas relacionadas con extracción de melodía, y realizamos una evaluación en un contexto de música sinfónica. Según los resultados obtenidos, la mayoría de métodos no son capaces de funcionar adecuadamente con estos datos, cuyas características incluyen un alto rango melódico y un bajo predominio energético de la melodía sobre el acompañamiento. Uno de los descubrimientos es que el uso de una función de saliencia tonal basada en un modelo fuente-filtro es especialmente útil en dicho contexto. En esta tésis proponemos su integración con métodos de seguimiento de melodía basados en la caracterización de contornos tonales, y los evaluamos en una amplia gama de géneros musicales. En primer lugar, adaptamos esta función de saliencia para la creación de contornos tonales, combinándola con otra basada en la suma de armónicos. Esta combinación aumenta la saliencia de la melodía y ayuda a mejorar su extracción con dos métodos de seguimiento de melodía diferentes: selección de contornos basándose en reglas heurísticas, y clasificación supervisada de contornos. En segundo lugar, este último enfoque se mejora con la estimación de características de contornos tonales relacionadas con su timbre, tonalidad y posición espaciales, que ayudan a diferenciar los contornos tonales que corresponden a la melodía de los que no. Finalmente, proponemos un método para la estimación de múltiples líneas melódicas basado en la clasificación de contornos, que promueve la continuidad dentro de las líneas melódicas. La combinación de enfoques supervisados y no supervisados lleva a mejoras en la extracción de melodía y muestra un camino prometedor para futuras investigaciones y aplicaciones.

Contents

List of Symbols

The following is a list of variables used in the dissertation along with a short description.

Symbol	Description
Δf_m	Frequency error of the salience function
RR_m	Reciprocal rank score of the melody salience peak amongst the rest of salience peaks
S1	Relative salience of the melody peak in comparison to the highest salience peak in a frame
S3	Relative salience of the melody peak in comparison to the mean salience of the 3 highest peaks
f_0	Fundamental frequency
f	Frequency
w	Index
N_h	Number of harmonics considered when computing the harmonic summation salience function
τ_+	Frame-based weighting factor for salience peak filtering
τ_σ	Weighting factor for salience peak filtering
τ_v	Voicing detection threshold
ν	Voicing parameter
α^{th}	Threshold for filtering out pitch contours with low melodic likelihood
Acc	Multiple pitch estimation overall performance
Prec	Multiple pitch estimation precision
Rec	Multiple pitch estimation recall
E_{tot}	Multiple pitch estimation total error score
E_{subs}	Multiple pitch estimation total substitution error score
E_{miss}	Multiple pitch estimation missed error score
E_{fa}	Multiple pitch estimation false alarms
A	Agreement
MA	Mutual Agreement
MMA^h	Mean Mutual Agreement between humans
MMA^a	Mean Mutual Agreement between algorithms
RCA^h	Raw Chroma Accuracy obtained by humans

Symbol	Description
RCA^a	Raw Chroma Accuracy obtained by algorithms
tol	Tolerance to mistunings (semitones)
ω	Combination weight
R	Number of spectral shapes in the accompaniment
H	Activation matrix in NMF-based methods
W	Basis matrix in NMF-based methods
H_{f_0}	Pitch salience function based on source-filter model
N_{iter}	Number of iterations in computation of source-filter model
U_{st}	Number of frequency bins per semitone
K	number of atomic filters in the source filter model
$\circ$	Element-wise (Hadamard) product
$\hat{X}_v$	Lead instrument matrix in source-filter model
X_{f_0}	Source matrix in source-filter model
X_Φ	Filter matrix in source-filter model
$\varphi_{spatial}$	Spatial features
φ_{timbre}	Timbre features
φ_{tonal}	Tonal features
Ψ	Set of all features
i	Index
j	Index
k	Index

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Motivation

In recent decades, we have seen an exponential growth in the amount of music created, partly thanks to an enormous decrease in the cost of computer software applications for audio recording, editing and production, commonly known as Digital Audio Workstations (DAW). We have also seen that the distribution and consumption of music has recently experienced huge changes, first thanks to audio formats such as the mp3, and currently due to on-demand music streaming services. Due to the growth in computing power and the large amount of potential music related applications, in the last 15 years many efforts have been devoted into Music Information Research (MIR), including fields such as music signal processing, automatic music description, music perception and cognition, recommender systems, or automatic music generation. Due to the large amounts of data currently available, much research has focused on the automatic analysis of musical audio recordings, which would be useful for music retrieval, content-based recommendation, music creation and for musicological studies.

One of the most important elements of music is *melody*. According to Selfridge-Field:

> "*It is melody that enables us to distinguish one work from another. It is melody that human beings are innately able to reproduce by singing, humming, and whistling. It is melody that makes music memorable: we are likely to recall a tune long after we have forgotten its text*".
>
> (Selfridge-Field (1998))

Given its importance, there are multiple applications which benefit from an automatic description of the melody in music recordings. In this dissertation we deal with the most relevant task in automatic melody description, which is the identification of the sequence of pitches which correspond to the melody of a polyphonic music recording. This task is denoted as audio melody extraction (AME) (Salamon et al.,

2014). Note that musicology distinguishes between different musical textures such as monophonic, homophonic, heterophonic, and polyphonic, but similarly to most MIR literature, in this dissertation the term "polyphonic" simply refers to music in which two or more notes can sound simultaneously, while "monophonic" refers to sounds or music with a single pitch. Since the melody is useful for distinguishing one work from another, a computational system could be used to find versions of a given song using melodic information (Salamon et al., 2013). Another application is the retrieval of music by singing or humming part of the melody (Ghias et al., 1995; Dannenberg et al., 2007). An application which has also driven much research in the field is the separation of the lead instrument from the accompaniment. In practice, the main focus has been set on singing voice, motivated by the possibility of automatically creating karaoke versions from music recordings (Durrieu et al., 2010; Bosch et al., 2012b; Marxer, 2013). Further applications are the visualisation of a simplified music score (Bosch et al., 2015) and automatic melody transcription (Poliner et al., 2007; Gómez & Bonada, 2013). Melody extraction is also a useful initial step for other tasks such as singer identification (Mesaros et al., 2007), intonation analysis (Koduri et al., 2012), melodic motif analysis and discovery (Gulati et al., 2016a) and raga recognition (Gulati et al., 2016b).

This dissertation has been partially conducted within the PHENICX project (Gómez et al., 2013), which focused on instrumental symphonic music recordings. Such complex data presents different melodic and signal-related characteristics in comparison to vocal music, which has been the main focus of melody extraction research. As in other kinds of instrumental music, the pitch range is very wide, and pitch sequences may change rapidly, including large jumps, which are not common in singing voice (Salamon et al., 2014). Previous evaluations results have shown that the melody extraction accuracy obtained by state-of-the-art methods generally decreases when analysing instrumental data (Salamon et al., 2014; Bittner et al., 2014). An important motivation for this dissertation is thus to propose melody extraction methods that can adapt to the characteristics of the data under analysis, and to different definitions of melody.

1.2 Terminology

Music basically deals with the organization of sounds in time. In order to understand the context of this work, it is thus useful to clarify the definition of some terms related to both music and sound events.

1.2.1 Pitch and fundamental frequency

Sounds are commonly characterised by their pitch, loudness, duration, and timbre. According to the American National Standards Institute (ANSI), pitch is "*that attribute of auditory sensation in terms of which sounds may be ordered on a scale*

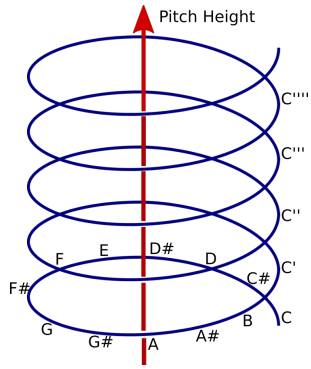

Figure 1.1: The helical model of pitch

extending from low to high". Baines & Temperley also defined pitch as "*a basic dimension of musical sounds, in which they are heard to be high or low*". Apart from this vertical dimension (pitch height) music psychologists also acknowledge a circular dimension (pitch class, or pitch chroma). This is due to the fact that humans perceive that musical notes are very closely related when their corresponding pitch is doubled. The musical scale (in classical western music) is based on this circular configuration. If we play this scale in ascending order, we go through the pitch class circle: A, A#, B, C, C#... and then reach G#, A, A# again, and so on. For this reason music psychologists have represented pitch as a 3D helix (see Figure 1.1), with both vertical and circular dimensions, with tones separated by octaves falling above one another on successive turns of the helix (Drobisch, 1846; Bachem, 1950).

Fundamental frequency (f_0) is the measurable physical counterpart of pitch, and is also understood as the inverse of the period of a periodic signal. In practise, most research in music signal processing uses both words as if they were synonyms. In this work, we will be consistent with the terms used in the MIR literature, and use both terms indistinctly.

Harmonic sounds are sounds with a spectral structure in which the dominant frequency components are approximately regularly spaced. In ideally harmonic sounds, the frequency of the harmonics (or overtone partials) are integer multiples of the f_0. In the real world, many instruments (e.g. piano or plucked string instruments such as guitar) are not perfectly harmonic (the partial frequencies are not in exact integer ratios), but they are still perceived as pitched. In fact, there are also instruments which are not harmonic but are perceived and used as pitched, such as mallet percussions (e.g. marimba, xylophone, vibraphone, etc.) or some drums (e.g. tabla, cowbell, etc.).

1.2.2 Melody

The definition of melody has evolved in the literature, depending on the context in which it was proposed. Ringer (2017) defines melody as "*pitched sounds arranged in musical time in accordance with given cultural conventions and constraints*". In the

Middle Ages, melody was associated with singing, and with the concept of song. In the 18th and 19th centuries, there were different viewpoints on whether melody was tightly related to harmony, or if it was independent. Helmholtz understood melody as the "*expression of motion in music, expressed in such a manner that the hearer may easily, clearly, and certainly appreciate the character of that motion by immediate perception*". Many other definitions have also been proposed in the 20th century. Solomon (2017) considered melody: "*a combination of a pitch series and a rhythm having a clearly defined shape*".

The lack of a unified definition is problematic for musicologists, but especially scientists, who would benefit from an objective and clear definition. In the context of MIR research, melody has however also been defined in several ways. Goto & Hayamizu (1999) defined melody as the "*series of notes [which] is more distinctly heard than the rest*". Levitin proposed a definition which considers multiple musical aspects:

> "*an auditory object that maintains its identity under certain transformations along the six dimensions of pitch, tempo, timbre, loudness, spatial location, and reverberant environment; sometimes with changes in rhythm; but rarely with changes in contour* " (Levitin (1999))

This definition is rather broad, but highlights several musical dimensions which are related to the way humans perceive melody, and the cues that composers and performers employ to make melodies perceptually salient. In any case the brain mechanisms used for processing melodic information are still far from being understood (Zatorre et al., 1994). Paiva defined melody in a somewhat different and slightly more objective way:

> "*the dominant individual pitched line in a musical ensemble*" (Paiva et al. (2006))

This definition highlights the characteristic dominance of the melody over the accompaniment, and the fact that there is a continuity in time and frequency (with the word *line*). Note that dominance may not just be related to loudness, since the rest of dimensions mentioned by Levitin (1999) also come into play. A related more objective recent definition by Salamon et al. is that:

> "*the melody is constrained to belong to a single sound source throughout the piece being analysed, where this sound source is considered to be the most predominant instrument or voice in the mixture*" (Salamon et al. (2014))

This is also the definition employed by the Music Information Retrieval Evaluation eXchange (MIREX) (Downie, 2008), in the Audio Melody Extraction (AME) task. Therefore, in the context of melody extraction, melody has been:

- a single pitched line: meaning that the melody is monophonic, played by an individual pitched instrument, and smooth in time and frequency. In practice, MIREX and most research in the field has until very recently also restricted the melody to be played by a single instrument in the whole recording, and the focus has been mainly set on vocal melodies.

- "dominant": in some sense, the melody needs to stand out from the rest of the music. Most research links dominance with energy, and partially pitch range (Durrieu et al., 2010; Salamon & Gómez, 2012). However, dominance is actually perceived by humans and there are other musical aspects involved (e.g. instrumentation, rhythm, tonality) which are often neglected.

- in a "musical ensemble": meaning that the signal is polyphonic (it contains multiple concurrent pitches), and polytimbral (there is more than one instrument playing).

However, it is also interesting to consider broader but less objective definitions such as the one proposed by Poliner et al., who stated that

> "*the melody is the single (monophonic) pitch sequence that a listener might reproduce if asked to whistle or hum a piece of polyphonic music, and that a listener would recognize as being the "essence" of that music when heard in comparison*" (Poliner et al. (2007))

This open definition involves cognitive processes, and is the one we adopt in this dissertation when we deal with symphonic music recordings (Chapter 3). However, we also consider other definitions in Chapter 4 for two main reasons. First, because we acknowledge the intrinsic ambiguousness of the melody, and second, because various definitions of melody may be useful in different applications contexts or different music genres. For instance, there are situations where multiple instruments play melodic content simultaneously, and therefore it would make sense to consider multiple single pitch sequences. It is therefore interesting that computational methods learn to estimate different kinds of melodies, according to our needs.

1.2.3 Melody extraction

The MIREX audio melody extraction evaluation task deals with the identification of the sequence of melody pitches corresponding to the melody from polyphonic musical audio. In this task, "*Pitch is expressed as the fundamental frequency of the main melodic voice, and is reported in a frame-based manner on an evenly-spaced time-grid*". Melody extraction was defined by Salamon (2013) in his PhD thesis as: "*fundamental frequency estimation of a single predominant pitched source from polyphonic music signals with a lead voice or instrument*". This definition is similar to

the pragmatic definition defined before, and is suitable for an engineering perspective. However, it is also very specific to certain kinds of musical data.

In this work we focus on: "*the identification of the fundamental frequency corresponding to the melody pitch from polyphonic music signals*". This task does not include any definition of melody, since we do not consider it to be unique. Generally speaking, we adopt a similar task definition but wider in comparison to previous research. Some of the differences are that we:

- allow a melody pitch to be played by several instruments, e.g. an orchestral section;
- allow the melody line to be played by different instruments in the same excerpt;
- allow different definitions of melody, even for the same excerpt
- pay special attention to non-vocal melodies, which present different characteristics in comparison to vocal melodies.

Furthermore, we consider an additional melody definition, which contemplates that multiple melody lines may be played simultaneously. The melody definition and the type of data under analysis are thus to be modelled by the proposed melody extraction methods.

1.2.4 Tonality

Music is typically formed by combinations of notes, which are commonly pleasant to listeners. Piston (1948) defines tonality as "*the organized relationship of tones in music*". Tonality implies a central tone (tonic), with all other tones tending toward it in one way or another. Tonality has also been defined as "*the systematic arrangement of pitch phenomena and relations between them*" (Hyer, 2016). Harmony refers to the combination of several notes, producing chords, and successively, to produce chord progressions. Perceived consonance between notes is not absolute, but there are physical phenomena which are closely related, and thus make it somewhat constant across different cultures. According to Terhardt (1977) consonance depends on the ratio between the fundamental frequencies, and consonant combinations correspond to the ratio $\frac{n+1}{n}$ where $n < 5$. In the equal temperament tuning system, every pair of adjacent notes has an identical frequency ratio, and thus the same perceived distance for a listener. Western popular and classical music is commonly based on the 12 tone equal temperament scale, in which an octave is divided into 12 equal steps, also known as semitones.

Chroma features (or pitch class profiles) have been very commonly used to represent tonality, and have been employed in multiple MIR tasks. They represent the relative intensity of the 12 semitones in an equal-tempered chromatic scale (pitch classes),

discarding octave information. In practise, it is also possible to compute subdivisions between the semitones so as to have a more fine-grained feature.

1.2.5 Timbre

According to the Acoustical Society of America (ASA) Acoustical Terminology, timbre is "*that attribute of auditory sensation which enables a listener to judge that two non identical sounds, similarly presented and having the same loudness and pitch, are dissimila*r" (ANSI , S1.1-1994). Timbre is related to several aspects of the sound: the frequency spectrum, the temporal variation of the frequency spectrum, the time envelope: attack time (and characteristics), decay, sustain, release (ADSR envelope) as well as the transients.

Timbre information is potentially useful for most audio-focused MIR tasks. For instance, timbre features (Fuhrmann, 2012; Bosch et al., 2012a) have been used for automatic musical instrument recognition in both monophonic and polyphonic music. Transcription methods have also taken advantage of timbre information, in order to distinguish between sources, when pitch information is not sufficient (Benetos, 2012; Duan et al., 2014). Sound production models such as the source-filter model have also been proposed to decouple timbre from pitch information (Noll, 1967), and several variants have been used for tasks such as melody extraction and source separation (Durrieu et al., 2010).

1.3 Research question and methodology

Now that we have revised the most important terms for understanding the context of this dissertation, we can introduce our research question, formulate our hypotheses and present the methodology we will follow to validate them. The main research question to be solved in this thesis is the following:

Can melody extraction algorithms benefit from modelling the context of the data to be analysed?

Let's analyse the elements of this question in order to properly understand its meaning. First, we will investigate *melody extraction algorithms*, which aim at automatically estimating the sequence of melody pitches from a musical audio signal. This involves both melody pitch estimation and voicing detection (determining if a frame contains a melody pitch or not). Second, we will deal with *data analysis and modelling*, and we will investigate the use of such models for melody extraction. Third, we will *evaluate* if data-driven methods lead to *benefits in melody extraction* e.g. better pitch estimation or voicing detection accuracy, or more generalisability. Related to this research question, we formulate three hypotheses:

1. Most melody extraction algorithms are focused on simple vocal data and may not generalise well to other, more complex musical contexts.

2. Features related to timbre, tonality, and spatial information would be useful for improving melody extraction algorithms.

3. Supervised and unsupervised learning from data would allow advancing the state-of-the-art in melody extraction.

In order to answer the research question and validate the hypotheses we follow the following methodology. Since we require data from different musical contexts, we first create a dataset on a challenging context (symphonic music), and annotate it with the melody. We then perform an evaluation of the state of the art, focusing on melody pitch estimation. We analyse seven complete methods as well as four intermediate steps to understand where their limitations come from. Based on this analysis, we propose new methods based on music signal processing and machine learning techniques to improve audio melody extraction in a wide range of musical data, with different melody definitions. We start from more traditional rule-based approaches, and increasingly add more knowledge extracted from data. We also propose methods to extract timbral, tonal and spatial information and evaluate the benefits of using them for melody extraction. Finally, we propose a method for multiple lines estimation, trained on a dataset which allows several melodic lines. Inspired by the principles of research reproducibility, and in order to facilitate open science, the datasets produced during this thesis, as well as the proposed melody extraction methods are publicly available[1].

1.4 Research context

As previously introduced, this thesis has been conducted within PHENICX (Gómez et al., 2013), a European project which focused on Western classical music in large ensemble settings. The main focus of the project was symphonic music, which is a largely unexplored area in MIR research, partially due to the complexity of the data. For instance, it involves a large number of instruments and a high spectral overlap, which pose many challenges to automatic music transcription or source separation methods. Beat tracking accuracy commonly decreases with expressive music (varying tempo), and harmonic descriptors are often limited to global key, which is usually not enough to represent the tonal content of symphonic music pieces (Gómez et al., 2013). Such musical context is thus ideal to analyse the applicability of melody extraction algorithms to other data than vocal melodies, which has been the main research focus of the literature.

[1] http://www.mtg.upf.edu/node/3737

Classical music is a very strong example of European cultural heritage, and this research project aimed at improving the musical experiences of a wide range of listeners, in live concerts or with digital applications. In the PHENICX project, we considered classical concerts to be multi-perspective experiences, since listeners bring multiple personal perspectives through their differing levels of musical knowledge (Roose, 2008).

One of the goals of the project was to create visualisations which could allow people from different musical backgrounds to have enriched music experiences, get them more interested in this kind of music, and maybe allow them to appreciate musical aspects with which they were not familiar before (Dobson, 2010; Melenhorst & Liem, 2015). An important focus of the project were people which had not been much exposed to symphonic music before, and had little or no musical training. One interesting visualisation for such focus group is a simplification of the musical score, where only the most essential elements would be displayed, e.g. the main melody and the instruments playing it. Within this project, this dissertation has contributed to the techniques for automatic music analysis, with a web-based tool for melody visualisation, as well as with the creation of two datasets. These datasets widen the scope of melody extraction, multiple pitch estimation, instrument identification and (score-informed) source separation research.

Apart from this musical context, we also work on a varied set of music with several melody definitions, in order to make our methods applicable in a wide range of genres, and to investigate their behaviour on different kinds of data.

1.5 Scientific contributions

This thesis extends the scope of melody extraction research, by allowing multiple melody definitions, and not only focusing on vocal data. We consider a more varied, complex and realistic set of data, including on the one hand a symphonic music dataset and on the other hand a dataset with a varied set of genres. The main contributions of this dissertation are introduced next, and a more detailed review is presented in Section 5.3.

State of the art review and evaluation of melody pitch estimation on symphonic music. We present an extensive state-of-the review in the context of audio melody extraction and pitch estimation, and analyse the performance of a selection of methods in the context of melody extraction on symphonic music. We also analyse the correlation between melody extraction accuracy and characteristics of both the melody and the audio signal itself, such as the degree of dominance of the energy over the accompaniment. We also propose a novel set of metrics to gain more knowledge from melody extraction algorithms, related to the smoothness of the melody contour.

Melody extraction methods. We propose the use of unsupervised, and supervised methods for melody pitch detection, as well as the combination of several pitch salience functions with melody tracking based on from pitch contour features. First, we propose a melody-oriented pitch salience function based on a source-filter model, which increases the salience of the melody pitch, and reduces the salience of non-melody pitches. We also propose a set of metrics to measure the salience of non-melody pitches, which is especially important when the salience function is used for pitch contour creation.

We then propose several melody extraction methods which combine the proposed salience function with pitch contour characterisation for melody tracking. The first one is based on pitch contour selection, and achieves very competitive performance on MIREX evaluation campaign (Bosch & Gómez, 2015). Moreover, it also improves over the state of the art in a large, realistic and varied dataset such as MedleyDB. Second, we propose the combination of a source-filter model with a melody decoding approach based on pitch contour classification. This method substitutes the heuristic rules used in pitch contour selection by a random forest classifier and Viterbi decoding. This approach allows to learn from the data to be modelled, avoids manual parameter tuning, and provides an easier integration of new pitch contour features. We then propose novel timbre, tonality and spatial position features, which generally lead to improvements in overall accuracy on a wide range of music data. We also propose a method for the joint decoding of multiple melodic lines, which exploits contour features for pitch contour classification, and feature continuity within melody lines. Finally, we investigate timbre-informed methods for melody pitch estimation, based on Probabilistic Latent Component Analysis (PLCA) framework (Benetos, 2012). We also study the expansion of the set of spectral templates, in an unsupervised fashion, by analysing the music signal under analysis. The source code of the proposed melody extraction methods is freely available, contributing to open science and research reproducibility[2].

We also evaluate the use of one method in practical applications such as source separation (melody from accompaniment), leading to improvements in separation quality. Finally we also explore melody visualisation prototypes, in web and tablet applications.

Melody annotation process and analysis of agreement. We propose a methodology for the creation and annotation of a melody extraction dataset in the context of symphonic music recordings, based on asking multiple subjects to sing along the music. After a manual agreement analysis, we annotate the notes that the participants consider as melody. We also conduct an automated analysis of agreement between both humans and algorithms, and study correlations with musical or signal-related factors.

[2] http://www.mtg.upf.edu/node/3737

Datasets. The first and most relevant dataset created in this work is *Orchset* (Bosch et al., 2016b), which is intended to be used for the evaluation of melody extraction algorithms. This collection contains 64 audio excerpts focused on symphonic music with their corresponding annotation of the melody. This dataset has been employed in the audio melody extraction task in MIREX evaluation exchange, and it has now become publicly available (Bosch et al., 2016b)[3]. Another dataset we created in the context of the PHENICX project is useful for tasks such as as score-informed source separation, score following, multipitch estimation, transcription or instrument detection. The *PHENICX-Anechoic dataset* consists of four passages of symphonic music from the Classical and Romantic periods (Pätynen et al., 2008) with the following instruments: violin, viola, cello, double bass, oboe, flute, clarinet, horn, trumpet and bassoon. We created a ground truth score, by manually annotating the notes played by each of the instruments, and used it for the evaluation of score-informed source separation methods (Miron et al., 2016).

1.6 Thesis outline

This thesis is divided in five chapters. **Chapter 1** is this introduction, where we have presented our motivation, research question and methodology, as well as some key terms to understand them.

In **Chapter 2**, we present the scientific context of this dissertation (Section 2.2), and then present an overview of the literature related to single pitch estimation, melody extraction and multipitch estimation (Section 2.3). We then present a review on strategies for pitch salience estimation (Section 2.4), melody pitch tracking (Section 2.5), voicing detection and polyphony estimation (Section 2.6). Finally we introduce the evaluation methodology of both melody extraction and multipitch estimation algorithms, commonly used datasets, and comment on previous MIREX Audio Melody Extraction task results (Section 2.7).

In **Chapter 3**, we study if state-of-the-art approaches are able to generalise well to a symphonic music context. To do so, we analyse the performance of eleven pitch estimation methods on the task of melody pitch estimation, with a novel melody extraction dataset in this musical context. The creation of an annotated symphonic music dataset for melody extraction reveals to be a challenge, partially due to the lack of a established annotation methodology when there is more than one instrument playing the melody. We propose a methodology which deals with the collection of excerpts in which human listeners agree in the sequence of notes that they hum or sing to represent it (Section 3.2). We also present an automatic analysis of agreement between humans and algorithms when estimating the melody, and study the correlation of both pitch estimation accuracy and mutual agreement, with musical characteristics from the annotated melodies (3.3). We then discuss the results obtained by 11 meth-

[3] mtg.upf.edu/download/datasets/orchset

ods, including melody extraction and multipitch estimation algorithms, as well as an intermediate representation: pitch salience functions, and then analyse how the combination of different pitch salience functions can improve melody pitch estimation (Section 3.5). Finally, in Section 3.6 we further study and adapt one of the evaluated methods to perform timbre-informed melody pitch estimation in a symphonic music context.

In **Chapter 4**, we analyse the benefits of exploiting data-derived knowledge in audio melody extraction on a wider range of music material, including genres such as pop, rock, opera, jazz, as well as symphonic music, and considering multiple melody definitions. We propose and evaluate several approaches, starting from rule-based algorithms, and increasingly exploiting available data. We address both tasks: pitch estimation and voicing detection, in contrast with Chapter 3, which focused on pitch estimation. Based on conclusions derived from Chapter 3 and Chapter 2, we propose methods based on the combination of source-filter modelling and pitch contour characterisation. We first deal with pitch salience estimation (Section 4.2), adapting a salience function created with a source-filter model for the formation of pitch contours. After creating pitch contours (Section 4.3), we track the melody pitch following either a heuristic approach (Section 4.4), or a data-driven approach (Section 4.5), and perform a comparative evaluation in the context of melody extraction and source separation. We then propose novel timbre, spatial and tonal features for pitch contour characterisation (Section 4.6), which we use for data-driven melody tracking. Finally, we propose a method for estimating multiple melodic lines (Section 4.7), based on for joint multiple pitch decoding, which characterises and models pitch contour transitions in the data.

Finally, in **Chapter 5**, we provide a summary of the work presented in this dissertation, and present prototype applications which have been build on top of it. We also discuss some future perspectives and present a detailed list of contributions.

This thesis also contains three appendix sections. In Appendix A, we list the relevant publications by the author. In Appendix B, we present the files used for the source separation experiment from Section 4.4.5. Appendix C presents the glossary of abbreviations and other terms used in this thesis.

Chapter 2

Scientific Background

2.1 Introduction

In this chapter, we present a review of the existing literature related to the work presented in this dissertation, commencing with an introduction to the scientific context of this thesis (Section 2.2). Subsequently, we provide an overview of the relevant literature in pitch estimation (Section 2.3), including single pitch estimation, audio melody extraction and multipitch estimation. We then present relevant work done for pitch salience estimation (Section 2.4), melody pitch tracking (Section 2.5), voicing and polyphony estimation (Section 2.6). A strong focus is set on approaches based on the computation of pitch salience and the use of pitch contours, since the methods proposed on Chapter 4 build upon them. Finally, we present an overview of the evaluation metrics and datasets for melody extraction and multipitch estimation tasks, and analyse MIREX Audio Melody Extraction task results (Section 2.7).

2.2 Scientific Context

2.2.1 Music Information Research

Music Information Research (also known as Music Information Retrieval, or more commonly MIR) was born due to the increasing need of methods for the automatic understanding, description, organisation, recommendation, transformation and retrieval of music. MIR covers a wide range of tasks, which deal with many different musical aspects (melody, rhythm, harmony, timbre, etc.), and involves multiple disciplines: psychoacoustics, musicology, signal processing, machine learning, informatics, etc. MIR has focused on both the symbolic domain (e.g. digitised scores) and audio domain (recordings), even though the research community has increasingly paid more attention to the latter, partially due to the potential applications in the real world.

The International Society of Music Information Retrieval Conference (ISMIR) is the yearly conference which gathers this scientific community. MIR algorithms have

been evaluated in public evaluation campaigns since 2004, when the Audio Description Contest (ADC) was held at Universitat Pompeu Fabra in Barcelona, during the 5th ISMIR Conference. Inspired by the well-established TREC framework, the Music Information Retrieval Evaluation eXchange (MIREX) was created shortly afterwards, which continued with the evaluation in a wide range of tasks. The number of algorithms kept increasing during the following years, and nearly tripled between 2007 and 2015. The increasing need of operating costs recently motivated the birth of cosmir (McFee et al., 2016), an ongoing effort to create an open and sustainable framework, with distributed computation, open content, and an incremental evaluation, currently focusing on the automatic instrument recognition task. Due to the interest of the community, melody extraction is one of the tasks evaluated yearly in MIREX. More details about melody extraction evaluation in MIREX are presented in Section 2.7.4. Music transcription and source separation are closely related tasks, which are also evaluated in the same forum.

2.2.2 Computational auditory scene analysis

Perception and cognition theories have informed the creation of MIR systems. For instance, melody extraction algorithms have used pitch perception theories, and timbre perception studies have provided useful grounds to model musical instruments.

In most common situations, humans do not listen to individual sounds, but a mixture (or combination) of them, e.g. a voice with the background of car traffic, or multiple voices and music in a bar. Additionally, acoustic sources typically generate complex sounds, having many frequency components. The perceptual process by which the auditory system separates the individual sounds in natural-world situations is called Auditory scene analysis (ASA). The grouping of these components can determine the perceived pitch, timbre, loudness, and spatial position of the resulting sounds. ASA deals with organising and segmenting components of sounds in the time-frequency space, and assigning them into auditory streams (Bregman, 1994). This process takes place by two inter-dependent processes: sequential grouping, which senses data over time, and simultaneous grouping, which groups components of sounds which arrive at the same time. Several grouping principles are defined for both processes, which are closely related to the Gestalt principles (or laws) of grouping in the visual domain. Such principles also explain that humans are able to distinguish (up to a certain extent) between different instruments playing in a song. ASA principles have inspired computational methods to emulate the human understanding of an auditory scene, in the field of "Computational Auditory Scene Analysis". ASA has also inspired many MIR approaches, in very different tasks. To give an example, melody extraction approaches commonly use principles of closure in the time-frequency domain to group pitches into the melody line, and segregate them from the pitches played by the accompaniment instruments (Goto, 2004; Durrieu et al., 2010; Salamon & Gómez, 2012).

2.2.3 Music transcription

In polyphonic music transcription the main problem is to detect the notes present in an acoustic music signal. Those notes can be concurrent on time and may come from different sound sources. The final output depends on the system and ranges from a mid-level music representation, using audio time as reference, to real music notation providing beat-related information. The former outputs a description related to pitches, onsets, offsets, loudness and may attempt to stream pitches by sources. The latter provides note names, key, rhythm, instruments and is based on score-time, thus needing to estimate beat-related information and identify dynamics/expression. Applications of music transcription systems include visualisation, computational musicology, search and annotation of musical information and interactive music systems.

Multipitch (or multiple-f_0) estimation methods deal with the core problem in music transcription, which consists on the estimation of the pitches present at a given time. Such methods may use techniques derived from signal processing, statistical modelling, spectrogram factorisation, machine learning, genetic algorithms, sparse coding, etc. Methods perform either an iterative or a joint estimation of pitches. The computed time-pitch representation (pitch activation matrix), is then further processed to detect note onsets and offsets. A related task is melody transcription, which deals with the estimation of the notes corresponding to the melody from an audio clip. Many methods also start by extracting the sequence of melody pitches, and then use onset and offset information to determine the beginning and end of the notes (Poliner et al., 2007).

One of the main challenges is to deal with sound sources with harmonics that interfere with each other, which usually occurs when they are played at the same time. Another challenge is to deal with different types of music, musical instruments and playing techniques (e.g. legato or staccato), due to the variety in onset and offset acoustic characteristics.

2.2.4 Source separation

Source separation deals with the recovery of one or several sound sources from one (or several) mixture(s). A classical problem is the separation of the different speakers, which is closely related to the "cocktail party problem" (Chan et al., 2015; Bronkhorst, 2000). In the case of music mixtures, most research has been motivated by applications such as karaoke or music remixing, requiring voice/accompaniment separation. In this task, the identification of the pitch played by the lead instrument (melody extraction) is a key step for some source separation algorithms (Durrieu et al., 2010), although other methods do not perform this intermediate step. Other music source separation tasks deal with the recovery of harmonic and percussive components (Canadas-Quesada et al., 2014), the separation of individual instruments, or the separation of melody, bass, drums and other instruments (Ozerov et al., 2012). Most algorithms deal with offline source separation, but recently several approaches have

been proposed to deal with online source separation (with very low latency) (Simon & Vincent, 2012; Marxer, 2013).

Source separation approaches may exploit spatial information, when there is more than one available channel, and sources in music mixtures are panned differently in the stereo image. Some approaches use spectral bin classification masks with panning information. For instance, Vinyes et al. (2006) use the pan and the inter-channel phase difference) features to classify spectral bins, and used this information for demixing. Marxer (2013) uses harmonic masks to complement panning when it is not sufficient.

2.3 Pitch estimation

As previously introduced, pitch estimation methods have been used for a wide variety of tasks, and different types of data. Pitch estimation is the first step for melody and multipitch transcription, and is useful also for source separation, or other MIR tasks such as chord and key estimation.

Such methods commonly work on the time domain or in the frequency domain, but some approaches have been proposed on the spectro-temporal domain. Time domain methods are based on the idea that harmonic sounds are periodic, and the autocorrelation function can be used to find the period of the signal. Such approaches can however not easily deal with signals with multiple harmonic sounds, since the periodicity is not clear. Frequency domain methods generally try to recognise the harmonic patterns of each of the pitches present in a signal. One of the problems is the overlap in the harmonics of the pitches, and the variation of the amplitude of the harmonics. Spectro-temporal methods aim at exploiting both temporal and spectral representations (Su & Yang, 2015).

The problem of mapping a sound signal from time-frequency domain to a 'time-pitch' domain has turned out to be especially hard in the case of polyphonic music signals, in comparison to monophonic signals, since several sound sources are active at the same time. Multipitch (multiple f_0) estimation is still one of the main challenges in MIR, as methods need to deal with masking, overlapping tones, mixture of harmonic and non-harmonic sources and the fact that the number of sources might be unknown (Schedl et al., 2014). While the focus of this thesis is set on the melody extraction task, we also study multiple pitch estimation methods, since they address similar problems.

2.3.1 Single pitch estimation

Single pitch estimation algorithms are based on the assumption that the signal only presents an active harmonic source at a time. De Cheveigné (2006) presented a review on related methods, dividing them according to the domain in which they work: temporal, spectral, and spectrotemporal.

One of the most simple temporal methods is based on computing the zero-crossing

rate (how often does the signal cross the zero level). The main problem of this approach is the high sensitivity to noise. Other early approaches attempted to model the auditory system to compute the perceived pitch (Terhardt, 1979; Terhardt et al., 1982). Gold & Rabiner (1969) proposed the combination of several periodicity estimates, coming from impulse trains derived from the signal. Most commonly, time domain methods are based on the idea that harmonic sounds are periodic, and use the autocorrelation function (ACF) to find the period of the signal. Periodic signals have the first major peak of this function at the fundamental period. Some ACF-based approaches have focused on the time domain (Medan et al., 1991; Talkin, 1995) and others on the frequency domain (Klapuri, 2000). Ross et al. (1974) also used ACF but used the average magnitude difference function, based on the city-block distance between two chunks of the signal. De Cheveigné (1998) proposed the squared-difference function, replacing the city-block distance with Euclidean distance instead. De Cheveigné & Kawahara (2002) proposed a normalized form of the squared-difference function for the YIN pitch estimation algorithm. This avoids spurious peaks near zero lag, avoiding harmonic errors. Mauch & Dixon (2014) proposed pYin, a modification of YIN in which they replaced the threshold parameter by a parameter distribution, obtaining several f_0 candidates per frame. In a second stage, they use a HMM which is decoded using the Viterbi algorithm (Forney, 1973). Results show improvements in recall and precision over YIN on a database of over 30 hours of synthesised singing.

Some spectral methods also used ACF, exploiting the fact that the partials of a harmonic sound occur at integer multiples of the fundamental frequency of that sound. The maximum of the ACF for a harmonic spectrum corresponds to the f_0. Lahat et al. (1987) proposed a method based on flattening the spectrum, and estimating the f_0 from ACF. A previous approach by Noll (1967) proposed the use of cepstrum analysis. The (power) cepstrum is defined as the inverse Fourier transform of the logarithm of the power spectrum of the signal. Periodic signals present a strong peak at the location which corresponds to the inverse of the f_0. Other approaches are based on harmonic matching, in which the peaks of the magnitude spectrum are matched against the expected locations of the harmonics of a candidate (Piszczalski & Galler, 1979; Maher & Beauchamp, 1994). Doval & Rodet (1993) proposed a maximum likelihood (ML) approach, based on the representation of an input spectrum as a set of sinusoidal partials, and used HMMs for tracking. Klapuri (2000) proposed a bandwise processing algorithm, in which the final estimation is computed by combining the pitch likelihoods from different bands. This provides robustness against inharmonicity, since in narrow bands the frequency distance between partials can be considered constant.

The most commonly found error is that the estimated pitch is an octave above or below the real pitch (octave errors). The main motivation of spectro-temporal f_0 estimation methods is the fact that spectral methods generally have a tendency to exhibit errors in integer multiples of the f_0 (harmonic errors), while methods on the temporal domain typically exhibit errors at submultiples of the f_0 (subharmonic errors) (Klapuri, 2003). The approach by Meddis & O'Mard (1997) splits the input signal using a filterbank,

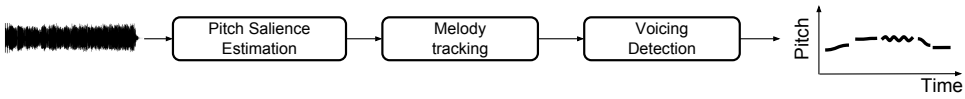

Figure 2.1: Schema of salience-based melody extraction methods

giving emphasis to a different range of frequencies in each channel. Then the ACF is computed for each channel in the time domain, and results are added to create a global autocorrelation function.

Monophonic pitch estimation is commonly considered to be a solved task, even though the results vary depending on the source and the input parameters. However, the estimation of the pitch corresponding to the melody, or the estimation of all pitches present in a polyphonic mixture are still very challenging tasks.

2.3.2 Melody extraction

As previously introduced, in melody extraction, we deal with the estimation of the melody pitch in polyphonic audio signals. Thus, the signal does not contain a single harmonic source, which would make the application of a monophonic pitch tracking algorithm fail in most cases.

Composers and performers use several cues to make melodies perceptually salient, including loudness, timbre, frequency variation or note onset rate. Melody extraction methods commonly use cues such as pitch continuity and pitch salience, and some of them group pitches into higher level objects (such as tones or contours), using principles from Auditory Scene Analysis (Goto, 2004; Paiva et al., 2006; Dressler, 2012b; Salamon & Gómez, 2012; Marolt, 2004b). Some approaches have also considered timbre, either within a source separation framework (Durrieu et al., 2010; Ozerov et al., 2007), with a machine learning approach (Ellis & Poliner, 2006), or in a salience based approach (Marolt, 2005; Hsu & Jang, 2010).

There are different strategies for melody extraction. Salamon et al. (2014) divided them in salience-based and separation-based. The former start by computing a pitch salience function and then perform tracking and voicing detection (see Figure 2.1), and the latter involve the separation of the melody from the accompaniment, which is more or less explicit depending on the approach. Salience-based methods are detailed in Section 2.4.

A clear example of a separation-based approach is Tachibana et al. (2010). In this approach, they propose the use of Harmonic-Percussive Sound Separation (HPSS), originally conceived to separate sources which are smooth in time (harmonic), and those smooth in frequency (percussive). The approach is based on modifying the window length to first separate chords (more sustained) from other content which is more variable (melody and percussion). Then, a HPSS is run again, but this time using the original window size, to separate melodic and percussive elements. The melody is then

estimated from the spectrogram of the separated (or at least, enhanced) melody signal using dynamic programming, to find the path which maximises the maximum posteriori probability of the frequency sequence. The probability of a frequency given the spectrum is proportional to the weighted sum of the energy at its harmonic multiples. Transition probabilities depend on the distance between two consecutive frequency values. Fan et al. (2016) propose another separation-based method but only for vocals. The algorithm performs first singing voice separation based on Deep Neural Networks (DNN) in a supervised setting, and then pitch tracking based on dynamic programming, considering both periodicity and smoothness. Hsu et al. (2011) also enhance the melody signal, in an approach which aims at vocal melody estimation with a trend estimation algorithm which dynamically adapts the pitch range for melody estimation according to the signal.

Some approaches combine concepts from salience-based methods and source separation methods. Durrieu et al. (2010) use an Expectation Maximisation (EM) method to compute a salience function by modelling the melody signal with a source-filter (S/F) model, and then perform an explicit separation of the lead for voicing estimation. Other approaches such as Hsu & Jang (2010) and Yeh et al. (2012) use HPSS to attenuate the accompaniment signal, and then compute a salience function from which they extract the melody.

Finally, there is a recent tendency to perform data-driven melody extraction, motivated by the advances in machine learning (especially deep learning) and the increasing dataset availability (see section 2.7.5). Poliner & Ellis (2005) proposed a machine learning based approach, based on training an SVM classifier to estimate the melody note from a feature vector derived from the power spectrum. Voicing detection was performed with a threshold on the magnitude squared energy found between 200 and 1800 Hz. Several algorithms using neural networks have been recently proposed but they have only been tested on vocal data. Kum et al. (2016) proposed a multicolumn deep neural network to predict the pitch label of singing voice, using spectrograms as input. The output of several Recurrent Neural Networks (RNN) with different pitch resolutions were combined. Voicing was handled separately, with a simple energy-based voice detector. Rigaud & Radenen (2016) propose an approach based on Deep Neural Networks, which estimates pitch activations (pitch salience) and singing voice detection separately. Verma & Schafer (2016) propose a neural network approach for estimating the f_0 of a periodic signal directly from the time domain waveform, without pre or post-processing. They quantize the frequency range into 24 states per octave and an additional state for silence or unvoiced speech. Even though the system was trained on speech (TIMIT database), it obtains similar *RPA* as the method proposed by Salamon & Gómez (2012) on MIR1k dataset. However, this method was not tested on more complex and realistic data, such as MedleyDB (Bittner et al., 2014). Finally, Bittner et al. (2015) propose a data-driven method for melody pitch tracking, based on Pitch Contour Classification (PCC), further described in Section 2.5.4. Some data-driven approaches make use of data augmentation (e.g. using time

stretching, random frequency filtering), which is reported to be useful for both pitch (Kum et al., 2016) and voicing estimation (Schlüter & Grill, 2015).

A summary of relevant melody extraction approaches for this dissertation is presented in Table 2.1, where they are classified as salience-based, separation-based, or data-driven approaches. As previously introduced, some data-driven approaches also exploit a pitch salience function as intermediate representation, but they use supervised methods in some part of the algorithms. This is a list of publicly available methods (‡ source code or † binary form) for melody extraction and pitch tracking, as well as intermediate representations (programming language in square brackets):

1. *LabROSAmelodyextract2005* ‡[4] [Matlab, java]: melody extraction algorithm submitted to MIREX 2006 by Poliner & Ellis (2005) .

2. *FChT* ‡[5] [Matlab/C++]: Fan Chirp Transform (FChT) and f0gram (salience function) proposed by Cancela et al. (2010).

3. *separateLeadStereo* ‡[6][python]: melody extraction method and lead instrument/accompaniment source separation methods submitted to MIREX 2010 by Durrieu et al. (2010).

4. *IMMF0salience* ‡[7][vamp plugin]: pitch salience function based on a source-filter model from Durrieu et al. (2010, 2011).

5. *MELODIA* †[8] [vamp plugin]: melody extraction method by Salamon & Gómez (2012) based on pitch contour selection. An open source implementation of this method is available in Essentia [9] (Bogdanov et al., 2013).

6. *Fuentes2012_ICASSP* ‡[10] [Matlab]: melody extraction method by Fuentes et al. (2012) based on PLCA.

7. *contour_classification* ‡[11][python]: pitch tracking method by Bittner et al. (2015), based on supervised pitch contour classification (and pitch contour creation from MELODIA).

8. *MelodyExtraction_MCDNN* ‡[12][python]: melody extraction method evaluated in MIREX 2016, based on deep neural networks by Kum et al. (2016).

[4] http://labrosa.ee.columbia.edu/projects/melody/
[5] http://iie.fing.edu.uy/investigacion/grupos/gpa/fcht.html
[6] https://github.com/wslihgt/separateLeadStereo
[7] https://github.com/wslihgt/IMMF0salience
[8] http://mtg.upf.edu/technologies/melodia
[9] http://essentia.upf.edu
[10] http://www.benoit-fuentes.fr/articles/Fuentes2012_ICASSP/index.html
[11] https://github.com/rabitt/contour_classification
[12] https://github.com/keums/MelodyExtraction_MCDNN

The melody extraction methods proposed in this thesis (Bosch & Gómez, 2016; Bosch et al., 2016a) are also publicly available[13].

2.3.3 Multiple pitch estimation

Multiple pitch estimation is one of the core problems in audio signal processing. For speech, it is useful for the recognition of multiple talkers (Cooke et al., 2010) or for prosody analysis (Jiang et al., 2005). In MIR, it is useful for many tasks such as automatic music transcription (Klapuri et al., 2006), source separation (Marxer, 2013; Duan et al., 2008) or melody extraction (Han & Chen, 2011).

In MIREX, multipitch analysis is evaluated at three levels. The first is to estimate pitch values of all concurrent sources at each individual time frame (multipitch estimation). A second level is note tracking, in which the task is to estimate continuous segments which correspond to notes. Most approaches use continuity in the time-frequency plane to connect pitch estimates. The last and more complicated level is to stream pitch estimates into a single pitch trajectory for the whole song (or excerpt), for each of the concurrent sources. Multipitch methods commonly calculate a pitch salience (or pitch activation) function, and then perform refinement or tracking to smooth pitch trajectories. However, most research has focused on the estimation of pitches (first two levels), and only few approaches have dealt with their assignment to different sources (third level).

Multipitch estimation algorithms have been categorised (Yeh, 2008) as joint or iterative. The former aim at a joint estimation of all pitches, while the latter extract the most prominent pitch in each iteration, until no additional f_0 can be estimated. The iterative estimation sometimes introduces errors which are propagated into the following iterations. Multipitch estimation approaches have also been categorised depending on the domain on which they operate: time domain, frequency domain, or a hybrid domain (similarly to single pitch estimation methods). The representation most commonly used in the time domain is the raw waveform, and auditory filterbanks. In the frequency domain, methods have employed Short-Time Fourier Transform (STFT) spectrum, Constant-Q Transform (CQT) spectrum, Equivalent Rectangular Bandwidth (ERB) filterbanks or specmurt representations. Benetos (2012) also classifies the algorithms based on their core approach, e.g. signal processing, maximum likelihood, Bayesian, spectrogram decomposition, sparse coding, rule-based, classification-based, etc.

Some time domain approaches use autocorrelation function (ACF) (Tolonen & Karjalainen, 2000), which sometimes have difficulties when multiple pitches are present since the periodicity is unclear. Other time domain approaches have been proposed, e.g. based on oscillators, which adapt their frequency and phase to input signal (Marolt, 2004a), or on probabilistic modelling (Walmsley et al., 1999; Davy & Godsill, 2003; Cemgil et al., 2006).

[13] http://www.mtg.upf.edu/node/3737

	Processing+Spectral Transform	Multipitch Representation	Pitch Grouping	Tracking method	Voicing	Type
Paiva et al. (2006)	Auditory model + autocorrelation peaks	Summary Correlogram	•	Multipitch trajectories + note deletion	Salience valleys	Salience-based
Marolt (2004b)	STFT + SMS harmonics plus noise	EM fit to tone models	•	Fragments + fragment clustering	Loudness filter	Salience-based
Goto (2004)	Bandpass filter + wavelet tranform	EM fit to tone models	•	Tracking agents	N/A	Salience-based
Ryynänen & Klapuri (2008)	STFT + spectral whitening	Harmonic Summ	•	Note event HMM + global HMM	Silence model	Salience-based
Dressler (2012b)	MRFFT + IF peak correction + magnitude threshold	Peak comparison	•	Heuristic rules	Adaptive threshold	Salience-based
Fuentes et al. (2012)‡	CQT	PLCA on the CQT		Viterbi	Energy threshold	Salience-based
Salamon & Gómez (2012)†	ELF + STFT	Harmonic Summ	•	PCS	Salience threshold	Salience-based
Durrieu et al. (2010)‡	STFT	NMF on S/F model		Viterbi	energy threshold	Salience + separation
Yeh et al. (2012)	HPSS + MRFFT + partial discrimination	Subharmonic summ.		Trend estimation + HMM	N/A	Salience + separation
Poliner & Ellis (2005)‡	STFT	Learned		Viterbi	Probability threshold	Data-driven
Bittner et al. (2015)‡	ELF+ STFT	Harmonic Summ	•	PCC	Probability threshold	Data-driven (Tracking)
Kum et al. (2016)‡	Resampling+STFT	Learned		Viterbi	Energy Threshold	Data-driven
Rigaud & Radenen (2016)	STFT+ HPSS	Learned		Viterbi	DNN	Data-driven
Verma & Schafer (2016)	N/A	N/A		N/A	N/A	Data-driven

Table 2.1: Summary of relevant melody extraction methods. N/A: Not applicable. ‡: source code available, †: binary available (links provided in Section 2.3.2).

Frequency domain approaches commonly try to recognize the harmonic patterns of each of the pitches present in the signal. In this case, the difficulty is that harmonics of different pitches in a musical signal typically overlap, and their amplitude varies along time. An example of an iterative, frequency domain approach is Klapuri (2003), which estimates the predominant pitch and subtracts the harmonics, and continues with the estimation of other pitches iteratively, while estimating the total number of sounds present. One of the difficulties is to know the amount of energy to substract for each of the pitches. Duan et al. (2010) estimate the pitches present with a Maximum Likelihood (ML) approach assuming spectral peaks at harmonic positions and lower energy elsewhere. They then employ a neighbourhood refinement method to create a pitch histogram in the vicinity of a frame to eliminate transient estimations, as well as to refine the polyphony estimation. Benetos & Dixon (2011) use Shift-Invariant Probabilistic Latent Component Analysis (SIPLCA), which is able to support multiple instrument models and pitch templates. Dressler (2012a) uses a salience function based on the pair-wise comparison of spectral peaks, and streaming rules for tracking.

Note-tracking methods may perform a post processing of the frame-based estimation, based on: e.g. thresholding, deletion of short notes, minimum duration pruning (Böck & Schedl, 2012; Fuentes et al., 2013; Carabias-Orti et al., 2011; Bertin et al., 2010; Dessein et al., 2010), an HMM (Ryynanen & Klapuri, 2005; Benetos & Dixon, 2011), or median filtering (Su & Yang, 2015), while some methods consider interactions between simultaneous pitches (Duan & Temperley, 2014). Other methods perform onset detection, followed by multipitch estimation between onsets (Marolt, 2004a; Emiya et al., 2010; P. Grosche et al., 2012; Cogliati & Duan, 2015), and finally there are methods that attempt to estimate notes directly from audio (Kameoka et al., 2007; Berg-Kirkpatrick et al., 2014; Ewert et al., 2015).

Further details about pitch salience functions and methods that compute them are discussed next.

2.4 Pitch salience estimation

Several names have been used in the literature to refer to what we here call pitch salience function: e.g. $f_0$gram, pitch activation function, pitch likelihood function, pitch strength function or simply multipitch representation. Essentially, they all deal with the same concept: the representation of the salience of pitches over time, which is related to the likelihood of them being present in the acoustic signal. Pitch salience functions ideally only contain peaks at the frequencies corresponding to the pitches present at a given instant. In the case of melody oriented pitch salience functions, the melody pitch should ideally be much more salient than the rest of pitches.

One of the problems of salience functions is that they assign high salience to pitches which are actually not present in the signal. In particular, the f_0 of multiples and submultiples is often salient, which may produce octave errors. In order to reduce

them, most algorithms rely on melody contour smoothness while doing pitch tracking. However, some algorithms also deal with this problem at an earlier stage, during pitch salience function estimation.

In this thesis, we focus on salience functions based on harmonic summation and source-filter models, but we also review other methods such as spectrogram decomposition methods or neural networks.

2.4.1 Preprocessing

The computation of pitch salience commonly starts with a time-frequency transformation such as the Short-Time Fourier Transform (STFT) (Salamon & Gómez, 2012; Durrieu et al., 2011; Marxer, 2013; Duan et al., 2010; Arora & Behera, 2013), multi-resolution transforms (MRFFT) (Dressler, 2012b) or constant-Q transform (CQT) (Cancela et al., 2010; Fuentes et al., 2012; Benetos & Dixon, 2011). Some of the approaches perform a pre-processing step such as Equal-Loudness Filters (ELF) (Salamon & Gómez, 2012; Marxer, 2013), or a posterior step like frequency refinement (Salamon & Gómez, 2012).

In the case of Salamon & Gómez (2012), the method uses the short-time Fourier transform (STFT)

$$X_l(k) = \sum_{n=0}^{M-1} w(n) \cdot x(n + lH) e^{-j\frac{2\pi}{N}kn} \tag{2.1}$$

where $l = 0, 1, \ldots$ and $k = 0, 1, \ldots, N-1$, $x(n)$ is the sampled input signal, $w(n)$ the windowing function, l the frame number, M the window length, N the FFT length, H the hop size, and a zero padding factor of x4. A multi-resolution transform did not improve the results on their evaluation. Then, they obtain the frequency and amplitude of the spectral peaks from which are then corrected, using instantaneous frequency refinement (or parabolic interpolation in the implementation in Essentia[14]) (Bogdanov et al., 2013).

2.4.2 Harmonic summation

One of the most commonly used methods for pitch salience estimation is harmonic summation (Klapuri, 2006), a frequency domain approach which computes the salience of each pitch by summing the energy of the spectrum bins which contribute to that pitch, weighted by the strength of their contribution. This approach is computationally inexpensive and has been used successfully in a variety of forms for predominant melody extraction (Salamon & Gómez, 2012; Dressler, 2012b) as well as multiple pitch estimation (Dressler, 2012a). Most algorithms use only spectral peaks to compute the salience function, unlike in (Klapuri, 2006), which is computed

[14] https://github.com/MTG/essentia

from the whole spectrum. This allows to discard the contribution from spectral elements which are less likely to correspond to the melody pitch (e.g noise, percussive elements, masked components). Dressler (2011) attempts to reduce the number of octave errors, by examining pairs of spectral peaks which potentially belong to the same harmonic series. If many spectral peaks with frequencies lying between the pair being considered have high amplitude, the result of their summation is attenuated. Cancela et al. (2010) proposes to attenuate the sum of harmonics at a certain f_0 if the mean amplitude of spectral components at frequencies $2k \cdot f_0$, $3k \cdot f_0/2$ and $3k \cdot f_0$ is above the mean of the components at frequencies $k \cdot f_0$. Note that this attenuates pitches whose f_0 is 1/2, 2/3 or 1/3 of the real f_0.

Due to its relevance in this thesis, we detail the approach by Salamon & Gómez (2012). Harmonic summation is computed from the peaks of the spectrum, and therefore it is possible to perform peak frequency refinement, improving the frequency accuracy of the salience function (Salamon et al., 2011). Instead of searching for energies at integer multiples of a candidate f_0, the salience is computed as a subharmonic summation (Hermes, 1988). The energy of each of the detected spectral peaks (p_i) is mapped (using a weighting scheme) to the frequencies of which p_i could be a harmonic partial, such as p_i/h, where $h = 1,2,..,N_h$ is an integer value which represents the harmonic number of f with respect to a candidate $f_0 = f/h$. Two factors affect the computation: number of harmonics considered N_h and the weighting scheme used. The salience function goes from $f_{min} = 55Hz$ to $f_{max} = 1.76kHz$ (around 5 octaves), corresponding to 1-600 bins on a cent scale (10 cents per bin). For a given frequency value $\hat{f}$ (in Hz), the corresponding bin is calculated as:

$$B(\hat{f}) = \left\lfloor \frac{1200 \cdot \log_2\left(\frac{\hat{f}}{f_{min}}\right)}{10} + 1 \right\rfloor \tag{2.2}$$

At every frame, the salience function $S(b)$ is computed from the previously computed peaks (p_i) (with frequencies $\hat{f}_i$ and linear magnitudes $\hat{a}_i$) ($i = 1 \cdots, I$, where I is the number of peaks found). The salience function is defined as:

$$S(b) = \sum_{h=1}^{N_h} \sum_{i=1}^{I} e(\hat{a}_i) \cdot g(b,h,\hat{f}_i) \cdot (\hat{a}_i)^{\beta} \tag{2.3}$$

where β is a magnitude compression parameter, $e(\hat{a}_i)$ is a magnitude threshold function, and $g(b,h,\hat{f}_i)$ defines the weighting scheme. The definition of the magnitude threshold function is:

$$e(\hat{a}_i) = \begin{cases} 1, if \;\; 20\log_{10}(\hat{a}_M/\hat{a}_i) < \gamma \\ 0, otherwise \end{cases} \tag{2.4}$$

where $\hat{a}_M$ corresponds to the magnitude of the highest spectral peak in the frame, and γ is the maximum allowed difference (in dB) between $\hat{a}_i$ and $\hat{a}_M$.

The weighting function defines the weight given to a peak with amplitude $\hat{a}_i$, when it is considered as the h^{th} harmonic of the f_0 corresponding to bin b:

$$g(b,h,\hat{f}_i) = \begin{cases} \cos^2\left(\delta \cdot \frac{\pi}{2}\right) \cdot \alpha_w^{h-1}, if|\delta| \leq 1 \\ 0, if|\delta| > 1 \end{cases} \tag{2.5}$$

where $\delta = |B(\hat{f}_i/h) - b|/10$ is the semitone distance between the frequency of the harmonic $\hat{f}_i/h$ and the centre frequency of bin b, α_w is the harmonic weighting parameter. This results in the contribution of each peak to more than a single bin in the salience function (with a cos^2 weight), to avoid problems on the quantisation and inharmonicity issues.

2.4.3 Spectrogram factorisation methods

Probabilistic approaches based on decomposition models such as Non-negative Matrix Factorisation (NMF), or Probabilistic Latent Component Analysis (PLCA) have gained interest, especially within source separation (Marxer, 2013; Durrieu et al., 2011), and music transcription scenarios (Benetos & Dixon, 2011; Carabias-Orti et al., 2011; Smaragdis & Brown, 2003). Many methods have been proposed in the literature to perform melody extraction and multiple pitch estimation using these learning techniques with different time-frequency representations, transcription models and post processing steps.

NMF is a subspace analysis method which is able to decompose an input time-frequency representation into a basis matrix with spectral templates for each component and a component activity matrix over time. Lee and Seung (Lee & Seung, 1999, 2001) popularised NMF in the field of image processing and clustering. NMF has also been used for audio and music processing, e.g. by Smaragdis & Brown (2003) and Virtanen (2007), among many others. PLCA (Smaragdis, 2004; Smaragdis et al., 2006) is as a probabilistic extension of the non-negative matrix factorization (NMF) algorithm using the Kullback-Leibler cost function (Kullback & Leibler, 1951). It provides a framework that is easy to generalize and interpret, and it can incorporate priors over the parameters and control the resulting decomposition. In both cases, it is possible to use either pre-extracted or estimated spectral templates (using parametric spectral models).

PLCA and NMF have also been employed for melody extraction. Fuentes et al. (2012) uses PLCA on a CQT to build a pitch salience function for a melody extraction (and separation) approach. The CQT of the signal is modelled as the sum of two CQTs, where the accompaniment is modelled with a standard PLCA. The melody spectrum is modelled in order to account for the non-stationary nature of pitch and spectral envelope of many musical instruments, especially human voice. The model is a weighted sum of fixed narrow-band harmonic spectral kernels, spectrally convolved by a time-frequency impulse distribution. After an initial estimation of the impulse distribution

(pitch salience function) for the whole excerpt, the pitch sequence path is found using a Viterbi algorithm (Forney, 1973), and pitches farther than a semitone from the path are set to zero. The estimation is applied again for a few iterations in order to let parameters converge to a new solution. It is then possible to separate melody and accompaniment applying time-frequency masks and then computing the inverse CQT. Durrieu et al. (2010) also use NMF for modelling the accompaniment. In that case, no fixed basis are used, and ideally, repetitions in the accompaniment could be captured for a better estimation of the lead source, as further detailed in Section 2.4.4.

Many methods using PLCA for music transcription have been proposed (Benetos & Dixon, 2011, 2012, 2013; Benetos et al., 2014). Benetos & Dixon (2011) proposed a convolutive probabilistic model, which extended the Shift-Invariant Probabilistic Latent Component Analysis method (SIPLCA). Shift-invariance is present by using constant-Q transform as a time-frequency representation, which provides a better support for tuning changes and frequency modulations (e.g. vibrato). Instrument basis are extracted for various instruments, for each note, using their whole note range. These are kept fixed during the estimation of the activations. One of the drawbacks of using fixed basis is that they do not commonly correspond to the spectral shape of the sources from the analysed music piece. Benetos et al. (2014), proposed a strategy for template adaptation, by first extracting the spectral shape of notes detected with high confidence, in a conservative transcription pre-processing step. Transcription is then performed with the new set of templates. Results in terms of multipitch detection and instrument assignment show consistent improvements in contrast with keeping the dictionary fixed, when evaluated on MAPS and Bach10 databases (for a description of these datasets, please refer to Section 2.7.5.2). To overcome the computational bottleneck of convolutive models, Benetos & Weyde (2015a) propose the use of a 5-dimensional dictionary of pre-extracted and pre-shifted sound state spectral templates, using variable-Q transform (VQT) as time-frequency representation. Two variants are presented: with HMM-based constraints controlling the appearance of sound states, or without any temporal constraints. Benetos & Weyde (2015b) presented a similar method in MIREX 2015, but using an Equivalent Rectangular Bandwidth (ERB) scale time-frequency representation, instead of VQT. ERB offers a compact representation, at the cost of losing the shift-invariance abilities, due to the non-linearity with respect to log-frequency. Results showed improvements in comparison to the previous submission to MIREX, based on VQT, which had in turn improved over a previous submission based on CQT.

2.4.4 Source-filter models

Source-filter models are used to decouple (to some extent) timbre information from pitch information. Source-filter models have been used in the context of source separation (Bouvier et al., 2016; Durrieu et al., 2010), musical instrument recognition (Heittola et al., 2009) or transformation (Caetano & Rodet, 2012), music signal reconstruction (Cheng et al., 2014), and also to create an intermediate pitch representation

(salience function), useful for melody extraction. Durrieu et al. (2010, 2011)[15] proposed an unsupervised method based on a Smoothed Instantaneous Mixture Model (SIMM) to model the leading voice, and apply Non-negative Matrix Factorisation (NMF) to create a melody-oriented pitch salience representation. The spectrum X of the signal is modelled as the lead instrument plus accompaniment $\hat{X} = \hat{X}_v + \hat{X}_m$. The lead instrument is modelled as: $\hat{X}_v = X_\Phi \circ X_{f_0}$, where X_{f_0} corresponds to the source, X_Φ to the filter, and the symbol $\circ$ denotes the Hadamard (element-wise) product. Both source and filter are decomposed into basis and gains matrices as $X_{f_0} = W_{f_0} H_{f_0}$ and $X_\Phi = W_\Phi H_\Phi$ respectively. The filter basis matrix W_Φ is further decomposed into a weighted sum of smooth spectral atoms: $W_\Gamma H_\Gamma$. H_{f_0} corresponds to the pitch activations of the source, which can also be understood as a representation of pitch salience (Durrieu et al., 2011). The accompaniment spectrum is modelled as: $\hat{X}_m = \hat{W}_m \hat{H}_m$, leading to Equation 2.6.

$$X \approx \hat{X} = (W_\Gamma H_\Gamma H_\Phi) \circ (W_{f_0} H_{f_0}) + W_m H_m \tag{2.6}$$

Marxer (2013) follows a similar strategy as Durrieu et al. (2011), but uses a Tikhonov Regularisation (TR), which is computationally cheaper and allows low-latency processing.

Two examples of pitch salience functions in a context of symphonic music are shown in Figure 2.2. The plot at the top corresponds to the approach by Salamon & Gómez (2012), implemented in the VAMP plugin MELODIA[16]. As it can be observed, there is no clearly salient melodic line using this salience function, since multiple pitches have similar salience values in any given frame. This suggests that symphonic music is especially challenging for melody extraction algorithms based on harmonic summation. The plot at the bottom corresponds to the pitch salience computed with the approach by Durrieu et al. (2011), which is visibly much sparser. Due to the wide range of values obtained with the latter approach, the salience function has been normalised per frame, for a better visualisation.

2.4.5 Neural networks

As previously introduced in Section 2.3.2, deep learning has been used for melody extraction, and one approach based on melody-oriented pitch salience estimation was recently presented. Rigaud & Radenen (2016) proposed a Deep Neural Networks (DNN) approach, in which the output layer returns a f_0 probability distribution for each time frame, which can be considered as a pitch salience (or activation) matrix. The method exhibits higher pitch estimation accuracy than MELODIA (Salamon & Gómez, 2012) on two different music databases, but is only evaluated in the context of vocal data.

[15] https://github.com/wslihgt/IMMf0salience
[16] http://mtg.upf.edu/technologies/melodia

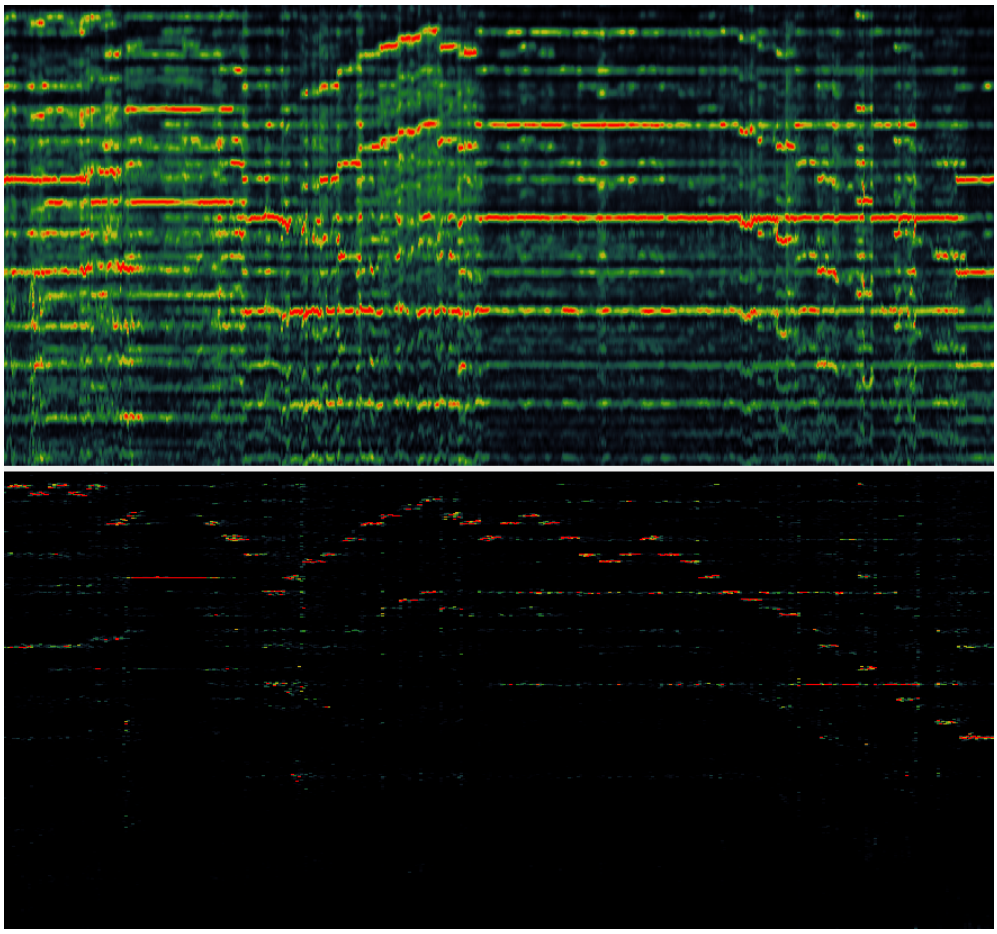

Figure 2.2: Pitch salience functions estimated from an excerpt of the 1st movement of Beethoven's 3rd symphony. They were computed with MELODIA (top) and Durrieu's approach (bottom), as VAMP plugins in Sonic Visualiser. The vertical axis corresponds to the frequency between 55 and 1760 Hz, in logarithmic scale. Horizontal axis corresponds to time, from 0 to 10 seconds. Both salience functions are normalised per frame, for a better visualisation.

Neural networks have also been used in the context of multiple pitch estimation and transcription. Several works have made use of neural networks for polyphonic piano transcription. Marolt (2004a) used neural networks for piano notes in a classification-based transcription method, in which each quantized pitch is a class. The input to the networks were the output values of oscillator networks. Best results were obtained with time-delay neural networks (TDNNs). Nam et al. (2011) trained deep belief networks using spectrogram bins as features, with both single notes and note combinations. Böck & Schedl (2012) employed recurrent neural networks for polyphonic piano transcription, using the output of two semitone filterbanks with different window frame sizes as features. A bidirectional long short-term memory (BLSTM) neural network was employed for note classification and onset detection. Sigtia et al. (2016) have also recently proposed an end-to-end neural network with an acoustic model and a music language model. Pertusa & Iñesta (2005) also previously used TDNNs for polyphonic music transcription, using pre-processed STFT bins as input. According to its title, the best performing algorithm in the MIREX multiple pitch estimation task is also based on neural networks (Elowsson & Friberg, 2014), but the details of the implementation have not been published at the time of publishing this thesis.

2.4.6 Multi-resolution fan chirp transform

Cancela et al. (2010) proposed a multi-resolution Fan Chirp Transform (FChT), which provides an acute representation of harmonically related linear chirp signals. The approach is based on a time warping followed by a Fourier Transform (Constant Q-Transform is also possible). Since the method is intended for main melody extraction, a pitch preference function is applied after the salience is computed (which can be disabled), to emphasise the pitches in a certain frequency range using a Gaussian function[17].

2.5 Melody pitch tracking

After the computation of pitch salience, many melody extraction and multipitch estimation methods incorporate perceptual principles or additional acoustic and musical knowledge (timbre, harmonicity, spectral smoothness, etc.) to separate partials and group salience peaks into streams, or even map them to a given pitched source. Melody extraction methods commonly rely on the predominance of melody pitches and smoothness in the melody trajectory for melody pitch tracking,

Many different approaches have been proposed at this stage, tracking peaks of the salience function directly (Durrieu et al., 2010; Fuentes et al., 2012; Marxer, 2013), or grouping them into pitch contours (fragments or trajectories) Salamon & Gómez (2012); Paiva et al. (2006); Cancela (2008). Durrieu et al. (2010) followed the first

[17] http://iie.fing.edu.uy/investigacion/grupos/gpa/fcht.html

approach, using a HMM in which each state corresponds to one of the frequency bins of the salience function, and the probability of each state corresponds to the estimated source activations (salience) (H_{f_0}). Pitch continuity is considered in the transition probabilities, favouring smoothness in pitch trajectories. Ryynänen & Klapuri (2008) proposed the use of a HMM derived for note events from fundamental frequencies, their saliences and an accent signal. Marxer (2013) proposes to model the peaks in the pitch likelihood (salience) function as Gaussian functions, and use the divergence between them as a transition probability in the HMM of the tracking stage, in an online scenario. Fuentes et al. (2012) used Viterbi smoothing to estimate the melody trajectory within a melody extraction method based on PLCA.

Pitch grouping has also been employed for melody tracking and have additionally proven to be a useful mid-level pitch representation for other MIR tasks. It can also help to reduce octave errors, since duplicate contours in this case have parallel trajectories with 12 semitone difference, and it is easier to detect and eliminate them. Salamon & Gómez (2012) base the decision on which is the duplicate on melody contour smoothness and contour salience. Grouping is commonly performed using time and pitch continuity principles inspired form ASA. Due to the importance in this thesis, we detail a subset of melody extraction methods based on pitch grouping, which use pitch contour selection with heuristic rules (Salamon & Gómez, 2012), or pitch contour classification (Bittner et al., 2015; Bosch et al., 2016b). Both types of methods are based on the creation of and characterisation of pitch contours.

2.5.1 Pitch contour formation

Pitch contours are groupings of fundamental frequencies which are continuous over pitch and time. A pitch contour corresponds to the time series $c(t) = (f(t), s(t))$ which are defined along a discrete, finite time interval $\{t_0, t_1, ..., t_n\}$, where $f(t)$ is the f_0 of the contour over time, and $s(t)$ corresponds to the "salience" of the contour over time. A given contour is thus only defined over the time interval $[t_0, t_n]$. In the contour formation process, there are several parameters which greatly affect the final output, in terms of amount, length and contour shape.

From a given pitch salience function, Salamon & Gómez (2012) form pitch contours by grouping continuous sequences of salience peaks. Several parameters need to be set (default values used in Salamon & Gómez (2012) are presented between brackets). The initial step is the removal of non-salient melody peaks per frame: peaks below a threshold factor τ_+ (0.9) of the highest salience peak in the frame are filtered out. Secondly, remaining peaks are filtered if their salience is below $\mu_s - \tau_\sigma \cdot \sigma_s$, where μ_s and σ_s are the mean and standard deviation of the salience of remaining peaks (in all frames). τ_σ (0.9) determines the accepted degree of deviation below mean salience. The first filter ensures the predominance of the remaining salience peaks in a given frame, while the second filter helps reducing voicing false alarms.

Contours are then created by grouping peaks which are close in time and frequency,

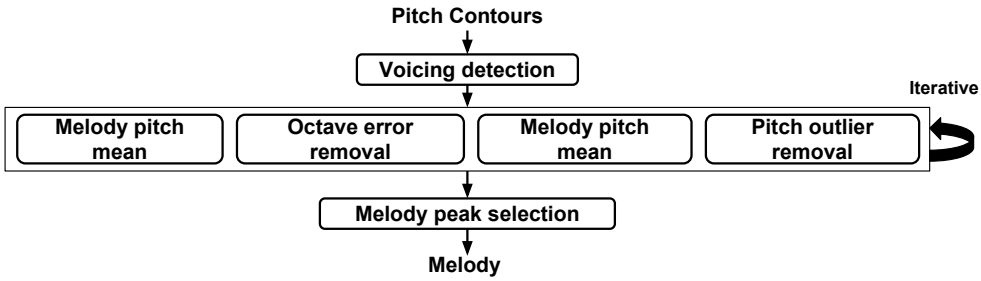

Figure 2.3: Melody tracking based on pitch contour selection

with several parameters. The first parameter refers to the minimum allowed contour duration (100 ms.): contours shorter than this value will be filtered. The second parameter is the maximum allowed pitch change during 1 ms time period (27.56 cents), and finally the maximum allowed gap duration in a pitch contour (100 ms.). We refer the reader to (Salamon & Gómez, 2012) for a detailed description of the contour formation process.

2.5.2 Pitch contour characterisation

Pitch contours have been a useful mid-level pitch representation for several tasks which involve the computation of pitch related information. Previous approaches are based on the computation of pitch contour features, and use them for the extraction of melody (Salamon et al., 2012a; Salamon & Gómez, 2012; Bittner et al., 2015) or bass, but also for genre classification (Salamon et al., 2012b), or for cover-song identification (Salamon et al., 2013).

Salamon & Gómez (2012) characterises contours with the following set of features: pitch (mean and deviation), salience (mean, standard deviation and sum), duration, and vibrato related features: presence (binary), rate (Hz), extent (cents) and coverage (fraction of contour with vibrato). These pitch contour characteristics are used for the selection of melody contours in methods based on pitch contour selection and pitch contour classification.

2.5.3 Pitch contour selection

Pitch contour selection (PCS) (Salamon & Gómez, 2012) deals with the tracking of the melody pitch by selecting melody contours following heuristic rules. The process is divided in three tasks: voicing detection, octave error minimisation/pitch outlier removal, and final melody selection, as shown in Figure 2.3.

Previously calculated contour features are used in this stage to filter out non-melody contours, using a voicing detection threshold τ_v, which is based on the salience distribution of the created contours: $\tau_v = \overline{C_s} - \nu \cdot \sigma_{C_s}$ where $\overline{C_s}$ and σ_{C_s} are the mean and

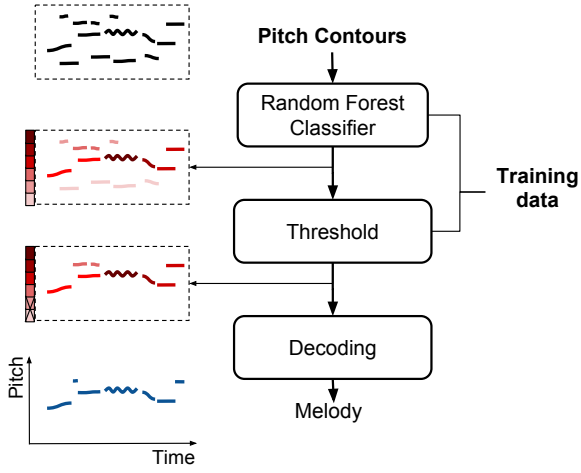

Figure 2.4: Melody tracking based on pitch contour classification

standard deviation of the salience of the contours. Parameter ν controls the amount of contours that are filtered out, set by default to 0.2 in (Salamon & Gómez, 2012).

2.5.4 Pitch contour classification

Salamon et al. (2012a) and later Bittner et al. (2015) also proposed pitch contour classification (PCC) methods based on contour features. The former uses a generative model based on multi-variate Gaussians to distinguish melody from non-melody contours, and the latter uses a discriminative classifier (a binary random forest) to perform melody contour selection. However, these classification-based approaches did not outperform the rule-based approach on MedleyDB (Bittner et al., 2015). One of the important conclusions of both papers was that the sub-optimal performance of the contour creation stage (which was the same in both approaches) was a significant limiting factor in their performance.

Since the supervised melody extraction approaches which we propose in this thesis are based on pitch contour classification as proposed by Bittner et al. (2015), we provide here further details. This method takes as input the pitch contours created by an intermediate step in the MELODIA vamp plugin[18] (Salamon & Gómez, 2012), and learns a Random Forest Classifier to discriminate melody from non-melody contours. After filtering contours with low likelihood to be melodic, remaining contours are decoded using the Viterbi algorithm. A basic schema of the process is shown in Figure 2.4.

To train the classifier, contours in the training dataset are labelled as being melodic or non-melodic, according to the amount of overlap between each contour and the

[18] http://mtg.upf.edu/technologies/melodia

ground truth annotation. The overlap corresponds to the Overall Accuracy measure in the frames where the pitch contour is present, using two pitch sequences: contour pitches as the estimation, and the annotated pitches as ground truth. If the overlap is higher that a threshold value (α^{th}), the contour is labelled as melody. Note that the higher the value of α^{th}, the less amount of contours will be used for training.

After assigning a class to each of the contours from the training set, the classifier is trained using contour features described in Section 2.5.2. Prior to the training, features are normalised per track to remove variance caused by differences between tracks. Salience features are each divided by the maximum salience value in the track, and duration is normalized between 0 and 1 for each track. Finally, total salience is rescaled to reflect the normalized duration. Scikit-learn (Pedregosa et al., 2011) is used to train a Random Forest Classifier (Breiman, 2001) with 100 trees. The maximum depth of the trees is computed by cross validating over the training set.

After training, the classifier is used to compute the probability of each contour belonging or not to the melody. The prediction is based on computing the fraction of trees from the Random Forest Classifier which classifies the contour as melodic, which provides a number from 0 to 1 (1 corresponding to a melodic contour).

Finally, the decoding stage generates a pitch sequence using the predicted contour probabilities. The first step is filtering contours whose melodic likelihood is below a threshold value, which is the value that maximises the class weighted F1 score on a validation set (F1-measure corresponds to the harmonic mean of precision and recall). From the remaining contours, those without an overlap with other contours are immediately assigned to the output melody. Then, the rest of the contours are divided into groups. Contours are assigned to the same group if the union of their intervals forms a contiguous interval. Finally, Viterbi decoding (Forney, 1973) is used in each of the groups to find the most likely path over time, where the state space is the set of contour numbers. The prior distribution is set to be uniform and the emission probability matrix is created using each contour's likelihood score. The transition matrix encourages continuity in pitch space, thus assigning a higher probability to transitions between contours whose (log) frequencies are close to each other. The most likely sequence of contours along time is used to assign their associated pitches to the output melody sequence.

2.5.5 Multiple pitch tracking

The tracking of multiple pitches in music has been used for multiple pitch estimation and transcription, but also in the context of melody extraction.

Rao & Rao (2009) proposed to dynamically track f_0-candidate pairs, generated by imposing specific harmonic relation-related constraints. This allowed to deal with the problem of having concurrent melody lines with similar salience. Results showed a better voice tracking, partially decreasing the amount of incorrect estimations of the accompanying instrument pitch as the melody, in the context of Indian classical

vocal music and Indian film music. However, this approach does not ensure that the two melody lines remain linked to their respective sound sources. Dressler proposed an streaming approach which mimics some characteristics of stream segregation in the human auditory system, used for both melody extraction (Dressler, 2011) and multiple pitch estimation (Dressler, 2012a). It is based on heuristics which consider the magnitude of tones, timbre-related information, note intervals and playback speed.

Some supervised methods create timbre models of sound sources, and then apply them for pitch estimation (Cont et al., 2007; Bay et al., 2012; Benetos & Dixon, 2013; Kirchhoff et al., 2013). Other approaches categorise the estimated pitches or notes in instrument classes (Wu et al., 2011), and some methods adapt the spectral shapes to the content of the analysed music (Carabias-Orti et al., 2011; Grindlay & Ellis, 2011; Benetos et al., 2014).

Unsupervised approaches do not classify pitches or notes into classes corresponding to instruments, but they cluster estimated pitches according to some timbre features (Duan et al., 2014; Mysore & Smaragdis, 2009; Arora & Behera, 2015). Duan et al. (2014) propose a constrained clustering method, with an objective function based on timbre consistency, and time-frequency locality constraints. The algorithm minimizes the objective function while satisfying as many constraints as possible. They also introduce a cepstrum feature for representing timbre in multi-source mixtures. Kirchhoff et al. (2013) identify prominent pitches in each time frame, and assign them to instruments in the mixture, using the Viterbi algorithm to find the most likely path through the candidate instrument and pitch combinations in each time frame. The transition probability is affected by: the frame-wise reconstruction error of the instrument combination, a pitch continuity measure, and the activity status of each instrument.

2.6 Voicing and polyphony estimation

Melody extraction algorithms have to classify frames as containing a melody pitch or not. Due to historical reasons, and the fact that most research has been conducted on vocal melodies, most melody extraction literature denotes as voiced those frames that contain a melody pitch, regardless of the instrument producing it. Unvoiced frames are therefore those which do not contain a melody pitch. In this thesis we follow the same naming convention.

The simultaneous estimation of pitch and voicing in music signals is a complex task. In the case of speech signals mixed with background noise, it is easier to discriminate a frame of speech and one containing noise, possibly due to the presence of a pitched structure. Therefore, there have been approaches which jointly estimate pitch and the presence of human voice, using for instance a single DNN (Lee & Ellis, 2012; Han & Wang, 2014). In the case of music signals, both melody and accompaniment contain harmonic pitched structures, and it is thus necessary to use some other information to

distinguish between them. Due to this complexity, voicing detection is is commonly performed on a separate step.

Most melody extraction approaches use static or dynamic thresholds on e.g. energy or salience (Fuentes et al., 2012; Durrieu et al., 2010; Dressler, 2012b; Arora & Behera, 2013). Salamon & Gómez (2012) follow a different strategy, and exploit pitch contour salience distributions. Bittner et al. (2015) determine voicing by setting a threshold on the contour probabilities produced by the discriminative model.

Separation-based approaches perform voicing detection in different ways. For instance, Durrieu et al. (2010) estimate the energy of the melody signal frame by frame. Frames whose energy falls below the threshold are set as unvoiced and vice versa. The threshold is empirically chosen, such that voiced frames represent more than 99.95% of the leading instrument energy. Fuentes et al. (2012) also use an energy threshold (of -12dB) on a low-pass filtered separated melody signal. In the case of Tachibana et al. (2010) voicing detection is also performed with a threshold, but in this case it is applied on the (Mahalanobis) distance between the estimated melody signal and the percussive signal.

Singing voice detection (SVD) is a very similar task, which aims at identifying the regions in a music recording where at least one person sings, for which timbral and temporal characteristics are commonly exploited. In comparison with melody extraction, this task is restricted to the vocals, and pitch does not need to be identified.

Mauch et al. (2011) proposed the use of standard MFCCs, and three features based on the extracted melody line: pitch fluctuation, MFCCs of the re-synthesized predominant voice, and the relative harmonic amplitudes of the predominant voice. A different approach was taken by Rao et al. (2013), who used the differences in singing style and instrumentation across genres to adapt acoustic features for this task. Lehner et al. (2014) proposed the Vocal Variance (VV) feature, which computes the variance of the first five MFCCs (Davis & Mermelstein, 1980; Logan et al., 2000), calculated over a window of around 800ms around the current frame. Lehner et al. (2015) also proposed a real-time approach using LSTM neural networks for SVD. Rigaud & Radenen (2016) propose a neural network approach for SVD based on a similar approach by Leglaive et al. (2015), using Bidirectional Long Short-Term Memory (BLSTM). Mel-frequency spectrograms were computed from pre-decomposed signals using Harmonic/Percussive separation, and given as input to 3 BLSTM layers of 50 units each, and a final feed-forward logistic output layer with one unit. The binary decision of voice detection is taken with a threshold (0.5) on the DNN output. They combined this SVD approach with the previously mentioned DNN for pitch salience estimation, achieving state-of-the-art overall accuracy on vocal data. A different approach is the algorithm by Ryynänen & Klapuri (2008), which incorporates a silence model into the HMM tracking part of the algorithm. Hsu & Jang (2010) also proposed the use of timbre to classify frames as containing human voice or not.

In the context of multipitch estimation, a related task is polyphony estimation, that is,

to estimate the amount of different concurrent sounds. This is a complex task even for musicians, who commonly underestimate the number of voices, when listening to four voice polyphonies employing homogeneous timbre (Huron, 1989). There are different approaches to this problem. Many methods apply a threshold commonly based on pitch salience / likelihood, either fixed (Benetos & Dixon, 2011), or dynamic (Dressler, 2012a). Klapuri (2003) proposed a statistical-experimental approach to control the stopping of the iterative f_0 estimation and sound separation process. Yeh et al. (2010) presented an approach based on STFT, with an adaptive noise level estimation method. Then, given a set of pitch candidates, the overlapping partials are detected and smoothed according to the spectral smoothness principle. Polyphony estimation is based on the increase of a score function using harmonicity, mean bandwidth, spectral centroid, and synchronicity features. Duan et al. (2010) also performed polyphony estimation, in order to control the number of iterations of the method using a threshold-based method on the likelihood function. The likelihood function is composed of the peak region likelihood (probability that a peak is detected in the spectrum given a pitch) and the non-peak region likelihood.

2.7 Evaluation strategies

This section introduces the metrics used for the evaluation of melody extraction and multipitch estimation algorithms. Given the importance of pitch salience functions, we also present state-of-the-art metrics for evaluating them in the context of melody extraction. Finally we present the datasets used in the MIREX evaluation campaign, as well as other related publicly available collections.

2.7.1 Pitch salience function evaluation

Salience functions are evaluated commonly evaluated from two different perspectives: pitch and salience estimation accuracy. Salamon et al. (2011) proposed four different metrics using the ground truth melody pitch. First, salience function peaks are computed, and then the peak closest to the ground truth is selected, and considered as the melody salience peak. The first metric is the frequency error of the salience function Δf_m, computed as the difference (in cents) between the frequency of the melody salience peak and the ground truth f_0. The following metrics deal with salience estimation. The first metric (RR_m) is the reciprocal rank score of the melody salience peak amongst the rest of salience peaks (the closer to one the better). The second (S1) is the relative salience of the melody peak in comparison to the highest salience peak in that frame. Last metric (S3) computes the salience of the melody peak, divided by the mean salience of the 3 highest peaks (the higher the better). We consider the latter as the single most important salience-related measure, since it quantifies the ability of a method to make the melody pitch more salient than the rest of the peaks, which is a key property of a salience function.

2.7.2 Melody extraction

Melody extraction algorithms are commonly evaluated by comparing their output against a ground truth, corresponding to the sequence of pitches that the main instrument plays. Such pitch sequence is usually created by employing a monophonic pitch estimator on the solo recording of the instrument playing the melody (Bittner et al., 2014). Pitch estimation errors are then usually corrected by the annotators.

The evaluation in MIREX[19] focuses on both voicing detection and pitch estimation itself. An algorithm may report an estimated melody pitch even for a frame which is considered unvoiced. This allows the evaluation of voicing and pitch estimation separately. Voicing detection is evaluated using metrics from detection theory, such as voicing recall (*VR*) and voicing false alarm (*VFA*) rates. We define a voicing indicator vector v, whose τ^{th} element (υ_τ) has a value of 1 when the frame contains a melody pitch (voiced), and 0 when it does not (unvoiced). We define the ground truth of such vector as v^*. We also define $\bar{\upsilon}_\tau = 1 - \upsilon_\tau$ as an unvoicing indicator.

- **Voicing recall rate** is the proportion of frames labelled as melody frames in the ground truth that are estimated as melody frames by the algorithm.

$$VR = \frac{\sum_\tau \upsilon_\tau \upsilon_\tau^*}{\sum_\tau \upsilon_\tau^*} \tag{2.7}$$

- **Voicing false alarm rate** is the proportion of frames labelled as non-melody in the ground truth that are mistakenly estimated as melody frames by the algorithm.

$$VFA = \frac{\sum_\tau \upsilon_\tau \bar{\upsilon}_\tau^*}{\sum_\tau \bar{\upsilon}_\tau^*} \tag{2.8}$$

Pitch estimation is evaluated by comparing the estimated and the ground truth pitch vectors, whose τ^{th} elements are f_τ and f_τ^* respectively. Most commonly used accuracy metrics are raw pitch (*RPA*) and raw chroma accuracy (*RCA*). Another metric used in the literature is the concordance measure, or weighted raw pitch (WRPA) which linearly weights the score of a correctly detected pitch by its distance in cents to the ground truth pitch. Finally, the overall accuracy (*OA*) is used as a single measure to measure the performance of the whole system:

- **Raw Pitch accuracy (*RPA*)** is the proportion of melody frames in the ground truth for which the estimation is considered correct (within half a semitone of the ground truth).

$$RPA = \frac{\sum_\tau \upsilon_\tau^* \mathcal{T}\left[\mathcal{M}(f_\tau) - \mathcal{M}(f_\tau^*)\right]}{\sum_\tau \upsilon_\tau^*} \tag{2.9}$$

[19] http://www.music-ir.org/mirex/wiki/2014:Audio_Melody_Extraction

$\mathcal{T}$ and $\mathcal{M}$ are defined as:

$$\mathcal{T}[a] = \begin{cases} 1, & \text{if } |a| < 0.5 \\ 0, & \text{else} \end{cases} \tag{2.10}$$

$$\mathcal{M}(f) = 12\log_2(f) \tag{2.11}$$

where f is a frequency value in Hertz.

- **Raw Chroma accuracy (*RCA*)** is a measure of pitch accuracy, in which both estimated and ground truth pitches are mapped into one octave, thus ignoring the commonly found octave errors.

$$RCA = \frac{\sum_\tau \upsilon_\tau^* \mathcal{T}[\| \mathcal{M}(f_\tau) - \mathcal{M}(f_\tau^*) \|_{12}]}{\sum_\tau \upsilon_\tau^*} = \frac{N_{ch}}{\sum_\tau \upsilon_\tau^*} \tag{2.12}$$

where $\| a \|_{12} = a - 12\lfloor \frac{a}{12} + 0.5 \rfloor$, and N_{ch} represents the number of chroma matches.

- **Overall Accuracy (*OA*)** measures the proportion of frames that were correctly labelled in terms of both pitch and voicing

$$OA = \frac{1}{N_{fr}} \sum_\tau \upsilon_\tau^* \mathcal{T}[\mathcal{M}(f_\tau) - \mathcal{M}(f_\tau^*)] + \overline{\upsilon_\tau}^* \overline{\upsilon}_\tau \tag{2.13}$$

where N_{fr} is the total number of frames.

2.7.3 Multiple pitch estimation

The evaluation of multiple pitch algorithms is performed at three different levels, depending on the task.

Multipitch Estimation: the task is to collectively estimate pitch values of all concurrent sources at each individual time frame, without determining their sources. In MIREX (Bay et al., 2009), systems should report the number of active pitches every 10ms. Two commonly used metrics are Precision (the portion of correctly retrieved pitches in all pitches retrieved for each frame) and Recall (the ratio of correct pitches to all ground truth pitches for each frame).

$$Prec = \frac{\sum_{t=1}^{T} TP(t)}{\sum_{t=1}^{T} TP(t) + \sum_{t=1}^{T} FP(t)} \tag{2.14}$$

$$Rec = \frac{\sum_{t=1}^{T} TP(t)}{\sum_{t=1}^{T} TP(t) + \sum_{t=1}^{T} FN(t)} \tag{2.15}$$

where TP correspond to True Positives, FP correspond to False Positives and FN correspond to False Negatives. An estimated pitch is evaluated as correct if it is within a half semitone of a ground-truth pitch for that frame. Note that only one ground-truth pitch can be associated with each returned pitch.

Accuracy (Acc) is a measure of overall performance, bounded between 0 and 1 where 1 corresponds to perfect transcription.

$$Acc = \frac{TP}{TP+FP+FN} \tag{2.16}$$

In order to have more information about the kind of errors, other metrics have been proposed, such as the total error score (E_{tot}), which is computed as the sum of frame level errors, normalised by the total number of f_0 values in the ground truth. If we define N_{ref} as the number of non-zero elements in the ground truth data, N_{sys} as the number of active elements returned by the system and N_{corr} as the number of correctly identified elements:

$$E_{tot} = \frac{\sum_{t=1}^{T} max(N_{ref}(t), N_{sys}(t)) - N_{corr}(t)}{\sum_{t=1}^{T} N_{ref}(t)} \tag{2.17}$$

The total error score can be divided into three different kind of errors: substitution errors, missed errors and false alarms. Substitution errors count the number of ground-truth f_0 values for each frame that were not estimated, but other incorrect f_0 values were returned instead.

$$E_{subs} = \frac{\sum_{t=1}^{T} min(N_{ref}(t), N_{sys}(t)) - N_{corr}(t)}{\sum_{t=1}^{T} N_{ref}(t)} \tag{2.18}$$

Missed errors E_{miss} counts the number of ground-truth f_0 values that were missed by the algorithm, but no other f_0 estimates were returned.

$$E_{miss} = \frac{\sum_{t=1}^{T} max(0, N_{ref}(t)) - N_{sys}(t)}{\sum_{t=1}^{T} N_{ref}(t)} \tag{2.19}$$

The false alarms E_{fa} counts the number of extra f_0 estimates that are not substitutes.

$$E_{fa} = \frac{\sum_{t=1}^{T} max(0, N_{sys}(t)) - N_{ref}(t)}{\sum_{t=1}^{T} N_{ref}(t)} \tag{2.20}$$

Note Tracking: the task is to estimate continuous pitch segments, which would typically correspond to individual notes. In this case, the measures used in MIREX are also Precision (ratio of correctly transcribed ground truth notes to the number of ground truth notes) and Recall (ratio of correctly transcribed ground truth

notes to the number of transcribed notes). A ground truth note is evaluated as correct if the system returns a note that is within a half semitone of that note and the returned note's onset is within a 100ms range (±50 ms) of the onset of the ground truth note, and its offset is within 20% range of the ground truth note's offset.

Timbre Tracking: the task is to estimate pitches and stream them into a single pitch trajectory over the musical excerpt, for each of the sources. This task has not been commonly evaluated in MIREX, due to very low participation. Duan et al. (2014) performed an evaluation, considering that a pitch is estimated as correct when it is within a half semitone of a ground-truth pitch for that frame, and it is assigned to the right stream.

2.7.4 MIREX audio melody extraction

Melody extraction algorithms have been yearly evaluated in MIREX Audio Melody Extraction[20] task, which deals with "the identification of the melody pitch contour from polyphonic musical audio". Pitch is here expressed "as the fundamental frequency of the main melodic voice, and is reported in a frame-based manner on an evenly-spaced timegrid". We now introduce the datasets employed in the evaluation of this task.

2.7.4.1 Datasets

ADC2004 was collected by Emilia Gomez, Beesuan Ong and Sebastian Streich of the Music Technology Group at Universitat Pompeu Fabra, Barcelona. It consists of 20 excerpts of around 20 s, which include real audio recordings (12), excerpts with a voice synthesiser singing the melody (4) and excerpts synthesised from MIDI (4). The collection is currently public, and the total play time is 369 s.

MIREX05 was collected by Graham Poliner and Daniel P. W. Ellis of the Laboratory for the Recognition and Organization of Speech and Audio (LabROSA) at Columbia University, and contains 25 excerpts of 10-40 s duration in the following genres: rock, R&B, pop, jazz and solo classical piano.

INDIAN08, which was compiled by Vishweshwara Rao and Preeti Rao of the Indian Institute of Technology Bombay. It consists of 8 audio clips of north Indian classical vocal performances, with one-minute length each. They include singing voice (male or female, singing the melody), tanpura (Indian drone instrument as background), harmonium (a secondary melodic instrument) and tablas (pitched percussion). The 8 excerpts were created from 4 original recordings, by mixing them twice, each time with differing amounts of accompaniment. The total play time of the collection is 501 s.

[20] http://www.music-ir.org/mirex/wiki/Audio_Melody_Extraction

MIREX09 was compiled by Chao-Ling Hsu and Jyh-Shing Roger Jang of the Multimedia Information Retrieval laboratory at the National Tsing Hua University, Taiwan. It consists of 374 excerpts of Chinese pop karaoke recordings, and features amateur singing over synthesised karaoke accompaniment. 3 different sub-collections were created by mixing the melody and accompaniment with different rations: -5 dB MIREX09 (-5dB), 0 dB and 5 dB. The total play time of each sub-collection is 10,022 s.

2.7.4.2 Results analysis

One of the best performing methods so far in MIREX in terms of overall accuracy is Salamon & Gómez (2012) (evaluated in 2011), which is based on the creation and characterisation of pitch contours. This approach obtains the highest overall accuracy in the INDIAN (0.84) and MIREX09 0dB (0.78) datasets, which contain vocal melodies. As previously introduced, this method uses a fairly simple salience function based on harmonic summation and then creates and characterises pitch contours for melody tracking and voicing detection. Recent approaches have slightly increased *RPA* in those datasets with respect to Salamon's approach, such as Wang et al. (2016) (based on Deep Neural Networks) who increased *RPA* in 0.02 in MIREX09 0dB. However, voicing detection (determining if a frame contains a melody pitch or not) is not improved, and therefore overall accuracy remains lower. Voicing detection is one of the strong aspects of Salamon's method, even though it might be improved further by incorporating timbre information. In contrast, alternative approaches employ more sophisticated salience functions, but the pitch tracking and voicing detection components are less elaborated (Durrieu et al., 2010; Fuentes et al., 2012). Voicing detection has in fact been identified as a crucial task for improving melody extraction systems (Durrieu et al., 2010; Salamon & Gómez, 2012).

The performance of most of the algorithms submitted to MIREX decreases for instrumental pieces. The method proposed by Dressler (2012b) performs better than the rest of methods on datasets containing instrumental data, such as ADC2004 (*OA*=0.86) and MIREX05 (*OA*=0.75), but as we have seen, other methods obtain better results on other collections where the melody is vocal. A main challenge for melody extraction methods is thus to cope with more complex and varied music material, with melodies played by different instruments, or with harmonised melodic lines (Salamon et al., 2014).

2.7.4.3 Limitations

Salamon & Urbano (2012) identified several challenges in the evaluation of melody extraction algorithms in MIREX. The first was related to the length of the clips, which are too short to predict performance on full songs, when considering overall accuracy. Short excerpts tend to be mainly voiced, and thus algorithms with poor voicing estimation accuracy will not be penalised as they would in the case of complete songs,

which commonly have more unvoiced sections. A second issue was the annotation protocol, which should be clarified in order to avoid time offsets, which can have a large effect on the results. Finally, they identified that results on smaller collections (ADC04, MIREX05 and INDIAN08) were not reliable enough. MIREX09 on the other hand is larger than necessary, and more importantly it is not representative of real-world data, since it only contains karaoke recordings, with amateur singing and synthetic accompaniment.

A further limitation of the MIREX campaign is that many collections are kept secret, and since it is run yearly, it is complicated to analyse the reasons why algorithms are failing. The creation of public and open collections is thus a very important task for the advancement of the state of the art in melody extraction (Salamon et al., 2014).

The dataset with symphonic music recordings proposed in Section 3.2 was added to the MIREX 2015, evaluation campaign, increasing the musical diversity and including new definitions of melody in this evaluation campaign.

2.7.5 Publicly available collections

2.7.5.1 Melody extraction

Apart from the development sets of some of the MIREX collection, the RWC dataset has been publicly available since 2006, and more recently MedleyDB has been presented, with the purpose of overcoming many of the limitations of previous collections. RWC[21] was built by the RWC Music Database Sub-Working Group of the Real World Computing Partnership (RWCP) of Japan, which contains manual annotations of the melody for popular and royalty-free subsets (Goto, 2006). The iKala dataset (Chan et al., 2015)[22] is useful for singing voice separation, Query by Humming, melody extraction, and more. It comprises 252 30-second excerpts sampled from 206 songs, and 100 hidden excerpts are reserved for MIREX singing voice separation task.

MedleyDB (Bittner et al., 2014) is currently the largest and more varied melody extraction collection, and it is publicly available[23]. It contains 108 melody annotated files, which are mostly full length songs between 3 and 5 minutes long, and cover a variety of instrumentation and genres. The audio is professional or near professional quality, and the annotations are accurate and well-documented. It provides three different melody annotations, **MEL1**: the f_0 curve of the predominant melodic line drawn from a single source (MIREX definition), **MEL2**: the f_0 curve of the predominant melodic line drawn from multiple sources, and **MEL3**: the f_0 curve of all melodic lines drawn from multiple sources. Note that MEL1 is the melody definition employed in MIREX and used in nearly all research conducted until 2014. Note that

[21] https://staff.aist.go.jp/m.goto/RWC-MDB/
[22] http://mac.citi.sinica.edu.tw/ikala/
[23] http://medleydb.weebly.com/

this collection is also useful for other tasks such as instrument recognition, source separation or automatic mixing since the multitrack recordings are available.

2.7.5.2 Multipitch estimation

There are several publicly available collections for the evaluation of multipitch estimation algorithms.

- RWC classical subset[24] contains 50 recordings of solo performances, chamber and orchestral music. The RWC jazz subset[25] with 50 recordings of different styles and instrumentations. In both subsets, a non-aligned MIDI is provided, but some automatically aligned versions are available[26,27].
- The MAPS database[28] contains 30 classical pieces, each of them played by 9 different piano models (virtual pianos + disklavier).
- LabROSA Automatic Piano Transcription dataset[29] contains 29 pieces with Disklavier piano.
- Bach10 dataset[30] contains 10 multitrack recordings of violin, clarinet, sax, bassoon quartet, with semi-automatically aligned MIDI ground truth.
- TRIOS dataset[31] contains 5 multitrack recordings of classical/jazz trios.
- MIREX multif_0 development dataset[32] contains one woodwind quintet multitrack recording and manual MIDI annotation.
- Score-informed piano transcription dataset[33], contains 7 Disklavier recordings (that present performance mistakes) with MIDI ground truth for recordings and "correct" performances.

[24] https://staff.aist.go.jp/m.goto/RWC-MDB/rwc-mdb-c.html
[25] https://staff.aist.go.jp/m.goto/RWC-MDB/rwc-mdb-j.html
[26] https://staff.aist.go.jp/m.goto/RWC-MDB/AIST-Annotation/SyncRWC/
[27] http://c4dm.eecs.qmul.ac.uk/rdr/handle/123456789/37
[28] http://www.tsi.telecom-paristech.fr/aao/
[29] http://labrosa.ee.columbia.edu/projects/piano/
[30] http://www.ece.rochester.edu/~zduan/resource/Resources.html
[31] http://c4dm.eecs.qmul.ac.uk/rdr/handle/123456789/27
[32] http://www.music-ir.org/evaluation/MIREX/data/2007/multif_0/index.htm
[33] http://c4dm.eecs.qmul.ac.uk/rdr/handle/123456789/13

Chapter 3

Melody in Symphonic Music Recordings

3.1 Introduction

Melody extraction algorithms have commonly been evaluated on relatively simple data, mainly focused on vocal melodies. Therefore, little is known about their performance on more complex instrumental data. The goal of this chapter is to study if melody extraction algorithms are able to generalise well to non-vocal melodies and more complex musical contexts. To do so, we analyse the performance of eleven state-of-the-art pitch estimation algorithms on a complex scenario such as symphonic music. We consider melody extraction (ME) and multipitch estimation (MP) algorithms, as well as an intermediate representation: pitch salience functions (SF), on the task of melody pitch estimation. Methods are selected by considering their relevance, availability (ideally as open source software, or by having access to their estimations on our dataset), and their performance in MIREX (audio melody extraction and multiple pitch estimation tasks). Table 3.1 presents an overview of the evaluated methods, which have been previously introduced in Chapter 2.

In order to carry out the evaluation, the first task is the creation of an annotated melody extraction dataset in this musical context, which we call "Orchset". As described in Section 3.2, after selecting an initial set of excerpts, we ask people to sing along with the music. After a manual agreement analysis, we keep the excerpts in which the participants agree on the notes, and we create the melody annotation by transcribing the sung notes.

Section 3.3 presents an automatic analysis of agreement between humans and algorithms when estimating the melody. In this analysis, we consider the melody extraction methods from Table 3.1. We also study the correlation of both pitch estimation accuracy and mutual agreement, with musical descriptors from the annotated melodies (see section 3.2.1).

In order to gain further knowledge about different pitch estimation methods in this musical context, in Section 3.4 we extend the analysis to the eleven methods from Table 3.1, which also include pitch salience functions and multiple pitch estimation algorithms. Since melody extraction algorithms are commonly based on a pitch salience representation, it is useful to evaluate such a representation separately, in order to better understand the impact of the different stages of the algorithms on melody pitch estimation accuracy. We also analyse how the combination of pitch salience functions can improve melody pitch estimation in orchestral music. In Section 3.5 we present and discuss evaluation results for the traditional melody extraction metrics, as well as some additional ones which provide further information about the considered methods.

Finally, in Section 3.6 we further study timbre-informed melody pitch estimation in symphonic music using PLCA. We compare the results obtained with different spectral templates and study the effect of pre- and post-processing. We also investigate methods for spectral template expansion, which allow us to adapt spectral templates learnt from training data to the characteristics of the signal under analysis.

	Type	Acronym	(Pre Proc.)+Transform	Salience/Multif_0 Estim.	Tracking	Voicing/Polyph.
Cancela et al. (2010)	SF*	CAN	CQT	FChT	-	-
Durrieu et al. (2011)	SF*	DUR	STFT	NMF on S/F model	-	-
Marxer (2013)	SF*	MAR	(ELF)+STFT	TR	-	-
Salamon & Gómez (2012)	SF*	SAL	(ELF)+STFT+IF	Harmonic summ.	-	-
Benetos & Dixon (2011)	MP	BEN	CQT	SIPLCA	[HMM]	[HMM]
Dressler (2012a,b)	MP&ME	DRE	MRFFT	Spectral peaks comparison	Streaming rules	Dynamic thd.
Duan et al. (2010)	MP	DUA	STFT	ML in frequency	[Neighbourhood refin.]	[Likelihood thd].
Durrieu et al. (2010)	ME	DUR	STFT	NMF on S/F model	HMM	Energy thd.
Fuentes et al. (2012)	ME	FUE	CQT	PLCA on the CQT	HMM	Energy thd.
Salamon & Gómez (2012)	ME	SAL	(ELF)+STFT+IF	Harmonic summ.	Contour-based	Salience-based

Table 3.1: Overview of evaluated approaches. The star (*) symbol denotes that pitch salience values are extracted for each of the estimated pitches. Square brackets denote that either tracking or polyphony estimation is not used in the evaluation. In the case of MP-DUA, two versions are considered, with and without refinement. IF: Instantaneous Frequency estimation, (SI)PLCA: (Shift-Invariant) Probabilistic Latent Component Analysis, ML: Maximum Likelihood.

3.2 Symphonic music dataset

The first step for evaluating melody extraction algorithms in the context of symphonic music is the creation of a dataset. This reveals to be a challenge, partially due to the lack of a established annotation methodology when there is more than one instrument playing the melody. Inspired by the definitions of melody in (Poliner et al., 2007; Selfridge-Field, 1998), we collect excerpts in which human listeners agree in their 'essence', that is, the sequence of notes that they hum or sing to represent it. The annotator agreement problem has been discussed in tasks such as chord recognition (Ni et al., 2013) or music similarity (Flexer, 2014). Several MIR datasets have also involved more than one annotator during their creation, e.g. for structure analysis (Smith et al., 2011), instrument recognition (Bosch et al., 2012a) or melody extraction (Bittner et al., 2014).

In this study, the dataset creation comprises several tasks: excerpts selection, recording sessions, analysis of the recordings and melody annotation. We first describe the procedure followed to collect music audio excerpts and describe the final music collection in terms of duration, instruments playing the melody and melodic features (Section 3.2.1). We then provide further details on the designed methodology for human annotation gathering (Section 3.2.2) and analysis of these annotations (Section 3.2.3).

3.2.1 Dataset description and statistics

The proposed dataset is focused on symphonies and symphonic poems, ballets suites and other musical forms interpreted by symphonic orchestras, mostly from the romantic period, as well as classical and 20th century pieces. Music recordings are taken from stereo commercial recordings, and selected to have an adequate recording audio quality. They are sampled to create short excerpts with a potential dominant melody, maximising the existence of voiced segments (containing a melody pitch) per excerpt.

To verify that the excerpts contain a clear melody and identify the exact sequence of notes, we collected human annotations by recording subjects singing the melody, as described in Section 3.2.2. From the starting set of excerpts, we select those in which subjects agree on the sequence of notes (melody), and annotate them as detailed in Section 3.2.3. An overview of the whole process is shown in Figure 3.1.

The final collection, which is freely available for research purposes[34], contains 64 audio excerpts with their corresponding annotation of the melody in MIDI format. This dataset has been used in the Audio Melody Extraction task in MIREX 2015 and 2016.

[34] http://www.mtg.upf.edu/download/datasets/orchset

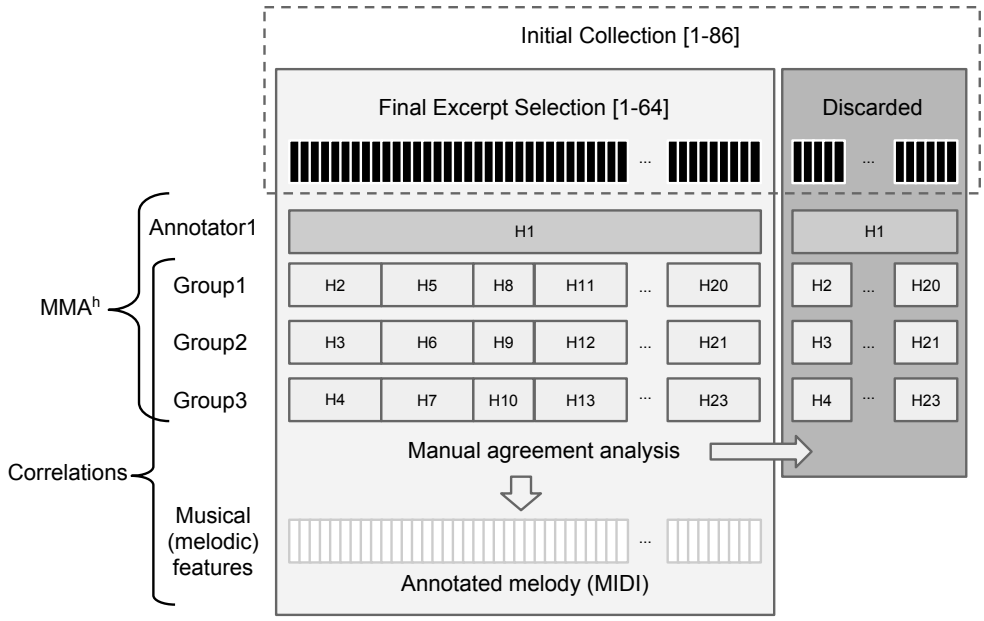

Figure 3.1: Dataset creation process. H1, H2, etc. refer to the recordings of each of the annotators, which correspond to several excerpts. Group1, Group2 and Group3 refer to different sets of subjects, and Annotator1 refers to the author of this thesis, who annotated all excerpts. "MMA^h" shows the subjects used to compute the Mean Mutual Agreement. "Correlations" shows the subjects considered to compute correlations with musical features.

The length of the excerpts ranges from 10 to 32 seconds (μ = 22.1 s., σ = 6.1 s.). For each excerpt we provide an annotation with the sequence of melody pitches using a sampling period of 10 ms. If no melody pitch is annotated at a specific time, the frame is considered as unvoiced, otherwise it is consider as voiced. 93.69% of the frames of the dataset are labelled as voiced while 6.31% are unvoiced (in which case the pitch is set to be 0). The number of excerpts per composer are: Beethoven (13), Brahms (4), Dvorak (4), Grieg (3), Haydn (3), Holst (4), Mussorgsky (9), Prokofiev (2), Ravel (3), Rimsky-Korsakov (10), Schubert (1), Smetana (2), Strauss (3), Tchaikovsky (2), Wagner (1).

In order to understand the characteristics of the annotated melodies, we compute a set of statistics about instrumentation, pitch and rhythm related features. Regarding instrumentation, only in one excerpt there is a single instrument (oboe) playing the melody (with orchestral accompaniment). In the rest of the dataset, the melody is played by several instruments from an instrument section, or a combination of sections, or even alternating sections within the same excerpt. Figure 3.2 (left) illustrates the statistics of the predominant instrumental sections playing the melody. Figure 3.2 (right) depicts the distribution of pitches of all frames of the dataset, and a Gaussian model (μ = 74.1, σ = 12.1). Using the MIDI Toolbox (Eerola & Toiviainen, 2004), we compute a set of melodic descriptors for each of the ground truth MIDI files (con-

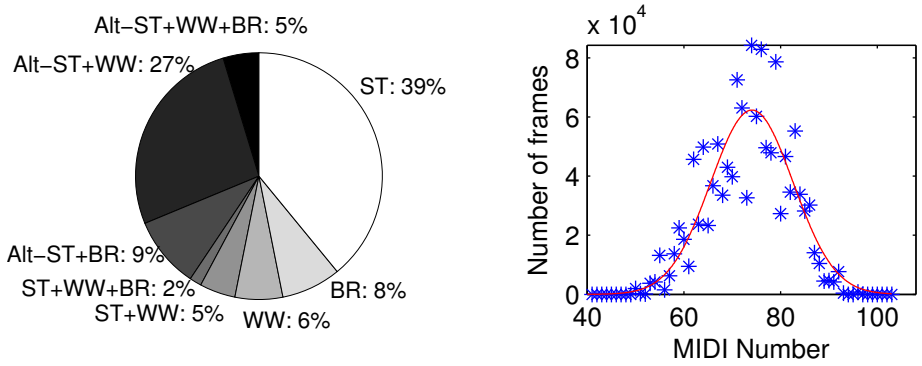

Figure 3.2: Distribution of the sections of the instruments playing the main melody (left) (ST: Strings, BR: Brass, WW: Woodwinds), where Alt- denotes that the sections alternate within the excerpt. Distribution and Gaussian model of the annotated 'melody' pitches (right).

taining the sequence of melody notes):

- Density: amount of notes per second.
- Range: difference in semitones between highest and lowest note pitch.
- Tessitura: melodic tessitura based on pitch deviation from median pitch height (Von Hippel, 2000).
- Complexity (pitch, rhythm, mixed): expectancy-based model of melodic complexity (Eerola & North, 2000) based either on pitch or rhythm-related components, or on a combination of them together.
- Melodiousness: 'suavitatis gradus' proposed by Euler, which is related to the degree of softness of a melody, and is a function of the prime factors of musical intervals (Leman, 1995).
- Originality: Different measurement of melodic complexity, based on tone-transition probabilities (Simonton, 1984).

Additionally, we compute the melodic intervals found in the dataset, as the difference in semitones between consecutive notes. Histograms with the distribution of the melodic features are depicted in Figure 3.3. We observe that although melodies in the dataset have varied characteristics in terms of the computed descriptors, there are some general properties. Melodic intervals generally lie in a relatively small range, according to the voice leading principle of pitch proximity (Huron, 2001). The most common sequence of two notes is a perfect unison, followed by a major second, and then minor second either descending or ascending. Previous works obtained similar

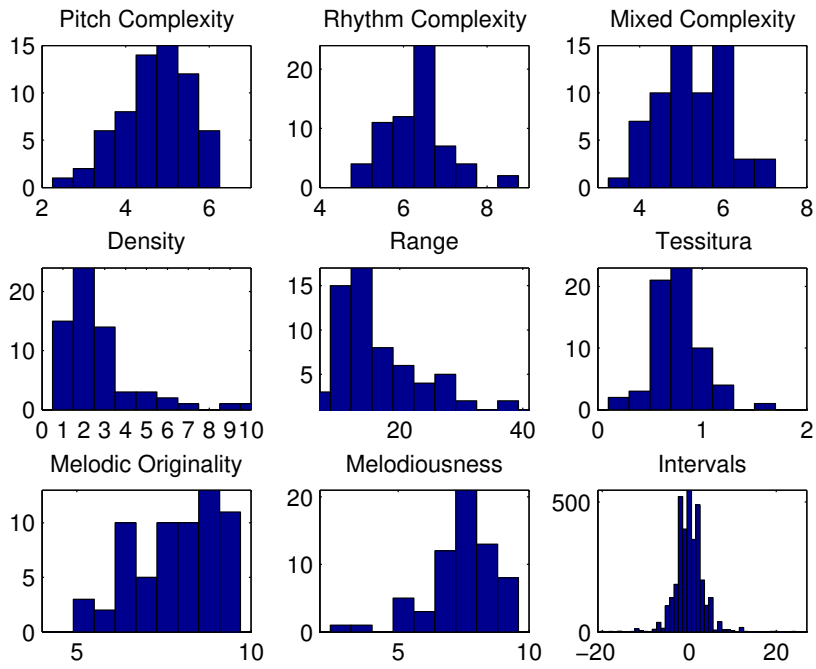

Figure 3.3: Melodic feature distribution.

conclusions, with a dataset of 6000 MIDI files from varied genres (Dressler, 2012b), or in a dataset of polyphonic ring tones (Friberg & Ahlbäck, 2009). The melodic density histogram shows that most excerpts present an average of less than three notes per second, which also corresponds to the results obtained by Dressler (2012b). Some differences with respect to the cited works are: the fact that our dataset presents a larger range of intervals, and that some excerpts present a higher amount of notes per second (and thus a lower inter-onset interval). Similar melodic features have been previously used in combination with classifiers to select the tracks containing the melody in a MIDI file (Rizo et al., 2006). In Chapter 4 we also use some of these characteristics for melody pitch tracking. In Section 3.3.6 and Section 3.5.2.3, we analyse the correlation between the presented melodic characteristics and algorithm accuracy.

3.2.2 Recording sessions

We carried out recording sessions where subjects had to carefully listen to the audio samples twice and then sing or hum along with the audio three more times. As excerpts were repeated and relatively short, subjects could more easily memorize them. A total of 32 subjects with a varied musical background and a common interest in

music took part in the recording sessions. The instructions provided to the subjects were to 'hum or sing the main melody (understood as the sequence of notes that best represent the excerpt)'. They were also instructed to focus on pitch information rather than on timing (onsets and offsets).

During the session, subjects rated how well they knew each of the excerpts before the experiment (ranking from 1 to 4). After the recordings, they also filled out a survey asking for their age, gender, musical background, amount of dedication to music playing, and a confidence rating of their own singing during the experiment, in terms of the percentage of melody notes that they considered they sang correctly ('Less than 30%', '30-60%', '60-90%', 'More than 90%'). We discarded 9 subjects which could not properly accomplish the task, based on both their confidence (those which responded 'Less than 30%') and their performance in some excerpts, which contained an easy to follow single melodic line. The selected 23 subjects sang a subset of the collection, and were distributed to have three different subjects singing each excerpt. Additionally, the author of this thesis sang the whole collection, so finally there were four different subjects per excerpt, as shown in Figure 3.1.

Personal and musical background statistics of the selected annotators are: age (min=23, max=65, median=31.5), gender ('male' (66.7%), 'female' (33.3%)); musical background ('None' (16.7%), 'Non-formal training' (16.7%), 'Formal training less than 5 years' (0%) and 'Formal training more than 5 years' (66.7%)); dedication to music playing ('None' (16.7%), 'Less than 2 hours per week' (16.7%), 'More than 2 hours per week' (45.8%), 'Professional musician' (20.8%)).

3.2.3 Manual analysis and melody annotation

Our next step is to analyse the sung melodies and select the excerpts in which the four subjects sang the same sequence of notes. Given the difficulty of singing some of the excerpts (fast tempo, pitch range, etc.), the notes sung by the participants are contrasted with the musical content of the piece, mapping them to the notes played in the excerpt. The objective is to transcribe the notes that the participants intended to sing, allowing small deviations in the sung melodies. Such deviations typically arise from an incorrect singing of some notes, notes which are not present in the piece but the participants sang, or from the presence of a chord in the excerpt, in which some subject sang a different note compared to the rest. In the final selection, we keep only the excerpts in which the four participants agreed in nearly all notes. In this process, we also consider the reported self-confidence on their singing, giving less importance to notes which disagree with the rest if they were sung by people with less self-confidence.

After selecting the excerpts, we create the melody annotations by manually transcribing the notes sung by the participants, adjusting onsets and offsets to the audio. Since vocal pitch range is different to the range of the instruments playing the main melody, notes are transposed to match the audio. For excerpts in which melody notes are

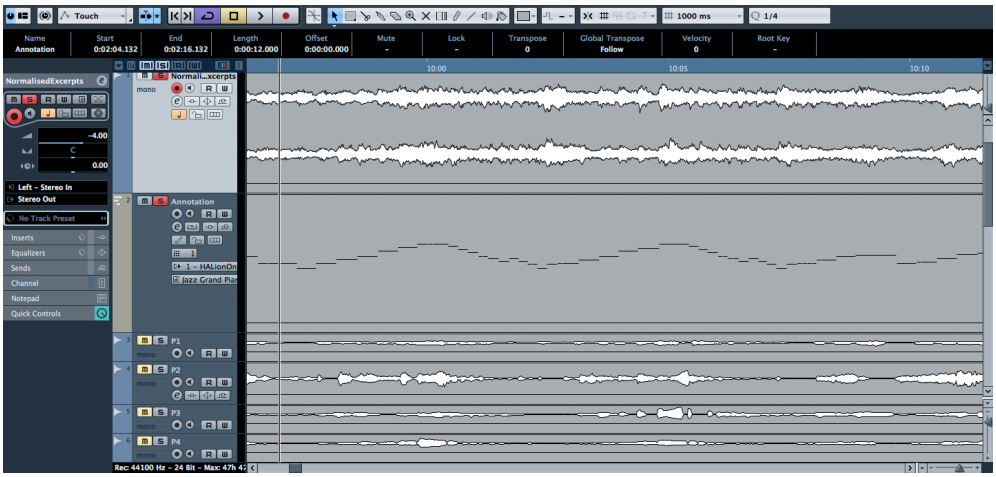

Figure 3.4: Recordings and MIDI annotation of the melody in a Digital Audio Workstation.

simultaneously played by several instruments in different octaves, we resolve the ambiguity by maximising the melodic contour smoothness (minimising jumps between notes). The recording sessions and the manual transcription of the melody notes are performed within a Digital Audio Workstation (Cubase 5), as shown in Figure 3.4. Figure 3.5 (top) shows the pitches sung by the four subjects, as well as the annotation of the melody for one of the excerpts. We observe that all subjects follow a similar melodic contour despite some slight differences, in some cases in different octaves (related to the gender of the annotator).

3.3 Mutual agreement

We further analyse the collected recordings and the output of state-of-the-art automatic melody extraction methods, in order to study the agreement between both humans and algorithms, inspired by a related work on beat estimation (Zapata et al., 2014). To do so, we first process the voice recordings in order to obtain the sequence of pitches corresponding to the singing. We then compute mutual agreement (*MA*) between pitch sequences that humans and algorithms considered as the melody of a given excerpt. Given the different pitch ranges under comparison (coming from human voices and symphonic orchestra instruments), we select chroma accuracy as the melody extraction evaluation metric used for agreement computation, since it ignores octave information.

One of the challenges for this comparison is that some subjects did not focus on timing, so their recordings are not properly aligned to the note onsets in the music. Since evaluation metrics are based on a frame-to-frame comparison, we apply a Dynamic Time Warping (DTW) technique to align both pitch sequences before the metric computation. We then compute Mean Mutual Agreement (*MMA*) based on chroma ac-

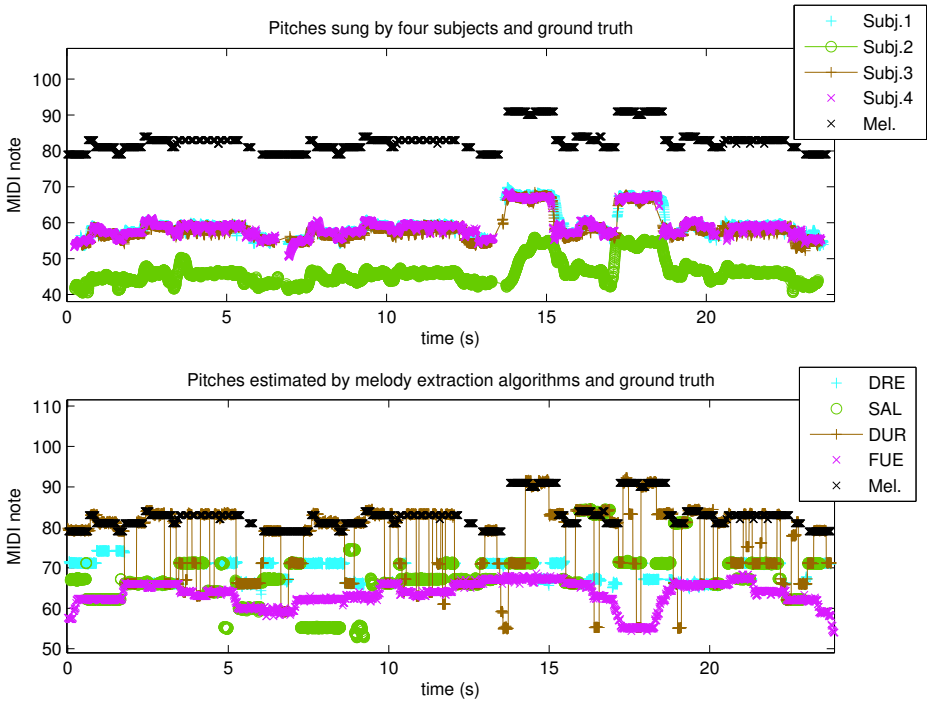

Figure 3.5: Pitches sung by four subjects and melody annotation, for an excerpt of the 4th movement of Dvořák's 9th Symphony (top). Pitches estimated by four melody extraction methods and melody annotation for the same excerpt (bottom).

curacy, and extract a set of melodic features to characterise each excerpt. Finally we study correlations between mutual agreement and different characteristics of music excerpts and subjects.

3.3.1 Human melody extraction

The recordings obtained in Section 3.2.2 are converted into pitch sequences using probabilistic yin (pYin)[35], a monophonic pitch estimator with temporal smoothing, which has been shown to have higher accuracy than other commonly used algorithms (Mauch & Dixon, 2014). The step size used corresponds to 5.8 ms. Some of the recordings are not properly converted, due to the fact that the subject was whistling instead of singing. In those specific recordings, a more accurate pitch estimation was obtained with MELODIA (Salamon & Gómez, 2012)[36] in the monophonic setting, with a range between 110 and 1760 Hz. As previously mentioned, pitch sequences of the sung melodies are commonly not properly aligned with the ground truth, or with

[35] https://code.soundsoftware.ac.uk/projects/pyin
[36] http://mtg.upf.edu/technologies/melodia

other extracted melody sequences, since people commonly sung with a delay in comparison to the note onsets. In order to minimise this effect, we align both sequences using a Dynamic Time Warping (DTW) technique[37] (Sakoe & Chiba, 1978; Müller, 2007) based on chroma information, which has been extensively used for a number of tasks such as cover version identification, audio-to audio or audio-score alignment, or to compute melodic similarity.

3.3.2 Automatic melody extraction

As previously introduced, for the analysis of agreement we employ the complete melody extraction algorithms from Table 3.1, previously introduced in Chapter 2. We adapt the frequency range to the dataset under evaluation (from 103 Hz to 2.33 KHz) for all algorithms except Salamon & Gómez (2012) and Dressler (2012b), which cannot be configured to these values.

3.3.3 Mean mutual agreement

In a previous study on beat tracking (Zapata et al., 2014), beat extraction evaluation metrics were used to compute the agreement $A_{i,j}$ between algorithms i and j which aim at automatically identifying the sequence of beats in an audio excerpt. In the present study, we adapt the concept of agreement to the task of melody extraction, and use raw chroma accuracy, since it is the most relevant metric for this particular context. The agreement between two sequences i, j, which correspond to the estimated melody of an excerpt e ($A_{i,j}[e]$) is here equal to the raw chroma accuracy when using sequence j as ground truth and i as the estimation, for an excerpt e. With such definition, it is important to note that we obtain different results depending on the sequence used as ground truth, so $A_{j,i}$ is different to $A_{i,j}$. We define Mutual Agreement as:

$$MA_i[e] = \frac{1}{N-1} \sum_{j=1, j\neq i}^{N} A_{i,j}[e], \quad MA_i = \frac{1}{N_{exc}} \sum_{e=1}^{N_{exc}} MA_i[e] \tag{3.1}$$

where N is the total number of estimators (algorithms or subjects), i is the index of the estimator, e the excerpt number, and N_{exc} the total number of excerpts in the collection.

We define the Mean Mutual Agreement for an excerpt e ($MMA[e]$) as the average $MA[e]$ for all N estimators, and *MMA* as the average Mean Mutual Agreement for all excerpts.

$$MMA[e] = \frac{1}{N} \sum_{i=1}^{N} MA_i[e], \quad MMA = \frac{1}{N_{exc}} \sum_{e=1}^{N_{exc}} MMA[e] \tag{3.2}$$

We use MMA^a to denote *MMA* for algorithms, and MMA^h for humans.

[37] http://www.ee.columbia.edu/ln/LabROSA/matlab/dtw/

3.3.4 Data gathering and methodology

Apart from the mutual agreement, we also compute the agreement of both humans and algorithms with the annotations. We employ Raw Chroma Accuracy as metric, in order to discard octave information. We also use the superscripts 'a', and 'h' to denote if we refer to algorithms or human melody extraction respectively, e.g. RCA^a refers to the raw chroma accuracy obtained by algorithms.

The collection creation schema, recordings and information used for the computation of the MMA^h and correlation is shown in Figure 3.1. As mentioned in Section 3.2, we collected singing data from 4 different subjects for each excerpt, one of them being always the author of this thesis (Annotator1). Since the amount of excerpts sung by Annotator1 is thus much higher than for the rest of subjects, we do not include his data in the computation of the RCA^h, nor in the statistical analyses, so as not to bias them. In the case of Mean Mutual Agreement, we use his recordings since we are only analysing musical factors.

In following subsections we analyse mutual agreement and its correlation with the melodic descriptors mentioned in Section 3.2.1 and subject related factors mentioned in Section 3.2.2. First, we analyse the agreement between humans and melody annotations. Second, we analyse agreement between algorithms and melody annotations. Third, we study mutual agreement between humans and finally between algorithms.

Raw chroma accuracy and mutual agreement results are provided as the average results for all excerpts in the database.

3.3.5 Agreement between humans and melody annotations

We first compare sung melodies against annotated melodies (manually created ground truth, in symbolic format). For each of the three takes we recorded, we compute the average RCA^h (μ_{RCA^h}) for all excerpts and the three subjects. In order to understand the influence of mistunings, we increase the tolerance (tol) in the evaluation measure from 0.5 to 1.5 semitones in steps of 0.25 semitones, as shown in Table 3.2.

	0.5	**0.75**	**1**	**1.25**	**1.5**
Take1	37.9	47.5	53.7	58.5	61.9
Take2	40.3	50.5	56.7	61.4	64.7
Take3	43.4	53.5	59.6	64.3	67.4

Table 3.2: μ_{RCA^h} in different takes, and with different tolerances in semitones.

Table 3.2 shows that accuracy values are relatively small in general, due to differences in timing and tuning between the compared pitch sequences. In fact, the chosen tolerance has a very clear impact in the results, indicating the presence of relatively small mistunings between both pitch sequences. In addition, we observe that subjects moderately increase their accuracy with each new take, with a similar increase when

evaluating with different interval tolerances. This suggests that this is not only due to a correction of mistunings but to the refinement of the selection of notes belonging to the melody. The results reported in the rest of the section have been computed using the third take, with tol = 1 semitone, in order to allow some flexibility with human errors in tuning. In the case of algorithms, we keep the standard value of tol = 0.5 semitones, since they are less affected than humans by tuning problems.

As mentioned, we have identified timing deviations in subjects' singing in relation to notes onsets and offsets, which ideally should not be considered when measuring the overall agreement. We therefore apply a DTW algorithm allowing only temporal deviations between -0.5 to 0.25 seconds, since subjects were typically slightly delayed. After the alignment, RCA^h increases from 59.6 to 67.4% for the considered take (3) and tolerance (1 semitone).

Table 3.3 shows the Pearson correlation between the chroma accuracy and the melodic features of the considered music excerpts, for both original and aligned pitch sequences. According to Table 3.3, there is a strong correlation of both RCA^h and $RCA^{h_{al}}$ with several musical parameters such as melodic range, density and melodic complexity (in pitch, rhythm and mixed). Correlation is negative with all of the mentioned factors, and the strongest one is pitch complexity. There is no strong correlation with melodiousness, originality and tessitura.

	RCA^h	$RCA^{h_{al}}$	RCA^a
excerpt knowledge	0.16	0.11	NA
age	-0.17	-0.16	NA
range	-0.37	-0.37	-0.13
density	-0.43	-0.35	-0.44
tessitura	0.06	0.06	0.06
pitch complexity	-0.45	-0.38	-0.33
rhythm complexity	-0.26	-0.21	-0.11
mixed complexity	-0.37	-0.32	-0.22
melodiousness	-0.08	-0.03	-0.05
originality	-0.01	-0.01	-0.12

Table 3.3: Correlation of melodic features with raw chroma accuracy obtained by humans in original (RCA^h) and aligned singing ($RCA^{h_{al}}$), and algorithms (RCA^a).

We then perform a variance decomposition analysis to study the individual contribution of each factor to the observed variance in the responses. We start with the saturated random-effects linear model containing all main factors, and iteratively simplify it by removing factors whose effect is not statistically significant (α=0.05). Once the model is simplified, we run an ANOVA analysis and compute the individual contributions to total variance. Table 3.4 shows the percentage of variance in raw chroma accuracy.

We observe that most variance in RCA^h is due to note density, melodic range, and musical background. An analysis of the results shows that people without musical

	% Var	% Var. (aligned)
density	26.69	16.03
range	15.66	19.35
excerpt knowledge	0.12	0.00
time playing	1.19	1.65
self confidence	1.18	0.90
musical background	24.99	26.54
sex	4.18	4.16
residual	25.98	31.34

Table 3.4: % of variance in human raw chroma accuracy, for both original (RCA^h) and aligned pitch sequences ($RCA^{h_{al}}$).

training obtain lower median accuracy (48.8%) than those with formal background (63.6%) or with non formal training (68.5%). There is also some residual variance that could be either related to subjects' or excerpts' characteristics which we are not considering, or from interactions between the considered factors. In both Table 3.3 and Table 3.4 we observe that the effect of note density decreases in the case of aligned performances, since we allow temporal deviations. Another important result is that the methodology followed in the recording sessions (subjects listened to four repetitions of the excerpt before the considered take) is enough to ensure that the degree of knowledge of the excerpt before the recording session would not affect the extracted melody: Table 3.3 shows that there is only a very small correlation of RCA^h with user knowledge of the excerpt for non aligned pitch sequences, and Table 3.4 shows that there is no variance due to the excerpt knowledge.

3.3.6 Agreement between algorithms and melody annotations

Results obtained by comparing the output of melody extraction algorithms against the annotated melodies are shown in Table 3.5.

	RPA	*RCA*	*OA*
ME-DRE	49.4	66.5	46.0
ME-DUR	66.9	80.6	62.6
ME-FUE	27.8	60.2	24.1
ME-SAL	28.5	57.0	23.5

Table 3.5: Values for Raw Pitch (RCA^a), Raw Chroma (RCA^a) and Overall Accuracy (OA^a) obtained by algorithms.

We observe that the highest accuracies are obtained by the algorithm proposed by Durrieu, especially in the case of *RPA* and *OA*. For *RCA*, Durrieu achieves 80.6% accuracy, 14.1% above the method by Dressler, which is the following one in terms of *RCA*.

Third column in Table 3.3 shows the correlation of RCA^a with musical properties

of the excerpts. We observe that the highest (negative) correlation found belongs to melodic density, followed by pitch complexity. Originality and range have a very small negative correlation. In the case of range, we observe that there is a difference between algorithms and humans, since humans are more (negatively) influenced by the melodic range. A variance components analysis (see Table 3.6) shows that most variance comes from algorithm ID, followed by melodic density, pitch complexity, and tessitura, and a residual variance of 23.7% due to other factors.

	% Var.
mixed complexity	1.03
rhythm complexity	9.10
pitch complexity	16.59
tessitura	12.45
range	0.64
algorithm ID	21.22
residual	23.70

Table 3.6: Percent of variance in RCA^a due to different factors.

3.3.7 Mutual agreement between humans

We computed the Mean Mutual Agreement (*MMA*) between all subjects (MMA^h), with three different tolerances (tol) for raw chroma accuracy computation. With a small tolerance (tol=0.5 semitones), Mean Mutual Agreement in humans only achieves 37.71%. We would expect the agreement to be higher than this value, since *MMA* was computed for the final excerpt selection, in which the manually analysed agreement between subjects was very high. However, by increasing the tolerance to 1 semitone, MMA^h increases up to 57.47%, and if we allow a deviation in chroma of 1.5 semitones, we achieve a mutual agreement of 68.16%, as we do not penalise possible mistunings.

The identified temporal deviations in subjects' singing also affect *MMA*. We now perform an alignment using DTW, as previously presented, but allowing temporal deviations between -1 and 1 second, since we need to align sequences of pitches produced by 2 subjects and need thus to consider higher differences in timing, e.g. when one subject is singing too early, and the second one is delayed. After the alignment, we increase from $MMA^h = 57.47\%$ to a $MMA^{h_{al}} = 76.07$ % for a tolerance of 1 semitone.

Table 3.7 shows the correlation of MMA^h and $MMA^{h_{al}}$ with musical properties of the annotated melody. These are the factors more strongly (negatively) correlated with *MMA*: density, range, pitch complexity and mixed complexity.

We now investigate if the manual selection of excerpts explained in Section 3.2.3 could have been automatically performed by selecting those excerpts with high $MMA^{h_{al}}$. The mean (μ) and standard deviation (σ) of $MMA^{h_{al}}$ for discarded excerpts is: μ = 62.53, σ = 9.51. In the case of the selected excerpts: μ = 76.07, σ = 10.14. While

	MMA^h	$MMA^{h_{al}}$	MMA^a
range	-0.49	-0.46	-0.14
density	-0.58	-0.48	-0.38
tessitura	0.10	0.09	0.12
pitch complexity	-0.62	-0.55	-0.33
rhythm complexity	-0.34	-0.25	-0.05
mixed complexity	-0.53	-0.44	-0.21
melodiousness	-0.035	-0.08	-0.06
originality	0.01	0.05	-0.22

Table 3.7: Correlation between musical factors and *MMA*.

the mean *MMA* in the selected excerpts is higher, some manually discarded excerpts have higher *MMA* than others which had been selected. As previously introduced in Section 3.2.2, this is due to the fact that in the manual selection process we consider not just pitch but also rhythm information from the singing which is not considered in the automatic analysis of agreement, as well as the musical content of the piece. An example of this fact is shown in Figure 3.6, where we observe differences in the pitch sequences sung by the subjects (ignoring octave information). The agreement between humans in this excerpt is quite low ($MMA^h = 34.41\%$, $MMA^{h_{al}} = 56.73\%$), and the agreement between algorithms is higher ($MMA^a = 63.44\%$). However, a manual analysis of the recordings (contrasting subjects singing to the musical content of the piece) reveals that participants agreed in most of the notes (corresponding to the annotated melody). Disagreement is due to the large melodic range, and difficulties in singing some notes.

3.3.8 Mutual agreement between algorithms

We compute the agreement between algorithms by comparing pairs of automatically extracted pitch sequences. In this case, there is no need to perform any temporal alignment. The highest agreement according to Table 3.8 is 69.7% obtained between ME-DRE (ground truth) and ME-DUR (estimator). The lowest agreement is 57.1%, between ME-FUE (ground truth) and ME-SAL (estimator). Note that the highest agreement is obtained with the two methods that obtain the highest melody extraction accuracies (see table 3.5). The Mean Mutual Agreement between all algorithms is $MMA^a = 61.71\%$.

	ME-DRE	**ME-SAL**	**ME-DUR**	**ME-FUE**
ME-DRE	100.0	66.1	69.7	62.4
ME-SAL	64.6	100.0	57.5	59.0
ME-DUR	68.1	57.4	100.0	60.0
ME-FUE	59.2	57.1	58.2	100.0

Table 3.8: Agreement between algorithms $A^a_{i,j}$, where the names of the rows represent the ground truth sequence, and the column names represent the estimation.

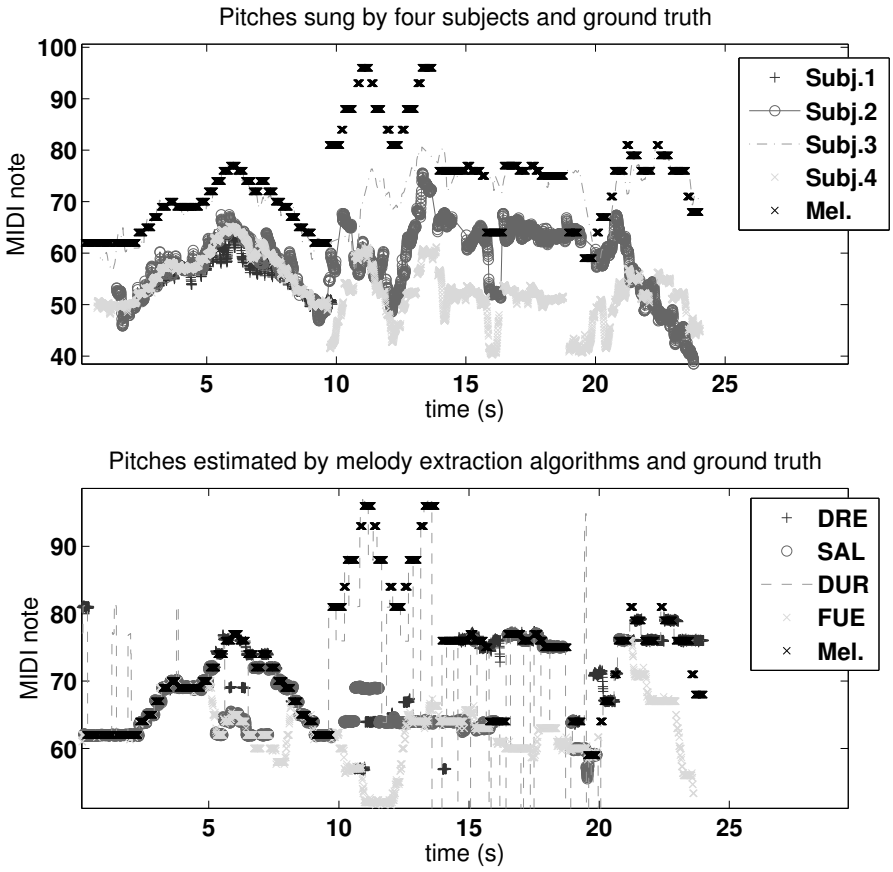

Figure 3.6: Sequences of pitches sung by the four subjects (top), four algorithms (bottom) and the ground truth annotation for the melody (Mel.).

The correlation analysis of MMA^a is presented in the third column of Table 3.7. Note density, pitch complexity, originality and mixed complexity are the factors more correlated to *MMA*. Range and rhythm complexity have much lower (negative) correlation than in the case of humans, meaning that algorithms are more robust to them. Originality has a medium negative correlation with MMA^a but practically no correlation with MMA^h, which suggests that the higher the melodic originality the least algorithms will agree.

3.3.9 Mutual agreement between humans and algorithms

We finally compute the correlation between the mutual agreements MMA^a and MMA^h, which is medium (0.3) as shown in Table 3.9. We also computed the correlations

between *MMA* and *RCA*, which are strong when they both refer to humans or algorithms, but correlation between MMA^a and RCA^h or between MMA^h and RCA^a is weaker.

	MMA^h	$MMA^{h_{al}}$	MMA^a
MMA^h	1	0.93	0.3
$MMA^{h_{al}}$	0.93	1	0.3
MMA^a	0.3	0.3	1
$\boldsymbol{RCA^h}$	0.58	0.53	0.21
$\boldsymbol{RCA^{h_{al}}}$	0.48	0.46	0.15
$\boldsymbol{RCA^a}$	0.24	0.18	0.63

Table 3.9: Correlation between Mean Mutual Agreements and with raw chroma accuracies.

3.3.10 Summary

After analysing several kinds of agreements between both humans and algorithms in the task of melody extraction in symphonic classical music, we observed that some melodic features are correlated to accuracy results and agreements. The analysis of agreement shows that melodic range and note density have a clear negative correlation with accuracy results obtained by people. In the case of algorithms, the highest (negative) correlation is with note density, and results suggests that algorithms are less affected by melodic range than humans, as long as pitches are kept within their limits of operation. With regard to subject-related factors, we found out that previous knowledge of an excerpt had almost no correlation with the accuracy obtained by humans, and no contribution to total variance, which validates the proposed design for data gathering in our dataset. With regard to automatic melody extraction, the mean raw chroma accuracy of the four algorithms is 66.1%, but with important differences between them. Durrieu's approach obtains the highest scores, reaching 80.6% raw chroma accuracy. Section 3.5 presents a more complete evaluation of these and other pitch estimation methods.

In the case of Mean Mutual Agreement, we observed a negative correlation with melodic density and complexity (specially pitch complexity), in both humans and algorithms. Comparing the agreement between humans and algorithms, we observed that excerpts with a higher melodic originality make algorithms differ more in their estimations than in the case of humans. Finally, we identified a strong positive correlation between raw chroma accuracy and Mean Mutual Agreement, for both humans and algorithms. However, there is a lack of a strong correlation between the raw chroma accuracy obtained by humans and the Mean Mutual Agreement obtained by algorithms, and vice versa.

3.4 Evaluation setup

We now present the methodology for the evaluation of state-of-the-art pitch estimation approaches on the proposed symphonic music dataset. In Section 3.3.6 we already had a first impression of the difficulty of this task on such data. As seen in Table 3.5, four melody extraction methods obtained very different accuracies, and were generally much lower than the results obtained in MIREX datasets.

In order to gain more insights about the performance of different approaches when estimating the melody pitch on such data, we consider a total of eleven pitch estimation algorithms for evaluation, including pitch salience functions, multipitch estimation methods and melody extraction algorithms (see Table 3.1). For the selection of the methods, we consider their relevance in the state of the art, availability (ideally as open source software, or by having access to their estimations on our dataset), and their performance in MIREX (audio melody extraction and multiple pitch estimation tasks). Additionally we propose a simple method which combines the salience functions of several methods, as described in Section 3.4.3. Finally we introduce an additional set of evaluation metrics, which provide further knowledge about the characteristics of the evaluated methods.

3.4.1 Methodology

Three types (SF: salience function, MP: Multiple Pitch estimation, ME: Melody extraction) of pitch estimation algorithms are evaluated on the proposed dataset. We are interested on the evaluation of both complete melody extraction algorithms, as well as intermediate representational levels in order to better understand the origin of differences between methods' results. Specifically, we evaluate the ability of salience functions and multipitch methods to output the ground truth pitch of the melody within the N most salient estimates. The motivation behind this evaluation strategy is twofold: first to understand which methods obtain better accuracy when estimating the melody pitch, and second to analyse the number of estimates that each of the methods needs to output, in order to have the ground truth pitch among the pitch estimates. This would be useful for tasks such as pitch tracking, since we would like to reduce the number of f_0's to be tracked.

In the case of pitch salience functions and multipitch algorithms, only the estimated pitch which is closest to the ground truth (in cents) is used in each frame for the calculation of raw pitch related measures (equation 2.9). For chroma related measures, we create the sequence $\widehat{p^{ch}}$ by keeping in each frame the pitch (in cents) which is both correct in chroma (chroma match) and closer in cents to the ground truth, or we set a 0 otherwise. For instance, if the ground truth is 440 Hz, and the output pitches are 111 Hz, 498 Hz and 882 Hz (N = 3) we would keep the last one.

Pitch salience functions are also evaluated by extracting the N = 10 highest peaks with a minimum difference of a quarter tone between them, and ordering them by salience.

For multipitch algorithms, we select a maximum of 10 estimates (commonly they output less than 10 pitches). In the case of MP-DRE, pitches are not ordered by salience, so we just consider N = 10.

Considering the characteristics of the dataset, the subjective nature of some part of the annotations (octave selection), and the objectives of the benchmark, we propose an evaluation based on the combination of well-established evaluation metrics presented in Section 2.7.2 and additional metrics presented in Section 3.4.4, which provide more information about the algorithms' performance and characteristics.

3.4.2 Approaches

We label each of the evaluated approaches according to its type (SF, MP, ME), and the three first letters of the first author's surname to refer to a specific method (e.g. SF-DUR refers to the salience function by Durrieu et al. (2011)). We evaluate the methods using the original implementation by the authors. We adapt the algorithm parameters (pitch estimation range) to fit our dataset, according to Figure 3.2 (right) (from 103 Hz to 2.33KHz), in all algorithms except SF-SAL, ME-SAL, ME-DRE and MP-DRE, which are not configurable to these values. An overview of the evaluated methods is provided in Table 3.1.

In SF-CAN, we also adjusted the Gaussian function to the statistics of this dataset as in (Cancela et al., 2010): tripling the standard deviation ($\sigma = 36.3$) and with the same mean ($\mu = 74.1$) compared to the fitted Gaussian model from Figure 3.2 (right).

As a reminder, we evaluate two salience-based approaches ((Salamon & Gómez, 2012)[38], and (Dressler, 2012b)) and two separation-based approaches ((Fuentes et al., 2012)[39], and (Durrieu et al., 2010)[40]). Salamon & Gómez (2012) (ME-SAL) use a pitch salience function based on harmonic summation and then create contours to do melody tracking using heuristic rules. Dressler (ME-DRE) uses almost the same system as in (Dressler, 2012a), except for the frequency range in the selection of pitch candidates, which is narrower in the case of melody extraction. Fuentes et al. (2012) (ME-FUE) use PLCA on a CQT to build a pitch salience function, and Viterbi smoothing to estimate the melody trajectory. Durrieu et al. (2010) (ME-DUR) use a pitch salience function based on a source-filter model, and a Viterbi algorithm for tracking. Voicing detection (deciding if a particular time frame contains a pitch belonging to the melody or not) is approached by the evaluated algorithms using a dynamic threshold (Dressler, 2012b), an energy threshold (Durrieu et al., 2010; Fuentes et al., 2012), or a salience distribution strategy (Salamon & Gómez, 2012).

Figure 3.6 (bottom) shows the pitches estimated by the four melody extraction algorithms, as well as the annotation of the melody. As it can be observed, this is a

[38] http://mtg.upf.edu/technologies/melodia
[39] http://www.benoit-fuentes.fr/articles/Fuentes2012_ICASSP/index.html
[40] https://github.com/wslihgt/separateLeadStereo

challenging excerpt since there are many estimation errors (including octave errors) as well as jumps between octaves.

We evaluate two variants of Duan et al. (2010), one with refinement (MP-DUA-Ref), and one without it (MP-DUA). In both cases, we do not use polyphony estimation, so that both algorithms output all estimated pitches. The approach by Benetos & Dixon (2011) is included in the evaluation, but we do not consider tracking, and no threshold for polyphony estimation, so as to only consider the intermediate non-binary pitch representation (MP-BEN). MP-DRE is a more recent implementation of the method by Dressler evaluated in MIREX (Dressler, 2012a), with the main difference that it outputs more pitches, which are not ordered by salience. Table 3.1 summarises the evaluated approaches.

3.4.3 Combination method

We additionally consider a simple hybrid method that combines the output of several pitch salience functions and then performs peak detection and neighbourhood-based refinement. The main assumption is that if several algorithms agree on the estimation of a 'melody' pitch, it is more likely that the estimation is correct. Related works also use agreement between algorithms for beat estimation (Holzapfel et al., 2012; Zapata et al., 2014). In Chapter 4 we present further pitch salience combination methods, focused on both improving melody pitch estimation and reducing the salience on unvoiced frames.

The combined salience function (COMB) is created frame-by-frame, placing a Gaussian with σ semitones standard deviation in the output pitches of each of the algorithms, weighted by the estimated salience of the pitch, and then summing all Gaussians. The selected value of σ was 0.2, so that the maximum value of the sum of two Gaussians separated more than a quarter tone is not higher than the maximum value of both Gaussians.

An alternative option is to combine the raw salience functions, however this method is preferred for this evaluation since it can be equally applied to methods estimating multiple discrete pitches. Additionally, the use of Gaussian functions allows to cope with small differences between the estimated and the melody pitch. Since each algorithm has a different pitch salience range, we normalise the values before combining them, so that the sum of the salience of all frequency bins in a given frame is equal to 1, following probabilistic principles. Finally, we multiply the salience values of each of the methods (M_s) by a different value ($\omega_{M_s} \in [0,1]$), allowing a weighted combination. A value of $\omega_{M_s} = 0$ is thus equivalent to not including a method in the combination. An example of the combination of salience functions is given in Figure3.7, where three salience functions with the same weight ($\omega_{MAR}, \omega_{DUR}, \omega_{CAN} = 1$) agree on the estimation of pitches around MIDI notes 75 and 87, while only one of them estimates pitches around MIDI notes 74 and 77. This gives a maximum salience in the sum (combination) to the pitch around 75, which corresponds to the annotated

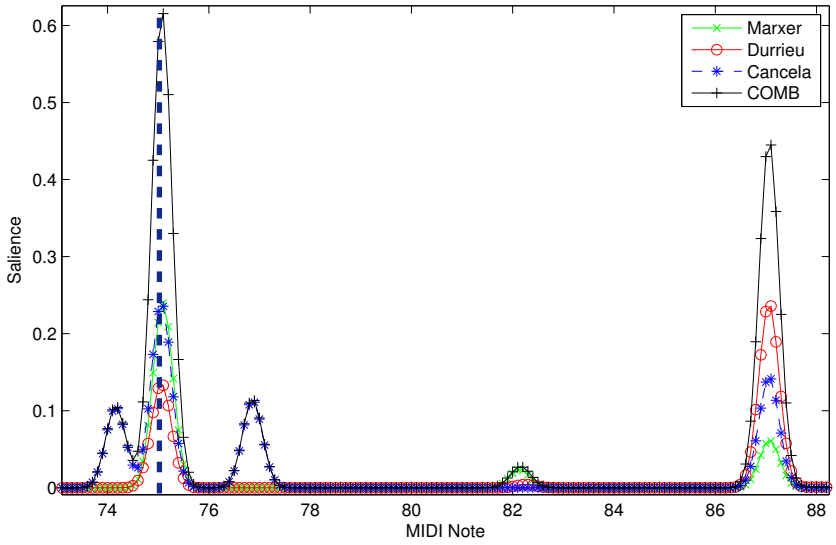

Figure 3.7: Gaussians centred at the pitches estimated by three salience functions (SF-MAR, SF-DUR and SF-CAN) at a given frame, and the sum of them (COMB). The maximum peak of the combination is found at the annotation of the melody pitch (vertical dashed line).

melody pitch. After the addition, we extract the N highest peaks with a minimum difference of a quarter tone between them.

A further refinement step is then performed to remove the f_0 estimates inconsistent with their neighbours, with a method similar to the one employed in MP-DUA-Ref (Duan et al., 2010). Our contribution is to weight each of the estimated pitches with its salience when computing the histogram, as opposed to the original method, which gives the same weight to all estimated pitches in a frame, regardless of their (estimated) salience. We denote this method as RCOMB. In the evaluation, the maximum number of extracted peaks is set to N = 10, as in the evaluation of the rest of salience functions and multipitch algorithms.

We test several combinations of SF-DUR, SF-CAN, SF-SAL and SF-MAR with different weights, in order to find the best performing configuration. We conduct a 5 fold cross validation with only 20% of the dataset for training, and 80% for testing. The combinations are named: COMB when no refinement is used, and RCOMB for the refined version, followed by the ω value and the identifier of each of the salience functions (e.g. COMB-0.5SAL-1DUR). We also use the name: RNSCOMB for the combination refined with the original method by Duan et al. (2010) (which is the same as RCOMB but does not use estimated salience information).

3.4.4 Proposed metrics

In order to better understand the algorithms' performance, and due to the subjective nature of some part of the annotations (octave selection), we propose an additional set of metrics. The motivation behind them is that the metrics used in MIREX do not inform about the continuity of the correctly estimated pitches (either in pitch or chroma), which is very relevant for tasks such as automatic transcription, source separation or the visualisation of melodic information.

We consider continuity in both pitch and time with three different metrics:

- **Weighted Raw Chroma accuracy (*WRCA*)** measures the distance in octaves (OD_i) between the correct chroma estimates and the ground truth pitches. The parameter $\beta \in [0,1]$ is introduced to control the penalisation weight due to the difference in octaves. If β is low the value of *WRCA* tends to *RCA*, and if β is high *WRCA* tends to *RPA*.

$$OD_i = round\left[(\widehat{p_i^{ch}} - p_i)/1200\right] \tag{3.3}$$

$$Ech_i = min(1, \beta \cdot |OD_i|) \tag{3.4}$$

$$WRCA = \frac{\sum_i (1 - Ech_i)}{N_{vx}} \cdot 100 \tag{3.5}$$

  where i is the index of a voiced frame with a chroma match, p_i is the value in frame i of the ground truth pitch, $\widehat{p_i^{ch}}$ is the value in frame i of the sequence $\widehat{p^{ch}}$, N_{vx} is the number of voiced frames.

- **Octave Jumps (*OJ*)** is the ratio between the number of voiced frames in which there is a jump between consecutive correct estimates in chroma, and the number of chroma matches (N_{ch}).

$$J_i = (OD_i - OD_{i-1}) \tag{3.6}$$

$$OJ = count(|J_i| > 0)/N_{ch} \cdot 100 \tag{3.7}$$

- **Chroma Continuity (*CC*)** quantifies errors due to octave jumps (EJ), and is influenced by their location with respect to other octave jumps, and by the difference in octaves between the estimated and ground truth pitch (Ech_i). The parameter λ is introduced to control the penalty weight due to the amount of octaves difference in an octave jump (J_i), and ranges from 0 to 1. The lower the value of λ, the more *CC* tends to *WRCA*.

$$EJ_i = \min(1, \lambda \cdot |J_i|) \tag{3.8}$$

$$MEJ_i = \max_{k \in [i-w,i]} (EJ_k) \tag{3.9}$$

$$CC_i = 1 - \min(1, Ech_i + MEJ_i) \tag{3.10}$$

$$CC = \frac{\sum_i (CC_i)}{N_{vx}} \cdot 100 \tag{3.11}$$

where $w = \min(F, i)$, $F = round\,[L/H]$, L is the length in seconds of the region of influence of an octave jump, and H is the hop size in seconds. The lower the value of L the more *CC* tends to *WRCA*.

The chroma continuity metric assigns the highest score to a result that is equivalent to the ground truth in terms of raw pitch. The score is also high if the extracted pitch sequence is transposed by one octave, but decreases if the octave distance is higher. The score also decreases with the amount of jumps between correct chroma estimates. If a set of errors are concentrated in one part of the excerpt, this metric penalises less than if it is distributed in different positions over the excerpt (errors propagate to the neighbouring frames, therefore location of errors also affects the metric).

The values of λ, β and L should be tuned according to the application where the algorithms are used. The pitch range of analysis in our case spans 4.5 octaves, and thus the maximum distance between correct chroma estimates is $OD_i^{max} = 4$ octaves. We decide to linearly divide the error Ech_i, and thus we set a value of $\beta = 1/OD_i^{max} = 0.25$. We equally weight both octave jumps and octave errors $\beta = \lambda = 0.25$, and set $L = 0.2$ s.

3.5 Melody extraction results

3.5.1 Overview

We now provide an overview of algorithm performance, and an analysis of obtained results, including the influence of instrumentation, melodic features and energetic predominance of the melody. We also discuss the results of the proposed evaluation measures. Finally, we present a generalizability study in order to assess the significance of these results in Section 3.5.5.

Table 3.10 summarizes the evaluation results of all considered methods for a single pitch estimate. Results for each evaluation metric are computed as an average of the results for each excerpt in the dataset. Additionally, standard deviations are presented between parentheses. We observe that the best performance is obtained by the melody extraction method ME-DUR for all metrics. Its raw pitch accuracy (*RPA*) is equal to 66.9%. The difficulty of this material for state of the art approaches is evident since ME-SAL obtains up to 91% *RPA* in the MIREX09+5dB dataset, and only 28.4% in our dataset. SF-DUR obtains the highest *RPA* among all evaluated salience functions and multipitch methods (61.8%), which indicates that the good performance of the complete melody extraction method is due to the salience function used. Table 3.10 also presents results obtained with a combination of two methods (SF-MAR and SF-DUR) with equal weight ($\omega = 1$) and two combination strategies: original (COMB)

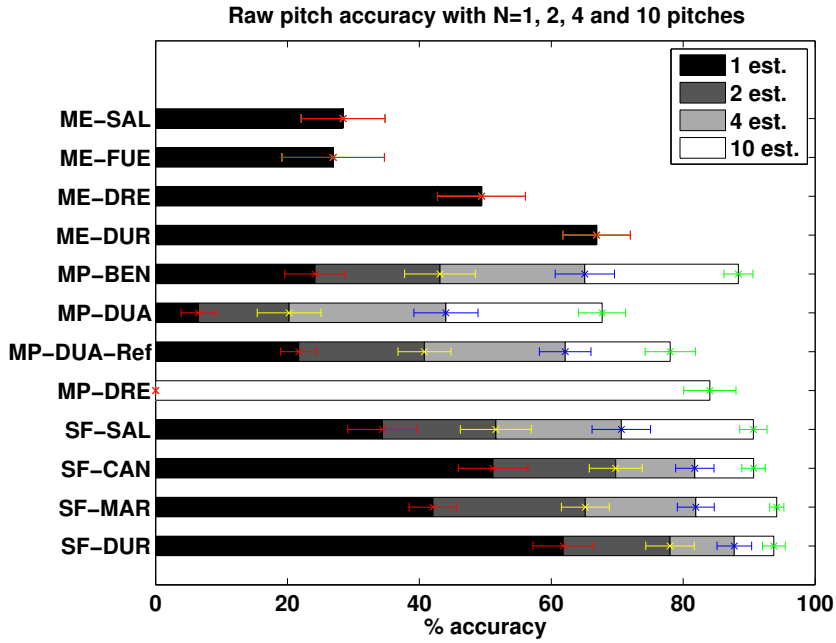

Figure 3.8: Mean raw pitch accuracy for N = 1,2,4 and 10 pitch estimates. Bars represent 95% confidence intervals. For MP-DRE we only provide the measure for N = 10 as the output pitches are not ordered by salience.

and with the proposed salience-based neighbourhood refinement (RCOMB). The refined combination method increases the *RPA* obtained with SF-DUR up to 64.8%. Further analysis about the proposed combination method is provided in Section 3.5.3.

Figure 3.8 shows the mean raw pitch accuracy (*RPA*) for all methods. For salience functions and multipitch estimation methods, *RPA* is computed for N = 1, 2, 4 and 10 estimated pitches. We observe that methods obtaining highest accuracies with many pitch candidates are salience functions, since multipitch methods often perform a candidate filtering step (e.g. MP-DRE or MP-DUA) that may erroneously discard the ground truth melody pitch. As expected, an increase in N provides an increase in accuracy, up to 94.2% for SF-MAR with N = 10, closely followed by SF-DUR. With N = 4, the maximum *RPA* decreases 6.1%, obtained by SF-DUR, followed by SF-MAR and SF-CAN. The lowest accuracy is obtained by SF-SAL, for N = 1, 2 and 4. These results indicate that although these methods do not generally estimate the melody pitch as the most salient in orchestral music data, they usually find it within the 10 most salient ones. In Section 3.5.2 we analyse the influence of salience functions in complete melody extraction algorithms in symphonic music.

In the case of multipitch estimation algorithms, the best accuracy for any value of N is obtained with MP-BEN, but is lower than any of the salience functions. One of the possible reasons is the fact that the instrument basis used were learnt from single

instruments from the RWC dataset, and may not applicable in large orchestral settings. In Section 3.6 we present further results on timbre informed pitch estimation, by substituting single string instrument templates for templates created from orchestral sections (e.g. violin section), and applying filtering to improve the melody pitch estimation in our dataset. MP-DRE obtains slightly lower results than MP-BEN for N = 10. Since this algorithm does not output pitch estimates ordered by salience, it is not possible to know accuracy results for lower values of N. MP-DUA does not perform as accurately even with refinement (MP-DUA-Ref). Possible causes include the use of a binary mask for the peak region in the definition of the likelihood, and the shape of the peaks, which may be significantly different than expected (Duan et al., 2010).

Given the potential of combining different methods, we further study the accuracy of the combination method with different weights. We perform a grid search with $\omega \in \{0, 0.5, 1\}$, for each of the 4 salience functions (SF-MAR, SF-DUR, SF-CAN, SF-CLA). The highest mean raw pitch accuracy over all excerpts is always obtained with $\omega_{DUR} = 1$, and $\omega_{MAR}, \omega_{SAL}, \omega_{CAN} = 0.5$ or 0. We then perform a finer search, with $\omega_{DUR} = 1$, and $\omega_{MAR}, \omega_{SAL}, \omega_{CAN} \in \{0, 0.2, 0.4, 0.6\}$. Figure 3.9 shows the results obtained in the testing set by the best performing combinations in the training set. This figure shows several combinations, with a different number of algorithms (from 2 up to 4). Results obtained with the proposed refinement method (RCOMB-) are also presented for two of the approaches, and results of SF-DUR are additionally included as a reference. The accuracy obtained with the weighted combination increases in comparison to the individual methods, especially with the proposed salience-based neighbourhood refinement, for all values of N.

	RPA	*WRPA*	*RCA*	*WRCA*	*OA*	*OJ*	*CC*
RCOMB-1MAR-1DUR	**64.8** (18.6)	47.2 (15.6)	79.3 (12.8)	75.5 (13.2)	60.6 (18.9)	2.2 (1.9)	**70.6** (14.4)
COMB-1MAR-1DUR	**61.6** (17.4)	44.8 (14.6)	77.5 (11.9)	73.3 (12.2)	57.5 (17.7)	11.3 (8.0)	**62.7** (14.3)
SF-DUR	**61.8** (18.4)	43.2 (14.2)	77.1 (12.5)	73.0 (13.0)	57.8 (18.7)	11.7 (8.3)	62.5 (15.1)
SF-MAR	42.1 (14.5)	30.7 (12.3)	68.9 (14.3)	61.6 (13.3)	39.3 (14.4)	11.1 (4.9)	48.4 (12.2)
SF-CAN	51.2 (21.1)	35.1 (16.9)	74.8 (13.1)	68.4 (13.0)	48.0 (20.7)	12.3 (9.4)	57.0 (16.2)
SF-SAL	34.4 (21.1)	25.3 (16.6)	62.7 (18.5)	54.1 (17.8)	32.3 (20.5)	18.0 (9.3)	41.4 (17.8)
MP-DRE	14.6 (9.9)	11.0 (7.9)	31.2 (15.1)	26.3 (13.0)	13.6 (8.9)	4.6 (3.7)	23.4 (12.3)
MP-DUA-Ref	21.7 (11.0)	14.7 (8.1)	47.6 (15.0)	39.0 (12.7)	21.5 (10.8)	8.1 (3.0)	29.7 (11.0)
MP-DUA	6.5 (10.5)	5.2 (8.3)	34.5 (16.6)	23.3 (14.5)	8.4 (10.8)	43.2 (23.8)	13.7 (13.6)
MP-BEN	24.2 (18.4)	12.3 (10.5)	51.0 (20.1)	40.7 (18.7)	22.8 (18.0)	6.8 (3.6)	32.0 (17.9)
ME-DUR	**66.9** (20.6)	**47.1** (16.0)	**80.6** (12.4)	**76.8** (13.2)	**62.6** (20.8)	**1.7** (2.2)	**73.3** (15.2)
ME-DRE	49.4 (26.7)	37.4 (21.3)	66.5 (20.5)	61.9 (20.7)	46.0 (25.4)	2.2 (2.8)	59.3 (21.6)
ME-FUE	26.9 (31.1)	22.5 (26.7)	59.4 (25.0)	49.7 (24.5)	23.4 (26.5)	5.1 (5.5)	45.0 (26.0)
ME-SAL	28.4 (25.4)	21.4 (19.6)	57.0 (20.7)	48.2 (20.8)	23.5 (19.2)	4.3 (3.8)	43.4 (22.0)

Table 3.10: Evaluation results for a single pitch estimation (N = 1), for metrics presented in Section 2.7. *RPA*: Raw Pitch accuracy, *WRPA*: Weighted Raw Pitch accuracy, *RCA*: Raw Chroma accuracy, *WRCA*: Weighted Raw Chroma accuracy, *OA*: Overall Accuracy, *OJ*: Octave Jumps, *CC*: Chroma Continuity. Mean values (and standard deviation) over all excerpts in the dataset are presented. Bold fond indicates especially relevant results.

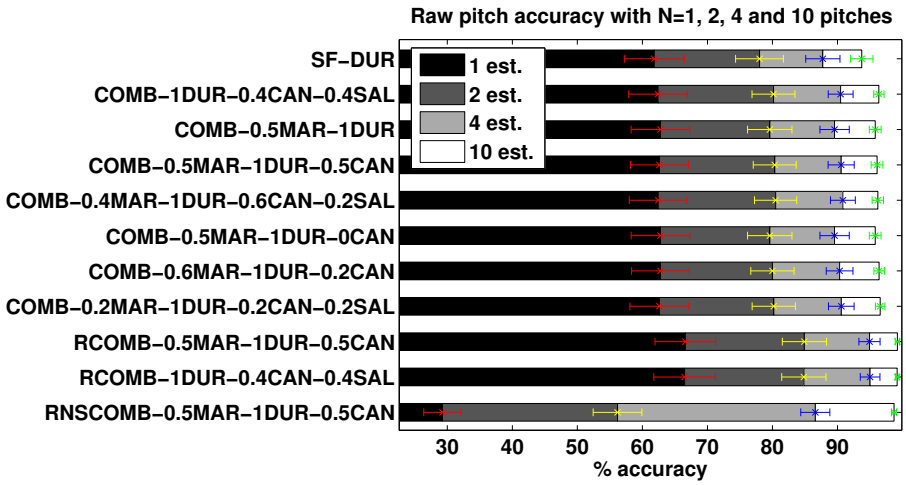

Figure 3.9: Mean raw pitch accuracy (*RPA*) for the combination of four salience functions: SF-DUR, SF-MAR, SF-SAL and SF-CAN for N = 1, 2, 4 and 10 pitch estimates. Bars represent 95% confidence intervals. RCOMB denotes a combination with the proposed neighbourhood refinement method. RNSCOMB corresponds to the refined estimation with the method proposed in (Duan et al., 2010). The values of the weights (ω) are indicated before the name of each method. SF-DUR is shown as a reference.

A manual examination of the estimation errors suggests that the most challenging excerpts contain chords and harmonisations of the melody, a highly energetic accompaniment, and in some cases percussion. Most accurate estimations are generally obtained in excerpts with a very predominant melody (e.g. those in which the orchestra plays mostly in unison). A more detailed analysis of the influence of several musical characteristics is presented in the following section.

3.5.2 Discussion

3.5.2.1 Comparison between melody extraction methods

We now study the performance of melody extraction methods, and analyse the influence of their salience functions. The focus is set on ME-DUR and ME-SAL since their respective salience functions are also available for evaluation.

The best results for a single pitch candidate are obtained with ME-DUR, partially due to the very good performance of its melody oriented salience function (SF-DUR) (Durrieu et al., 2011), which has relaxed constraints in the source filter model compared to (Durrieu et al., 2010). This allows modelling several harmonic sources and makes this approach applicable to a broader range of signals. According to Table 3.10, SF-DUR obtains 61.7% raw pitch accuracy even without any smoothing, and with the full melody extraction method (using Viterbi algorithm for tracking), ME-DUR obtains the highest raw pitch accuracy: 66.9%. In the case of the overall accuracy (*OA*), ME-DUR also benefits from the fact that it estimates nearly all frames (99.8%) as

voiced, which is appropriate for the low percentage of unvoiced frames of this dataset.

The accuracy obtained with SF-SAL is the lowest one compared to the rest of salience functions. In comparison to SF-DUR, it achieves 27.4 percentage points (pp) less *RPA* for N = 1, which partially explains that the complete melody extraction method (ME-SAL) also performs much worse in comparison to ME-DUR (38.5 pp less *RPA*). Note that while ME-DUR improves the results obtained with SF-DUR (5.1 pp), ME-SAL obtains a lower accuracy than SF-SAL (6 pp). This heuristic rule-based approach (which obtains some of the highest overall accuracies in MIREX datasets) seems to be tuned to the pitch contour features of vocal music (pop, jazz), and encounters difficulties to generalise to the characteristics of our dataset. The salience-based voicing detection method in ME-SAL is quite conservative in this dataset, and classifies only 57.4% of the frames as voiced, possibly because it is tuned to vocal excerpts, and there is a higher dynamic range in symphonic music. For this reason, both false alarm rate and voicing recall are the lowest from all methods. Since the proposed dataset contains a high ratio of voiced versus unvoiced frames, the overall accuracy obtained with ME-SAL is more reduced than with other methods compared to the raw pitch accuracy.

ME-DRE achieves higher accuracy than ME-SAL, possibly due to the fact that it does not assume specific features of human voice, and is thus more general. This agrees with the results obtained in datasets used in MIREX which contain non-vocal melodies, such as ADC2004 and MIREX05 (Salamon et al., 2014). However, the results in our dataset are not as good as those obtained with ME-DUR. Since we do not have access to the salience function used by ME-DRE (based on the pairwise analysis of spectral peaks), it is difficult to get further insights on the limitations of this approach. A possible explanation of the better performance is the fact that the salience function is melody-oriented in ME-DUR, while ME-DRE uses a similar salience function as in the multipitch method. The source filter model seems to adapt to the spectrum of the melody source even if it does not correspond to a single lead instrument, or even to a single instrumental section. A further reason is that Durrieu's approach employs a more complex generative model, in which the filter shape is learnt from the data (related to the timbre of the lead instrument), and the repetitions in the accompaniment are exploited in the model to compute melody pitch salience. Finally, ME-FUE presents the lowest accuracy in this dataset. Its probabilistic model and the smoothing method employed seem not to be adequate for this kind of data.

In order to study octave errors produced by melody extraction methods, we observe the difference between raw pitch and raw chroma accuracy in Table 3.10. The smallest difference (and thus lowest amount of octave errors) is found in ME-DUR, and the highest one in ME-SAL and ME-FUE. As already observed by Durrieu et al. (2011), the signal representation employed in SF-DUR produces few octave errors. A possible explanation is that SF-DUR performs a joint estimation of the salience of all possible pitch candidates. Additionally this method jointly estimates the timbre of

	ST	BR	WW	Non-Alt	Alt
SF-DUR	**67.4**	**66.1**	**53.8**	**65.3**	**56.7**
SF-MAR	45.2	46.6	38.0	45.6	36.9
SF-SAL	34.9	53.6	24.0	35.5	32.8
SF-CAN	55.9	65.3	39.9	53.5	47.7
MP-DRE	12.8	21.2	16.4	15.1	13.8
MP-DUA	7.0	9.8	1.9	6.1	7.1
MP-DUA-Ref	24.6	27.1	6.2	22.6	20.3
MP-BEN	25.4	45.0	16.6	26.7	20.5
ME-DRE	49.5	71.2	40.6	51.8	45.9
ME-DUR	**70.7**	**73.0**	**58.8**	**70.4**	**61.8**
ME-FUE	26.5	50.1	14.5	**26.4**	**27.7**
ME-SAL	27.7	44.7	22.7	**28.7**	**28.0**

Table 3.11: Raw pitch accuracy results for all evaluated methods (with N = 1 for SF and MP), in relation to the predominant instruments playing the melody: ST - strings, BR - brass, WW - woodwinds, as well as the division between alternating (Alt) and non-alternating instruments (Non-Alt). Bold fond indicates especially relevant results.

the pitch candidates corresponding to the melody over a long time span, which also helps reducing the amount of octave errors. This suggests that ME-DUR has reduced octave errors since the pitches are correctly estimated from the first step, and there is no need for any further octave correction. Estimating salience of each pitch candidate independently with harmonic summation, and performing an octave error removal step afterwards (as in ME-SAL) leads to a lower accuracy in this symphonic music dataset.

3.5.2.2 Influence of instrumentation

In order to illustrate the influence of instrumentation in algorithm performance, Table 3.11 presents mean *RPA* results for excerpts with a melody predominantly played by either strings, brass or woodwinds sections.

We also compute the mean *RPA* for excerpts with a melody which is alternatingly played by two or more instrument sections, and compare it against results obtained from excerpts with no alternation. Although there is only a small number of excerpts for certain instrument sections, we still identify some trends in algorithm performance. For instance, Table 3.11 shows that ME-SAL or ME-FUE are less influenced by the alternation of the melody between instrument sections, while ME-DUR presents a higher difference in accuracy. This is probably due to the fact that SF-DUR aims to learn the timbre of the lead instrument for each excerpt, and if the timbre of the instrument playing the main melody changes throughout the excerpt, the extraction may be affected. However, even with alternating instruments, SF-DUR learns timbral basis that are generic enough (Durrieu et al., 2010), and creates a salience function that outperforms the rest of algorithms in terms of pitch estimation accuracy. In contrast,

	RPA(r)	***RPA***(τ)	***RPA***(ρ)	***RCA***$(\mathbf{r})$	***RCA***(τ)	***RCA***(ρ)
range	-0.04	-0.05	-0.09	-0.13	-0.12	-0.18
density	**-0.2**	**-0.14**	**-0.19**	**-0.44**	**-0.33**	**-0.48**
tessitura	0.04	-0.01	-0.02	0.06	-0.03	-0.05
pitch complexity	**-0.18**	**-0.13**	**-0.18**	**-0.43**	**-0.32**	**-0.46**
rhythm complexity	-0.09	-0.05	-0.07	-0.24	-0.15	-0.22
mixed complexity	**-0.17**	**-0.11**	**-0.15**	**-0.41**	**-0.29**	**-0.42**
melodiousness	0.04	0.02	0.03	-0.05	-0.03	-0.05
originality	-0.04	-0.05	-0.07	-0.12	-0.09	-0.13

Table 3.12: Correlations between raw pitch and chroma accuracy of the considered melody extraction methods (ME-DUR, ME-DRE, ME-SAL, ME-FUE) with the extracted melodic features, for 3 different correlation types: Pearson (r), Kendall (τ), Spearman (ρ). Bold fonts indicate highest (negative) correlation values.

ME-SAL does not exploit timbre, which explains why there is just a small difference between excerpts with alternating and non-alternating instrumentation playing the melody. This occurs with both the salience function (SF-SAL) and the complete melody extraction algorithm (ME-SAL).

According to Table 3.11, it is generally easier to extract the melody in excerpts in which it is played by the brass section, while in the case of the strings section, accuracies are generally lower. The relative decrease in accuracy reaches up to almost 50% in the case of ME-FUE. An important exception is ME-DUR, for which melodies played by strings are equally well recognised as with brass, probably due to the fact that timbre information is exploited by learning the lead instrument filter basis for each excerpt. This aspect has a large influence on the average results of this dataset, given the high percentage of excerpts which contain a string section playing the melody.

3.5.2.3 Influence of melodic characteristics

In order to further study the influence of melodic characteristics (described in Section 3.2.1) on melody extraction performance, we present a correlation analysis in Table 3.12, now also including *RPA*. Results obtained with three different correlation measures show that note density and pitch complexity are the features that most affect accuracy, while melodic originality and tessitura have almost no effect on it. Correlations are stronger with *RCA* compared to *RPA*, since some algorithms commonly produce octave errors (difference between *RCA* and *RPA* in Table 3.10).

3.5.2.4 Influence of energetic predominance of the melody

Finally, we study how the energetic predominance of the melody pitch affects algorithm performance. We estimate the ratio (from 0 to 1) between the energy of the melodic source(s) and the overall energy. The energy of the melody is estimated on a frame basis by applying an informed source separation method that isolates the

	Correlation
SF-DUR	0.45
SF-MAR	0.51
SF-SAL	0.83
SF-CAN	0.53
MP-DRE	0.53
MP-DUA	0.67
MP-DUA-Ref	0.54
MP-BEN	0.86
ME-DRE	0.71
ME-DUR	**0.36**
ME-FUE	0.76
ME-SAL	0.77

Table 3.13: Correlation between raw pitch accuracy (with N = 1) and the ratio between the energy of the melodic source(s) and the overall energy. The lowest correlation is marked in bold.

melody signal from the background using the ground truth pitches (Durrieu et al., 2010). The ratio is computed for each excerpt as the mean of the ratios in each voiced frame (containing melody), so as not to be influenced by the amount of unvoiced segments. We then compute the correlation between the estimated melodic predominance ratio and the accuracy results, as shown in Table 3.13. In the case of salience functions (with N = 1), we observe that SF-SAL has the highest correlation (0.83). On the other hand, SF-DUR presents the lowest correlation (0.44). Other approaches such as SF-MAR (0.51) or SF-CAN (0.53) obtain intermediate correlations. This shows that the harmonic salience function used by SF-SAL is less capable than SF-DUR of identifying melodic pitches as the most salient ones when they are not energetically predominant over the accompaniment, at least in our symphonic music dataset. Since salience functions strongly affect the performance of complete melody extraction algorithms, ME-DUR presents the smallest correlation among them (0.36), while ME-SAL (0.76), ME-FUE (0.75) and ME-DRE (0.71) present much stronger correlations.

These results suggest that approaches which perform signal modelling related to source separation are especially useful in the context of orchestral classical music. Results show that ME-DUR is better able to extract melodies played by non-predominant instruments, partially due to the melody-oriented pitch salience function, based on a source-filter model.

3.5.3 Combination method

In this section we analyse the performance obtained by combining the results of different algorithms, using the methodology presented in Section 3.4. The highest *RPA* obtained in the training data with N = 1, and no refinement reached 62.7%, with COMB-

0.6MAR-1DUR-0.2CAN. This combination increases the accuracy obtained with SF-DUR alone in more than 1 pp. However, 7 different combinations obtained a *RPA* with a difference of less than 0.1 pp compared to the maximum, all of them with $\omega_{DUR} = 1$, and several combinations of weights for the rest of algorithms. The best combination using only two algorithms was among them: COMB-0.6MAR-1DUR. For N = 2, best *RPA* in the training set was obtained with: COMB-0.4MAR-1DUR-0.6CAN-0.2SAL, reaching 80.4%. The best combination with 3 algorithms is: COMB-0.5MAR-1DUR-0.5CAN, achieving 80.3% and with 2 algorithms, COMB-0.5MAR-1DUR achieved 79.5%. It is worth noting that other combinations, with different algorithms also produce similar results: COMB-1DUR-0.4CAN-0.4SAL obtains 80.1%. For N = 4 and N = 10, SF-DUR obtained 87.7%, 93.7% respectively, and the best combinations obtained a 3% absolute improvement in *RPA*. Figure 3.9 shows the evaluation results for the test set. It is worth mentioning that raw pitch accuracies are very similar to the ones obtained in the training set. Also note that the best performing combinations are those that give the highest weight to the salience function with highest raw pitch accuracy (SF-DUR), and lower weights to other salience functions (different ones depending on the value of N). As future work, it would be interesting to study the influence of pitch range in the performance of each method, and use this to improve the combination method.

Results can be further improved using the salience-based neighbourhood refinement method (RCOMB) presented in Section 3.4.3. For a single estimate, this is observed in Table 3.10, where the combination RCOMB-1MAR-1DUR obtains around 3% more raw pitch accuracy than the best performing salience function (SF-DUR), and 22.7% higher than the method which achieves the second highest accuracy (SF-MAR). Figure 3.9 shows that a weighted combination (e.g. RCOMB-0.5MAR-1DUR-0.2CAN) can improve the results from SF-DUR around 7% with N >1, or up to 4.5% with N = 1. The refined combination achieves up to 99.2% raw pitch accuracy with N = 10, while the best salience function (SF-MAR) obtains 94.2%. Additionally, the 95% confidence interval is smaller with the combination (98.9% - 99.5%) than with SF-SAL (93.1% - 95.3%). We also observe that considering the salience of pitch estimates in the refinement step is crucial for a better performance, especially for small values of N, as we can see in the difference between RCOMB (refinement considering pitch salience) and RNSCOMB (refinement without considering pitch salience) in Figure 3.9.

Finally, we study the influence of the width of the Gaussian function used in the combination method. We evaluate the estimations obtained with different values of the standard deviation (σ), ranging from 0.05 to 1 in semitones. Even though the results slightly vary with the specific combination, we observe some general trends. The highest accuracy for N = 10 is obtained with the default value $\sigma = 0.2$. The accuracy decreases with lower values of σ, since the combination only creates salience peaks if the pitches estimated by different methods are very close to each other. On the other hand, wider Gaussians (up to $\sigma = 0.8$) increase the accuracy for N = 1 (less

than 1 pp), since more distant pitches can be combined. However, if N increases, the accuracy decreases with wider Gaussians, because of the higher interference between all combined pitches.

3.5.4 Proposed metrics

We have focused so far on evaluation measures such as raw pitch accuracy, raw chroma accuracy and overall accuracy, which are useful to get a general understanding of the performance of the algorithms. However, we can gain further insights on their behaviour by means of the proposed metrics. For instance, in Table 3.10 we observe that the octave jumps ratio (*OJ*) is higher in methods where no tracking is performed, such as salience functions, as opposed to melody extraction algorithms. We also observe that the proposed neighbourhood refinement technique increases pitch continuity between correct estimates in chroma, since RCOMB has a lower *OJ* than COMB. The difference between *WRCA* and *RCA* shows that algorithms such as ME-DUR and ME-DRE estimate pitches at a closer octave to the ground truth octave, in comparison to ME-SAL or ME-FUE, since the latter present a higher difference. The *CC* measure is useful to obtain information about both smoothness and accuracy of the extracted melodic contour, since it combines *WRCA*, *NJ* and localisation of jumps. As an example of the usefulness of this measure, we observe that it allows us to differentiate between COMB and RCOMB, and to gain knowledge about their behaviour which can not be obtained with traditional MIREX measures. Both methods obtain relatively similar *RPA* and *RCA* scores, but there is an important difference in *CC*. The novel metric reflects the fact that pitch sequences estimated by RCOMB are much smoother thanks to the application of the refinement, which is a desirable property of a pitch estimation method, e.g. for visualisation purposes.

3.5.5 Generalisability study

In order to measure the reliability of the proposed dataset, and thus the validity of the obtained results, we perform a study based on Generalisability Theory (GT) (Urbano et al., 2013). GT is based on Analysis of Variance (ANOVA) procedures, and allows to differentiate between the sources of variability in evaluation results, which could arise from differences between algorithms, music excerpts, or the interaction effect between algorithms and music excerpts. Ideally, all variance should be due to differences between algorithms and not due to variability of the excerpts. If the considered music excerpts are very varied, or if differences between systems are too small, then we need many excerpts to ensure that our results are reliable.

The GT study has two stages: a Generalisability study (G-study), which estimates variance components on the evaluation results for each of the metrics, and a Decision study (D-study), which computes reliability indicators for a larger set of excerpts, based on the previous analysis of variance. We calculate two commonly used indicators: the index of dependability Φ, which provides a measure of the stability of abso-

lute scores, and the generalisability coefficient $E\rho^2$, which provides a measure of the stability of relative differences between systems (the closer to one the better). For our evaluation, we obtain values of Φ and $E\rho^2$ over 0.97 for *CC*, as well as for pitch and voicing detection metrics. This indicates that the variability of the scores was mostly due to differences between algorithms and not to differences between excerpts, which validates the obtained results. According to Salamon & Urbano (2012), some of the melody extraction datasets used in MIREX obtain the following values of Φ for raw pitch accuracy, when evaluating a larger set of state-of-the-art algorithms: ADC04 (Φ=0.86), MIREX05 (Φ=0.81), or INDIAN08 (Φ=0.72). Large scale collections for text information retrieval, obtain on average $E\rho^2$ = 0.88 and Φ=0.72 (Urbano et al., 2013). The proposed dataset is thus very reliable (Urbano et al., 2013), especially in comparison with some of the collections used in MIREX for audio melody extraction evaluation.

3.6 Timbre-informed melody pitch estimation

In this section we try to adapt a supervised approach for melody pitch estimation to symphonic music, where we exploit the fact that instruments are known in advance. In previous sections, we evaluated an approach based on probabilistic latent component analysis (PLCA) with predefined instrument templates (Benetos & Dixon, 2011). Those templates were created using recordings of single instruments from the RWC collection. Such basis are appropriate for classical music recordings, but they are not specific for symphonic music recordings, which commonly contain string sections (e.g. violin section). We here further study the application of timbre-informed spectrogram decomposition approaches for melody pitch estimation on a symphonic music dataset.

We use an efficient latent variable model for multiple-f_0 estimation, based on an ERB-scale time-frequency representation. According to Benetos & Weyde (2015a), ERB offers a compact representation, at the cost of losing the shift-invariance abilities in comparison to other methods using CQT, due to the non-linearity with respect to log-frequency. The transcription model is based on PLCA with pre-extracted spectral templates for several instruments.

In our experiments, we first create spectral basis for symphonic music transcription, then study the effect of pre- and post-processing to emphasise melody pitches, and we finally expand the original templates in order to adapt the spectral basis to the timbral characteristics of the target music signal.

3.6.1 Pitch template extraction

We first create a set of pre-extracted templates from instruments found in symphonic music, using the rendition of a chromatic scale by the Sibelius[41] Sounds (Lite) library

[41] http://www.avid.com/sibelius

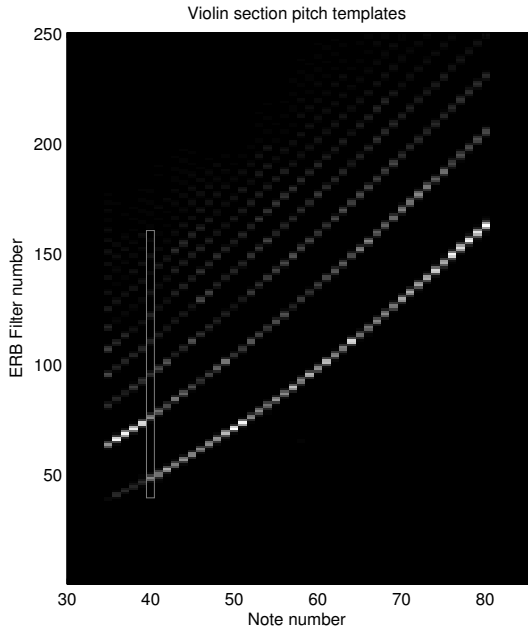

Figure 3.10: Violin section templates for sustain state.

for each of the considered instruments (MIDI note range between brackets): flute (60-96), oboe (58-91), clarinet (50-89), bassoon (34-72), horn (41-77), trumpet (52-86), viola section (48-89), violin section (55-100), cello section (36-77) and doublebass section (28-63). We create the templates using mezzoforte dynamics for all instruments and detaché articulation for strings, since notes should be separated to create the templates.

One spectral template is extracted for each pitch, instrument, and for each sound state corresponding to the states in the evolution of a note (attack, sustain, and decay). To create the template, the input audio signal goes through a set of 250 filters, with frequencies linearly spaced between 5Hz and 10.8kHz on the ERB scale. The output of each filter is divided into frames (every 23ms), and for each (log-)frequency (w) and time (t) indexes, we compute the rms magnitude $V_{w,t}$. The templates are extracted by using one-component PLCA on the output of the filters.

Figure 3.10 represents the spectral templates learnt for each note played by the violin section in the sustained state. The box corresponds to the range visualised in Figure 3.11, which presents a comparison between the learnt spectral templates for a clarinet and a violin section (C4 note in sustained state). The clarinet template presents stronger odd harmonics in comparison to even harmonics.

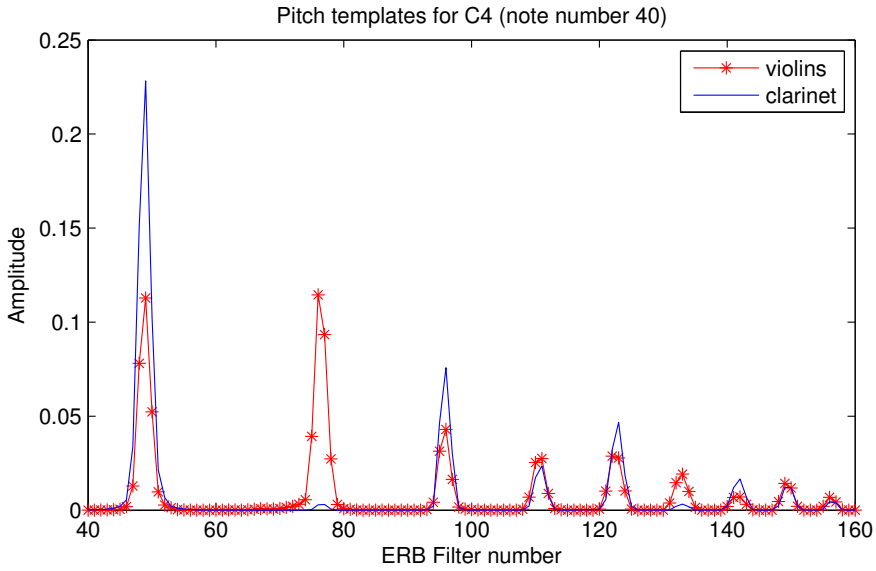

Figure 3.11: Comparison of a violin section and clarinet templates for a C4 note.

3.6.2 Multipitch detection

During multipitch detection, the ERB representation $V_{w,t}$ is approximated with a bivariate probability distribution $P(w,t)$, which is decomposed into a series of spectral templates per pitch, and instrument, as well as probability distributions for each of them. The model is formulated as:

$$P(w,t) = P(t) \sum_{q,p,s} P_t(w|q,p,s) P_t(s|p) P_t(p) P_t(q|p) \tag{3.12}$$

where q denotes the sound state, p denotes pitch, and s denotes instrument source. $P(t) = \sum_w V_{w,t}$, which is known, $P(w|q,p,s)$ is a 4-dimensional tensor that represents the pre-extracted spectral templates per sound state q, pitch p and instrument s. $P_t(s|p)$ is the instrument source contribution per pitch over time, $P_t(q|p)$ corresponds to the time-varying sound state activation per pitch and $P_t(p)$ corresponds to the multipitch detection output.

We use the expectation-maximization (EM) algorithm to iteratively estimate the unknown model parameters: $P_t(s|p)$, $P_t(p)$ and $P_t(q|p)$. During the Expectation step, the following posterior is computed:

$$P_t(q,p,s|w) = \frac{P(w|q,p,s) P_t(s|p) P_t(p) P_t(q|p)}{\sum_{p,s} P(w|p,s) P_t(s|p) P_t(p) P_t(q|p)} \tag{3.13}$$

Unknown model parameters are updated in the Maximization step as:

$$P_t(s|p) = \frac{\sum_{w,q} P_t(q,p,s|w)V_{w,t}}{\sum_{s,w,q} P_t(q,p,s|w)V_{w,t}} \tag{3.14}$$

$$P_t(p) = \frac{\sum_{w,s,q} P_t(q,p,s|w)V_{w,t}}{\sum_{p,w,s,q} P_t(q,p,s|w)V_{w,t}} \tag{3.15}$$

$$P_t(q|p) = \frac{\sum_{w,s} P_t(q,p,s|w)V_{w,t}}{\sum_{q,w,s} P_t(q,p,s|w)V_{w,t}} \tag{3.16}$$

No update rule is considered for $P(w|q,p,s)$, since it is considered fixed in the model. In order to control the polyphony level and instrument contribution in the resulting transcription, we make use of sparsity constraints on $P_t(p)$ and $P_t(s|p)$. The multipitch estimation result is given by $P(p,t) = P(t)P_t(p)$.

Finally, we perform a 5-sample median filtering for note smoothing, and no thresholding in order to compute a pitch activation function.

3.6.3 Pre- and post-processing

Given the frequency range of the melodies in this dataset, we study the effect of the application of an increasing linear frequency weighting (with weight K_{pre}) in the spectrogram prior to transcription, which is applied to both the learnt templates and the input representation. We also analyse the effect of such weighting to pitches in the salience as a post-processing step (with weight K_{post}).

3.6.4 Template adaptation

Given the difference between instruments, microphones and room acoustics between the training and testing set, there are also differences between the template dictionary from the training dataset and the spectral shape of the instruments to be transcribed. We now investigate if the unsupervised expansion of the dictionary using the data under analysis helps improving melody pitch estimation results. The method for template expansion is similar to (Benetos et al., 2014), which uses a conservative transcription step, allowing only a few notes to be transcribed. The spectral shape of the detected notes is then extracted and used to adapt the template dictionary.

In our case, we first perform an initial estimation of the pitch activations, using the pitch templates we created for symphonic music. We then perform thresholding on the global pitch activation matrix, in order to find the most salient notes in the signal, which are used to obtain the pitch template of the instrument playing that note. We investigate the use of two different methods for finding salient notes: the first one employs a global threshold, computed as a fraction of the maximum activation (we try with different constants), and the second one uses a time-dependent threshold, computed as a fraction of the maximum activation in each frame. The motivation for the second method is the fact that we want to capture the notes with are more

salient with respect to concurrent notes (present at the same time). This will reduce the interference from the partials coming from other instruments (which play non-predominant notes).

After thresholding, we perform dictionary expansion, by first collecting all spectra from the detected notes, and then extracting the templates from each set of spectra. Similarly to Benetos et al. (2014), we collect the spectra that correspond to each pitch p by applying a binary spectral mask of a harmonic comb h_p. Instead of artificially creating the harmonic comb in ERB scale, the mask is created by applying a low threshold on the pitch templates of the violin section. The collection of spectra collected for each pitch in the recording $\hat{V}^{(p)}$ is computed as:

$$\hat{V}^{(p)} = V_{w,t} \circ h_p \tag{3.17}$$

where $\circ$ denotes the Hadamard (element-wise) product, and $V_{w,t}$ corresponds to the signal spectrogram. From $\hat{V}^{(p)}$ we create new pitch templates, following two different strategies. The first approach is to obtain a single spectral representation which becomes the additional template: the reduction from a set of spectra to a single spectrum is performed using PLCA with a single component. Since this set of spectral templates may actually correspond to different instruments which sequentially play the most salient notes, we also explore a second approach that deals with the extraction of multiple additional templates N_{temp}. To do so, instead of using instrument specific activations as Benetos et al. (2014), for each of the pitches we perform k-means clustering with cosine similarity to group similar spectra. We then perform single component PLCA on each of the clusters with more that 3 elements, to obtain the additional templates (each element corresponds to the spectrum of a note, detected in a single frame). Finally, the transcription is executed again with the expanded dictionary. The maximum number of clusters (N_{cl}) is set as: $N_{cl} = max(N_{temp}, 3)$. In our experiments, we also studied the effect of changing the number of additional templates.

3.6.5 Results

We evaluate the approaches by their ability to make the melody pitch more salient, as in previous sections in this chapter. We first compare the use of spectral templates from symphonic music vs. templates created from single orchestra instruments. Results show an increase in the median raw pitch accuracy of 4.3% points when using symphonic music templates in comparison to default templates used by Benetos & Weyde (2015a). The largest improvement in *RPA* is however obtained with pre- and post-filtering. Figure 3.12 shows the results obtained with different weights, all with symphonic music templates. Pre-filtering has a higher impact on the results in comparison to post-filtering, but both increase the *RPA* with increasing values of K_{pre}, and K_{post}. When they are combined, they further improve the results, achieving 53.9% *RPA* with $\{K_{pre}$=16, $K_{post} = 128\}$, and up to 55.9% with $\{K_{pre} = 128, K_{post} = 128\}$.

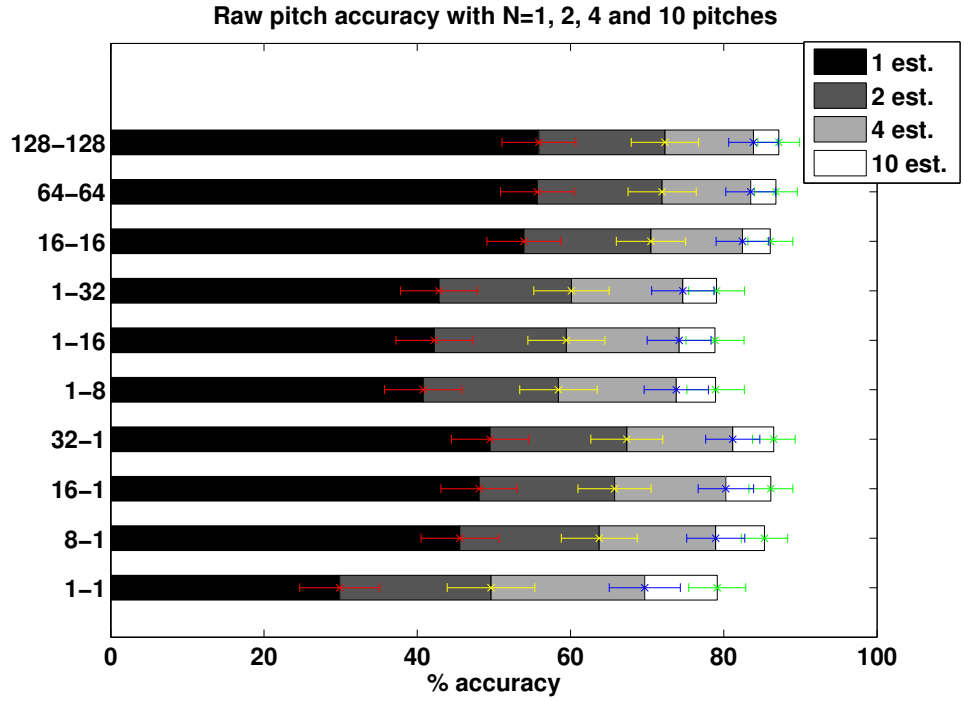

Figure 3.12: Effect of pre (first number) and post (second number) filtering in the raw pitch accuracy. E.g. 16-1 corresponds to $K_{pre} = 16$, and $K_{post} = 1$.

Finally we analyse the effect of template adaptation. The results of a representative set of the experiments is shown in Figure 3.13. Our first remark is that the raw pitch accuracy is higher with $K_{pre} = K_{post} = 64$ in comparison to $K_{pre} = 16$, also when using template adaptation. The best results are obtained when clustering is employed ($N_{cl} = 3$), since it helps differentiating between the detected spectra which may correspond to different instruments. A frame-based threshold generally produces higher accuracies than a global threshold: since we aim at collecting the spectra of the notes that are more predominant than the concurrent other notes, it is more effective to use a threshold based on the strength of the pitch activations in each frame.

Further work deals with using a different representation, which may be beneficial for template adaptation and expansion. For instance, with a representation such as VQT or CQT it would be possible to shift the extracted templates from one pitch to neighbouring pitches. It would also be interesting to study the use of instrument activation matrices to adapt the instruments' spectral templates, instead of creating additional templates as Benetos et al. (2014). Further work would also be to perform cross-validation to learn the best parameters (e.g. for pre- and post-filtering).

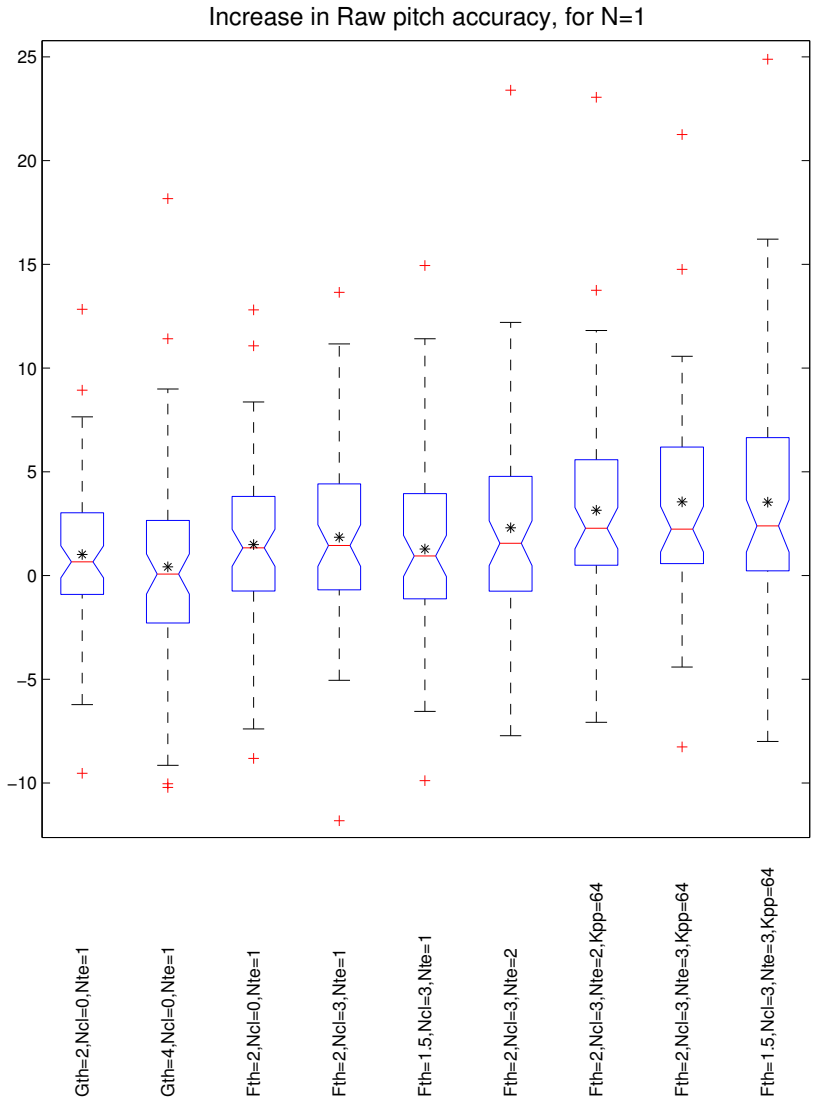

Figure 3.13: Increase in raw pitch accuracy due to template adaptation. $K_{pre} = K_{post} = 16$, except those notated with Kpp=64, for which $K_{pre} = K_{post} = 64$. Gth: global threshold, Fth: frame-based threshold. Ncl: number of clusters, Nte: number of additional templates extracted. The horizontal line in each box corresponds to the median, and the asterisk to the mean value.

3.7 Conclusions

This chapter presents an evaluation of state-of-the-art pitch estimation algorithms for melody extraction on symphonic music recordings. The selection of approaches was based on their relevance in the field, their performance in open evaluation campaigns, as well as their availability.

The main conclusion is that symphonic music is a very challenging context for melody extraction, and that most algorithms present lower pitch estimation accuracies than in other musical contexts such as vocal music. The pitch salience function which most algorithms use as an initial stage of their algorithms is generally not appropriate for this kind of data. For instance, a salience function based on harmonic summation is very dense, presenting high salience on multiple pitches, which makes melody pitch tracking very complicated. In addition, some approaches present a suboptimal melody pitch tracking after pitch salience computation, which further reduces melody extraction accuracy. We have seen that the most accurate melody extraction algorithm in this data uses a pitch salience function based on a source-filter model, which also performs best amongst the evaluated multipitch representations. One of the main features of this salience function is that it is much sparser than other approaches, since it tries to maximise the salience of the pitches which are likely to correspond to the lead melodic instrument. This approach creates a mid-level representation of the music signal under analysis in an unsupervised fashion, including pitch and part of timbre information of the lead instrument.

In the case of multipitch estimation approaches, we have seen that in order to improve the estimation of the melody pitch, it is useful to adapt to the characteristics of this dataset. This is exemplified in a supervised approach by using spectral basis containing orchestral sections, as well as emphasising higher pitches where the melody is more commonly found. We demonstrated that it is also useful to expand the dictionaries of spectral basis used with templates learnt from the music signal being analysed.

We have also analysed the characteristics that make melody pitch estimation in symphonic music complex, and which make humans and algorithms disagree more in their estimations. In terms of musical characteristics, melodic range and note density have a clear negative correlation with accuracy results obtained by people when singing the melody. In the case of algorithms, the highest (negative) correlation is with note density, and results suggest that algorithms are less affected by melodic range than humans, as long as pitches are kept within their limits of operation.

One of the problems of the evaluated approaches is that they do not learn from a given set of annotated examples. This would be very useful to adapt to the melodic and signal-related features of a given musical context, and especially to deal with different melody definitions. In Chapter 4 we propose melody extraction approaches which combine unsupervised and supervised learning, and we focus the evaluation on both pitch and voicing estimation, using a wide range of musical data.

Chapter 4

Advancements in Melody Extraction

4.1 Introduction

The goal of this chapter is to analyse the benefits of exploiting data-derived knowledge in audio melody extraction, using both supervised and unsupervised methods. We propose and evaluate several approaches, starting from more traditional rule-based algorithms, and increasingly exploiting available data. We consider both instrumental and vocal music, including a wide range of music material, from genres such as pop, rock, symphonic music, opera and jazz.

In this chapter, we address both tasks: pitch estimation and voicing detection, in contrast with Chapter 3, which focused on pitch estimation. An important conclusion from Chapter 3 is that a method based on a source-filter model (Durrieu et al., 2010) achieves the highest raw pitch accuracy in symphonic music. As we have seen, this unsupervised melody extraction method adapts better to the characteristics of the data under analysis. Another conclusion obtained from MIREX results (see Section 2.7.4) is that pitch contour-based approaches (Salamon & Gómez, 2012) achieve very good voicing detection and overall accuracy in other types of musical data (especially on vocal data). We hypothesise that the combination of both approaches leads to a better overall melody extraction accuracy, considering both pitch estimation and voicing detection. Furthermore, we propose transitioning from heuristics-based melody tracking to a data-driven approach, in order to replace hand-crafted rules by a machine learning classifier. This allows automatically learning the characteristics of melodic pitch contours in a given dataset, and eases the incorporation of new features, as it avoids the need to manually devise rules to exploit them.

Figure 4.1 presents the building blocks of the melody extraction methods proposed in this chapter. The main steps are: pitch salience estimation, pitch contour creation and melody tracking, with each of them in turn consisting of one or more separate

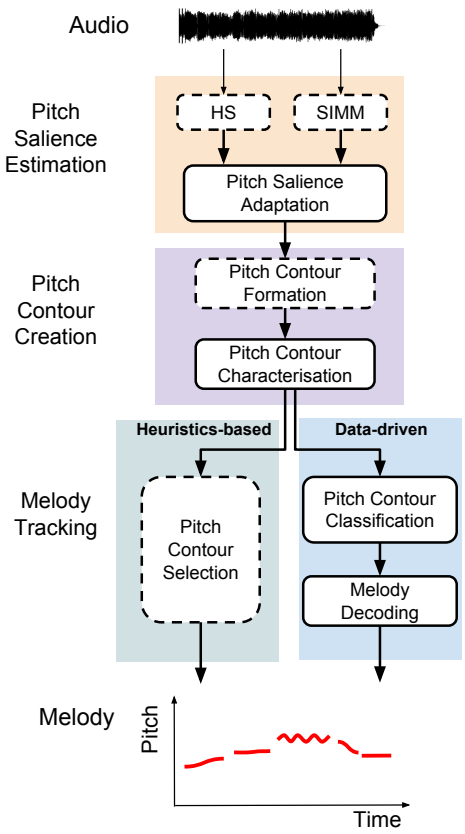

Figure 4.1: Block diagram of the proposed melody extraction methods.

processing blocks. Blocks represented with a solid line contain contributions in this chapter.

We first propose methods for pitch salience estimation (Section 4.2), based on a source-filter model (Durrieu et al., 2011). We use a Smooth Instantaneous Gaussian Mixture Model (SIMM) to model the leading voice, and create a melody-oriented pitch salience function, which is then adapted for pitch contour formation. We compare several salience functions, using metrics related to the pitch estimation accuracy, and the predominance of the melody over the rest of pitched elements. From the proposed salience functions, we form pitch contours using peak streaming (Salamon & Gómez, 2012) (Section 4.3). Melody tracking is then performed on the created set of contours, following either a heuristic approach (Section 4.4) based on Pitch Contour Selection (PCS) (Salamon & Gómez, 2012), or a data-driven approach (Section 4.5), based on Pitch Contour Classification (PCC) and Viterbi decoding (Bittner et al., 2015). We then propose novel timbre, spatial and tonal features for pitch contour characterisation (Section 4.6), which we use for data-driven melody tracking. Finally, we propose a method for estimating multiple melodic lines (Section 4.7), for which we

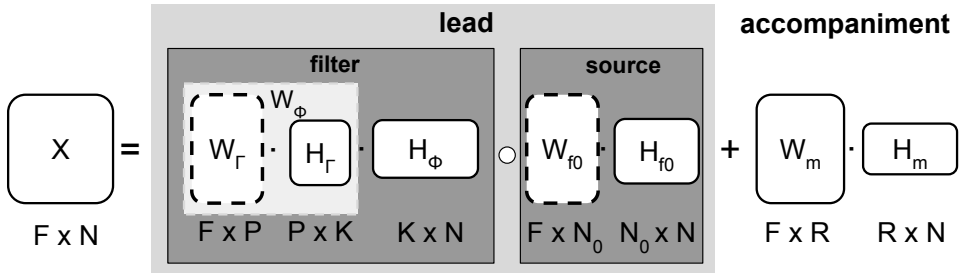

Figure 4.2: SIMM model for the leading voice, and accompaniment. Dashed lines refer to the matrices which are fixed, while the rest are iteratively estimated (see Section 2.4.4)

adapt the pitch contour classification to the fact that multiple melody contours may be concurrently present. We also propose a novel method for the simultaneous decoding of multiple melodic lines, which characterises and models contour transitions in the data.

Our study is carried out on two datasets: Orchset, and MedleyDB, and the evaluation is focused on the standard melody extraction accuracy metrics. We mainly focus on two melody definitions: 1) the f_0 curve of the predominant melodic line drawn from a single source (MEL1) and 2) the f_0 curve of the predominant melodic line drawn from multiple sources (MEL2). However, in Section 4.7, we also consider a third definition, which includes the f_0 curve of all melodic lines drawn from multiple sources (MEL3). The evaluation shows that learning from data (in both supervised and unsupervised fashions) improves melody extraction results in a wide range of music material. Additionally, in Section 4.2 we use pitch salience estimation metrics to evaluate our approaches, and source separation metrics in Section 4.4.5, since we also evaluate melody extraction in a source separation context.

4.2 Pitch salience estimation

As introduced in Section 4.1, we hypothesise that combining a pitch salience function based on a source-filter model (SIMM, see Section 2.4.4 or Durrieu et al. (2011)) with pitch contour-based melody tracking will lead to improvements in melody extraction. However, pitch contour creation in (Salamon & Gómez, 2012) is performed with a pitch salience function based on Harmonic Summation (HS), which presents very different characteristics in comparison to a salience function based on SIMM. The creation of pitch contours for melody extraction is based on the assumption that melody pitches are more salient, and that the range of their salience values is similar. However, a salience function based on a source-filter model presents a large range of values, since this NMF-based method does not prevent values (activation weights) from being very high or very low. In this section, we therefore propose two different salience functions which aim at adapting the characteristics of a salience function based on a source-filter model to a melody tracking stage based on pitch contours.

Method	Salience	Description
HS	HS	Harmonic Summation (Salamon & Gómez, 2012)[42]
SIMM	H_{f_0}	Source activations in SIMM (Durrieu et al., 2011)
CB	HS$\cdot H_{f_0}$	Combination of Harmonic Summation + SIMM
EW	$ES \cdot H_{f_0}$	Energy-based SIMM normalisation

Table 4.1: Pitch salience function overview.

Figure 4.2 represents the blocks of source-filter model (lead+accompaniment), where several parameters need to be specified: the number of bins per semitone (U_{st}), the number of possible spectral shapes in the accompaniment (R), the number of atomic filters in W_Γ (K), and the maximum number of iterations (N_{iter}). Parameter estimation is based on Maximum-Likelihood, with a multiplicative gradient method (Durrieu et al., 2010), updating parameters in the following order for each iteration: H_{f_0}, H_Φ, H_m, W_Φ and W_m. N corresponds to the number of frames, and F the number of frequency bins of the spectrogram. Note that H_{f_0} corresponds to the pitch salience function computed using SIMM.

The first proposed approach (CB) combines a salience function based on a source-filter model (H_{f_0}) (Durrieu et al., 2010, 2011) with a salience function based on harmonic summation (HS) (Salamon & Gómez, 2012)), as detailed in Section 4.2.1. The second approach (EW) uses an estimate of the energy of the melody to as detailed in Section 4.2.2. Both approaches employ Gaussian filtering, since we hypothesise that such smoothing is useful to make melody pitches more salient, particularly in the case of "ensemble" sounds. Table 4.1 presents an overview of the four considered pitch salience methods.

We reuse code from Durrieu[43] and Essentia[44] (Bogdanov et al., 2013), an open source library for audio analysis with a different implementation of (Salamon & Gómez, 2012) compared to MELODIA[45]. Our source code is available for research reproducibility[46].

4.2.1 Combining source-filter models and harmonic summation

In order to adapt H_{f_0} for pitch contour based tracking, we propose its combination (CB) with a Harmonic Summation salience function (HS), since pitch contour creation was originally adapted to this kind of representation (Salamon & Gómez, 2012).

The computation of HS in our method is based on the implementation of the open-source library Essentia. It starts with a Short Time Fourier Transform (STFT) as

[42] http://essentia.upf.edu
[43] https://github.com/wslihgt/separateLeadStereo
[44] http://essentia.upf.edu
[45] http://mtg.upf.edu/technologies/melodia
[46] https://github.com/juanjobosch/SourceFilterContoursMelody

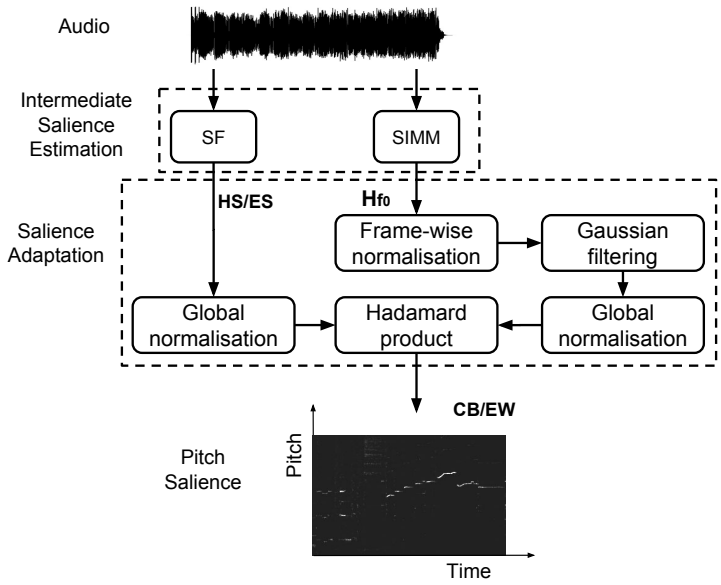

Figure 4.3: Proposed method for pitch salience estimation. SIMM outputs H_{f_0}; SF: salience function, either Harmonic Summation (outputs HS) or Energy-based Salience (outputs ES); Combining H_{f_0} with HS we obtain CB. Combining H_{f_0} with ES we obtain EW.

time-frequency transformation, applies Equal-Loudness Filters (ELF), finds spectral peaks positions and magnitudes, and then refines them using parabolic curve fitting, instead of using the instantaneous frequency (as Salamon & Gómez (2012)).

We normalise and combine the considered pitch salience functions HS(k,i) and $H_{f_0}(k,i)$, where k indicates the frequency index (bin) and i the frame index. The process is illustrated in Figure 4.3:

1. **Global normalisation** of HS, dividing all elements by their maximum value $max_{k,i}(HS(k,i))$.

2. **Frame-wise normalisation** (Fn) of H_{f_0}. For each frame i, divide $H_{f_0}(k,i)$ by $max_k(H_{f_0}(k,i))$.

3. **Convolution in the frequency axis** k of H_{f_0} with a Gaussian filter to smooth estimated activations. The filter has a standard deviation of .2 semitones.

4. **Global normalisation**, whose output is $\widetilde{H_{f_0}}$ (see Figure 4.4 (c)).

5. **Combination** by means of element-wise product: $S_c = \widetilde{H_{f_0}} \circ HS$ (see Figure 4.4 (d)).

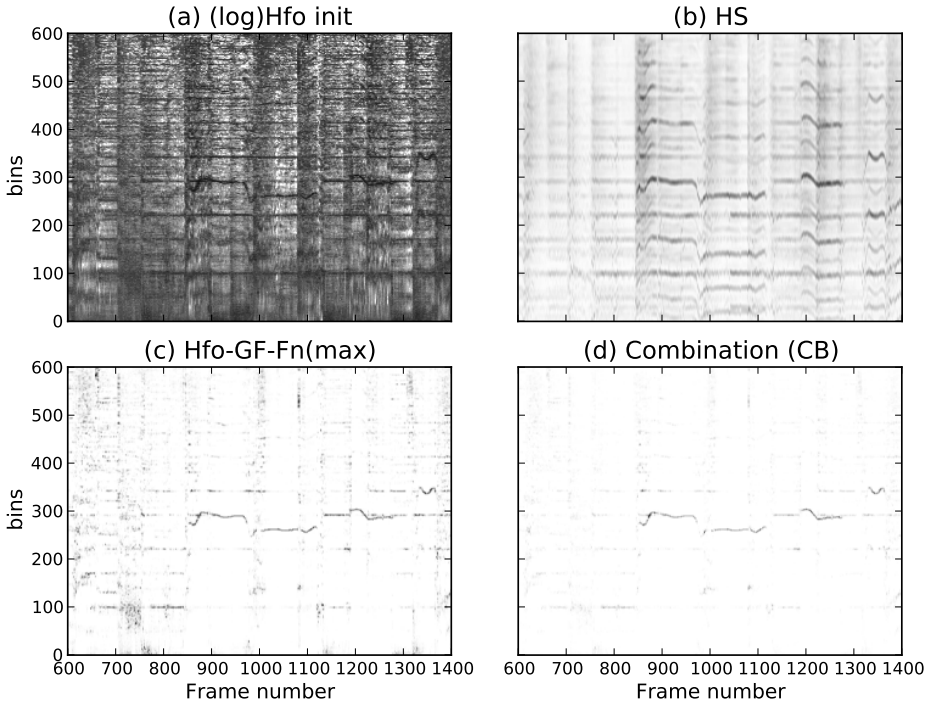

Figure 4.4: Time-frequency pitch salience representation of an excerpt from "MusicDelta_Beatles.wav" (MedleyDB) with (a) SIMM: $log_{10}(H_{f_0})$ is represented, to reduce the range of values for visualisation purposes) (b) Harmonic Summation: HS (c) H_{f_0} (max) normalised per frame and Gaussian filtered (d) Combination (CB).

4.2.2 Energy-based normalisation

In order to reduce the range of salience values of H_{f_0}, another possibility is to normalise each frame by its maximum salience. The drawback of this approach is that after normalisation, high salience values also appear in unvoiced frames. This turns voicing detection based on pitch contour selection into a complicated task. In order to reduce salience in unvoiced parts, we employ a frame-wise energy estimate of the melody line, using the same method as (Durrieu et al., 2010). For energy estimation, a HMM is employed, where each state corresponds to one bin of the pitch salience function (H_{f_0}), and the probability of each state corresponds to the estimated salience. Pitch continuity is considered in the transition probabilities, favouring smoothness in pitch trajectories. The energy of the melody source for each frame i (E_i), is computed using the decoded pitch sequence and the matrix decomposition computed before.

The estimated energy is used to create a matrix (ES) with the same size as H_{f_0}, in which all columns in one frame have a value equal to the estimated energy in that frame: $ES(k,i) = E_i$, $\forall k$. ES is then combined with H_{f_0} to create the salience function EW, following the same steps introduced in Section 4.2.1 (see Figure 4.3), with the difference that in the frame-wise normalisation (Fn), $H_{f_0}(k,i)$ is divided by

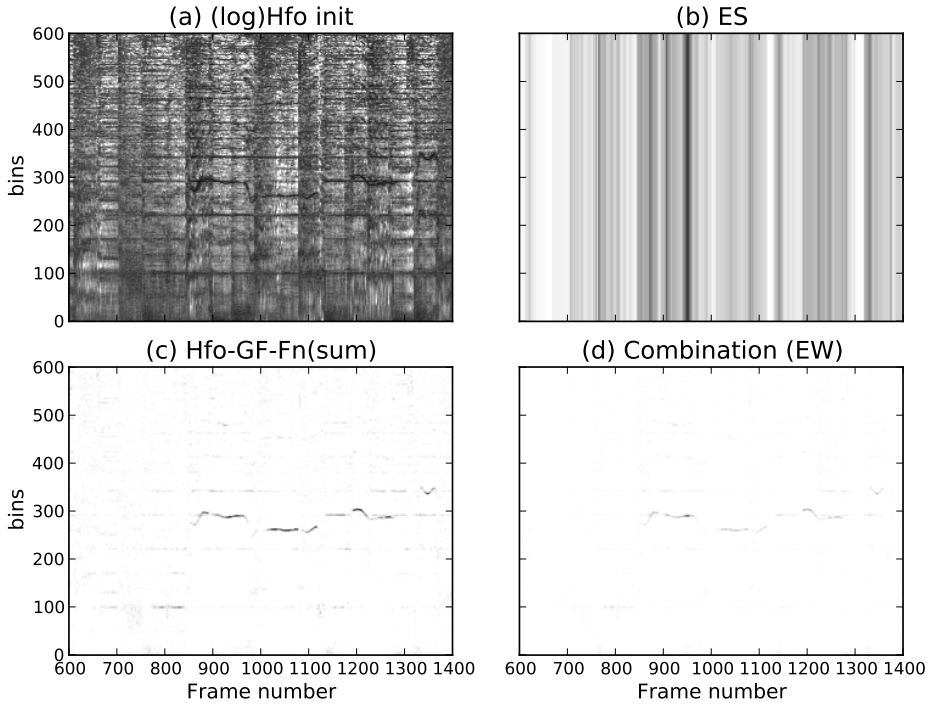

Figure 4.5: Time-frequency pitch salience representation of an excerpt from "MusicDelta_Beatles.wav" (MedleyDB) with (a) SIMM: $log_{10}(H_{f_0})$ is represented, to reduce the range of values for visualisation purposes) (b) Energy-based matrix: ES (c) H_{f_0} normalised per frame and Gaussian filtered (d) Combination (EW).

$\sum_k H_{f_0}(k,i)$, instead of the maximum value, also following Durrieu's approach. Figure 4.5 illustrates the combination.

4.2.3 Experimental setup

The proposed salience functions are evaluated and compared to two different state-of-the-art approaches in terms of their usefulness for melody extraction: the source-filter model from Durrieu et al. (2011) (H_{f_0}) and Harmonic Summation (HS) Salamon & Gómez (2012). Table 4.1 presents an overview of the evaluated methods. The pitch resolution (number of bins per semitone) is set to $U_{st} = 10$, sampling rate is 44100 Hz, hop size is 128 samples, and frequency limits are set to $f_{min} = 55$ Hz and $f_{max} = 1760$ Hz for all algorithms, following Salamon & Gómez (2012). This evaluation is conducted on MedleyDB and Orchset datasets, converted to mono as (left+right)/2.

As previously introduced, salience functions are evaluated from two different perspectives: pitch and salience estimation accuracy. To do so, we compute different metrics similarly to Salamon et al. (2011), using the ground truth melody: (1) the

frequency error of the salience function Δf_m, (2) the reciprocal rank score (RR_m) of the melody salience peak amongst the rest peaks, (3) the relative salience of the melody peak in comparison to the highest salience peak in a frame (S1), (4) the relative salience of the melody peak in comparison to the mean salience of the 3 highest peaks (S3). The latter quantifies thus the ability of a method to make the melody pitch more salient than the rest of the peaks, which is a key property of a melody-oriented salience function.

These evaluation metrics are frame-based, and they are not affected by the performance of the salience function in unvoiced frames. However, in a melody-oriented pitch salience function pitch salience should be zero in unvoiced frames, especially if the voicing detection algorithm depends on the salience function. This is the case of Salamon & Gómez (2012), since voicing detection depends on the contours' salience, which is derived from the pitch salience function.

We propose an evaluation measure to capture both the relative salience in comparison to other peaks in the same frame, and the ratio between the melody salience and the maximum salience in unvoiced frames, motivated by the salience peak filtering step in the contour formation process in (Salamon & Gómez, 2012). As previously introduced in 2.5.3, after computing the peaks of the salience function, a filtering process is carried out in two stages. First, peaks are filtered on a per-frame basis by comparing their salience to that of the highest peak. Second, the salience mean μ_S and standard deviation σ_S of the remaining peaks (in all frames) are computed. Peaks with salience below $\mu_S - \tau_\sigma \cdot \sigma_S$ are then filtered out. We thus propose the use of the Relative Voiced Salience (RVS) metric, in order to measure the difference between the salience in time-frequency bins corresponding to melody pitches, and the mean value of the maximum salience peaks in both voiced and unvoiced frames.

$$RVS = \frac{mean((S_{Mel} - \bar{M}_{S_i}))}{\sigma(M_{S_i})} \tag{4.1}$$

$$S_{Mel} = S(k,i), where\{k,i\} \in Mel \tag{4.2}$$

$$\bar{M}_{S_i} = \frac{\sum max_k(S(k,i))}{L} \tag{4.3}$$

where S corresponds to the pitch salience matrix, k corresponds to the frequency index (bin), i corresponds to the frame number, Mel corresponds to the sequence of time-frequency bins $\{k,i\}$ containing melody pitches and L is the total number of frames.

Finally we propose the metric PDD (Probability Density function Discontiguity) to measure how separated the distribution of saliences of melody pitches is, in comparison to the distribution of maximum saliences in unvoiced frames. The motivation for this metric is that, the more separated the distributions are, the easier will be to distinguish between voiced and unvoiced frames (containing a melody pitch or not,

respectively). We just consider the maximum salience on each unvoiced frame because low salience peaks will be removed during the initial frame-based peak filtering stage. To compute PDD, we first estimate the probability density function (pdf) of the salience of melody pitches, using Gaussian kernels:

$$p_m(S) = pdf(S_{Mel}) \tag{4.4}$$

and the pdf of the maximum salience values in unvoiced frames:

$$p_u(S) = pdf(max_k(S(k, i_u))), i_U = i \in U \tag{4.5}$$

where U represents the set of unvoiced frames (with no annotated melody pitch). We then calculate the area of intersection of the distributions p_m and p_u, and PDD is finally computed as

$$\text{PDD} = 1 - \int (p_m(S) \cap p_u(S)) dS, \text{PDD} \in [0, 1] \tag{4.6}$$

The higher the overlap between $p_m(S)$ and $p_u(S)$, the lower the value of this metric. The higher the value of PDD the more separated the distributions are, and thus, the easier will be to differentiate voiced from unvoiced frames.

4.2.4 Results

Figure 4.6 shows the evaluation results. In order to have an idea of the variance between excerpts, we compute the mean value of the metrics for each excerpt, and we then visualise evaluation results from all excerpts with a boxplot. The box shows the quartiles of the data, and the whiskers extend to show the rest of the distribution (except for points that are determined to be "outliers"). The line inside each box represents the median, and the mean value is represented with a star sign. Boxes are plotted with a "notch" to indicate the 95% confidence interval for the median.

Before analysing the results, we would like to note that the normalisations and energy weighting performed in the proposed EW salience function do not affect the frame-based (pitch and salience) evaluation metrics. Any difference in results between EW and H_{f_0} is thus only due to the proposed Gaussian filtering performed in each frame of the salience function. However, the proposed non frame-based voicing estimation measures are affected by all steps.

The lowest median frequency error (Δf_m) is obtained with CB, but differences amongst all approaches are not significant on MedleyDB (with both melody definitions). In the case of Orchset, H_{f_0} and the proposed methods obtain lower errors than HS. Note that on Orchset, results do not really represent the difference from the closest salience peak and the real melody pitch, since melody notes are played by orchestral sections, and individual instruments contributing to the melody are playing slightly different pitches. Additionally, ground truth pitches in Orchset are actually quantized at the

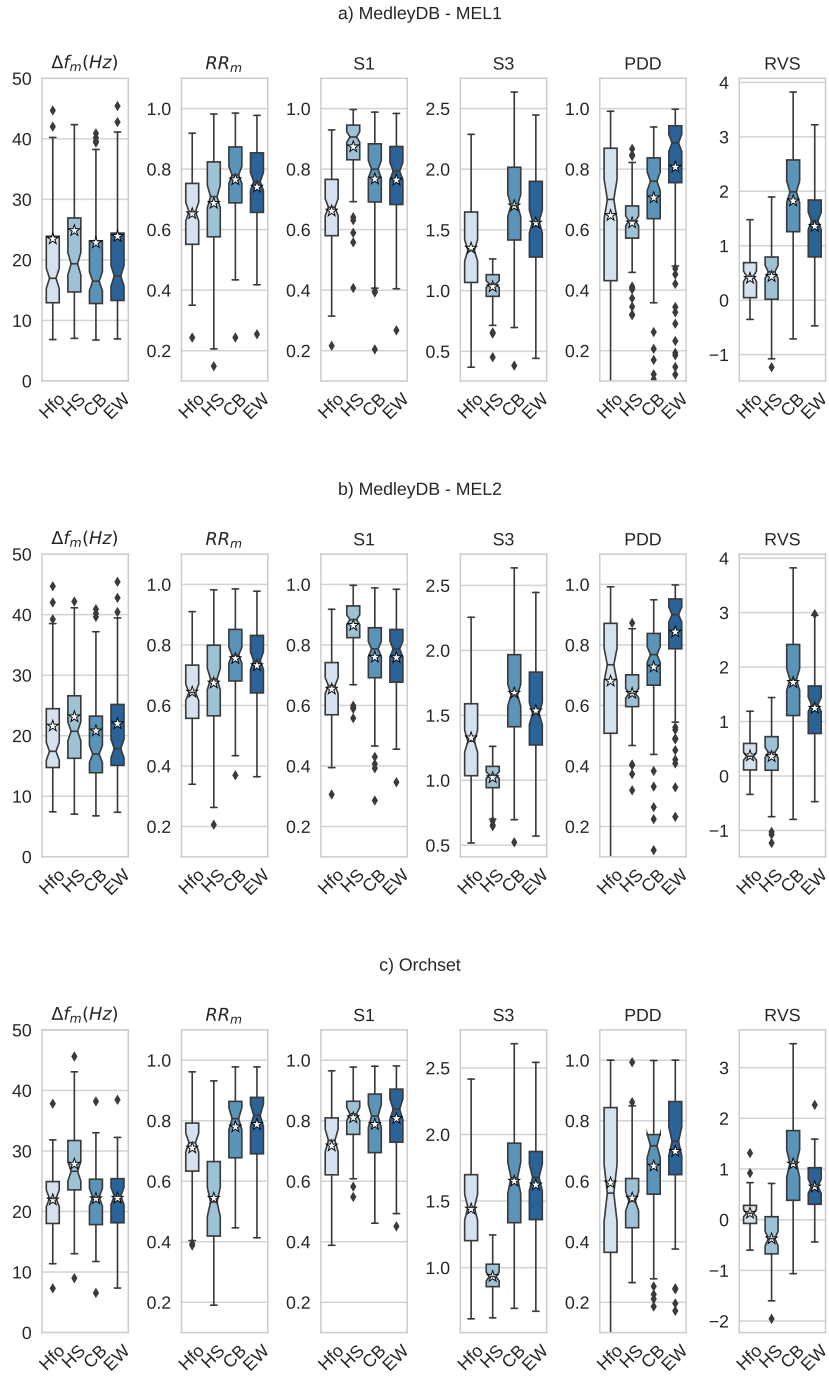

Figure 4.6: Salience function evaluation results a) Orchset. b) MedleyDB with the MEL2 definition c) MedleyDB with the MEL1 definition. Mean values represented with a star. Proposed methods correspond to CB and EW

semitone level, since they were derived from MIDI notes, without tuning information.

With regard to salience related metrics, we observe that the reciprocal rank RR_m of EW and CB is higher than the rest. We also observe that HS performs better on MedleyDB than on Orchset, while H_{f_0} behaves similarly in both datasets. The performance of CB is higher than EW on MedleyDB, presumably because of the synergy obtained when combining two salience functions. In the case of Orchset, the performance of the combination (CB) in comparison to EW is decreased since HS does not perform as well as H_{f_0} in orchestral data.

HS obtains the highest mean value of S1 for MEL2 on MedleyDB, however best S3 results are obtained with CB. As previously introduced, S1 compares the salience of the melody peak and the highest salience peak in a frame. S3 measures if the melody peak stands out from the other peaks of the salience function and by how much. These results show that HS achieves a high S1 score because the highest salience peaks do not actually present a high difference between them (the value of both S1 and S3 are close to one). HS obtains a median S3 of less than 1 on Orchset, which attending to the definition of the metric, means that (in average) the salience of the melody peak is smaller than the mean of the three highest peaks. H_{f_0} on the other hand presents a higher difference between the melody peak and the following most salient peaks. Finally, with regard to the proposed metrics RVS and PDD, highest values are obtained for all datasets with the proposed salience functions.

We thus conclude that the proposed salience functions reduce the estimation error of the melody pitch frequency (Δf_m) but not significantly with respect to the compared approaches. However, our approaches increase the salience of the melody pitch above the rest of concurrent pitches, particularly CB. Finally, PDD and RVS additionally show that the proposed salience functions are better at increasing the salience in melody pitches in comparison to the salience of unvoiced frames. This is a very important property of salience functions which are used for melody extraction using pitch contours.

4.3 Pitch contour creation

In this section, we use pitch contour formation and characterisation to create pitch contours from the proposed salience functions. The pitch contour formation process is proposed in (Salamon & Gómez, 2012), and summarised in Section 2.5.1, but we recall here the effect of the contour formation parameters, and describe our experimental setup.

4.3.1 Pitch contour formation

The first step in the contour creation process deals with detecting the peaks of the salience function. Then, non-salient peaks are filtered to minimise the creation of

noise contours. Peaks are first filtered on a per-frame basis: peaks below a threshold factor τ_+ of the highest salience peak in the frame are filtered out. The effect of this parameter is clear: the lower the value, the more peaks will remain in a given frame. In the second peak filtering stage, the salience mean μ_s and salience standard deviation σ_s of all remaining peaks (in all frames) are computed. Remaining peaks, with a salience below $\mu_s - \tau_\sigma \cdot \sigma_s$ are filtered. Therefore, τ_σ controls the accepted degree of deviation below mean salience: the higher the value, the more peaks will remain. The first filter ensures that the remaining peaks in a given frame are predominant, while the second filter helps reducing voicing false alarms. Peaks which are close to each other are then grouped into contours, following auditory streaming cues. In this process, small gaps in time are allowed, if they don't exceed a certain length (tc). Also small gaps in frequency are allowed if they are smaller than a certain pitch distance (d_p).

4.3.2 Pitch contour characterisation

Pitch contours are characterised by the following set of features: duration, pitch (mean and standard deviation), salience (mean and standard deviation), and total salience, following (Salamon & Gómez, 2012). In Section 4.6 we propose and evaluate novel features for pitch contour characterisation, related to tonal, timbre and spatial information, but experiments in Section 4.4 and Section 4.5 are conducted with the initial set of features.

Note that in our methods we do not exploit vibrato related characteristics, since preliminary experiments showed that they do not produce any significant variation in melody extraction results. This is also in agreement with the findings by Bittner et al. (2015), which show that vibrato features are by a large margin the least important ones for distinguishing melodic from non-melodic contours, in comparison to the rest of pitch contour features proposed by Salamon & Gómez (2012). In their experiments, they obtained that the presence of vibrato contributed to discriminating only around .03% of the training examples in their experiments on MedleyDB. In the same article, they mentioned that vibrato features considerably worsen melody extraction results for the generative model proposed by Salamon et al. (2012a).

4.3.3 Experimental setup

Unless otherwise stated, the results of the proposed methods in this chapter are reported with the same parameters values as Salamon & Gómez (2012) (τ_+= 0.9, τ_σ = 0.9, d_p= 80 cents), except for the maximum allowed gap in a contour, which we set to tc = 50 ms. An analysis of the effect of this and other parameters is presented in Section 4.4.4. Before using the created contours within complete melody extraction methods, we compute the pitch contour recall as Bittner et al. (2015). To do so, we measure the amount of reference melody that is covered by the extracted contours in terms of pitch overlap, by selecting the best possible f_0 curve from them.

	HS	CB	EW
MEL1	.60 (.22)	.65 (.20)	.64 (.19)
MEL2	.59 (.20)	.64 (.18)	.63 (.18)
Orchset	.45 (.21)	.58 (.18)	.68 (.19)

Table 4.2: Amount of reference melody covered by contours from different salience functions.

4.3.4 Results

Table 4.2 presents melody coverage results, computed for contours created with three different salience functions: Harmonic Summation (HS), and the proposed salience functions (CB and EW). These results represent the highest raw pitch accuracy that could be obtained by any melody extraction method using these sets of contours. The pitch contour formation process is thus also a very important step in contour-based melody extraction methods. The difference between salience functions is more noticeable in Orchset: HS obtains .45 while the proposed methods reach a much higher coverage, especially EW (.68). The highest coverage of MEL1 and MEL2 (MedleyDB) is obtained with CB, but is still relatively small, reaching only a maximum *RPA* of .65. On the one hand, the contour formation process sets a maximum limit on *RPA*, but on the other hand, it helps reducing voicing false alarms. Note that the results of each method depend on the parametrisation used in the contour creation stage (Salamon & Gómez, 2012), since more relaxed thresholds would lead to the creation of a higher amount of contours, and therefore to a higher melody coverage. In fact, it would be straightforward to create a method with maximum coverage, by simply creating contours on all time-frequency bins. However, melody decoding would be almost impossible, due to the very low precision in the contour creation stage.

4.4 Melody extraction based on pitch contour selection

4.4.1 Method

After studying salience functions results, we focus on complete melody extraction methods, in order to verify if the combination of source-filter models with pitch contour-based melody tracking leads to any improvements. In this section, we propose two methods that exploit the pitch salience functions proposed in Section 4.2 for pitch contour creation. From the extracted contours, we track the melody by means of a rule-based selection process (Salamon & Gómez, 2012), which we introduced in Section 2.5.3 as Pitch Contour Selection (PCS). Our contribution with respect to Salamon & Gómez (2012) is therefore an improved pitch salience function. We also consider a simple method which performs a *Combination of Pitch and Voicing* (CPV) directly from the output of two algorithms: estimated pitches from DUR (Durrieu et al., 2010) and voicing estimation from SAL (Salamon & Gómez, 2012), motivated

	Salience	Tracking	Voicing
DUR (Durrieu et al., 2010)	H_{f_0}	Vit(S)	Energy threshold
SAL, ESS (Salamon & Gómez, 2012)	HS	PCS	Salience-based
BIT (Bittner et al., 2015)	HS	PCC+Vit(C)	Probability-based
CBS	HS·H_{f_0}	PCS	Salience-based
EWS	EW	PCS	Salience-based
CPV	HS, H_{f_0}	PCS, Vit(S)	Salience-based
CBC	HS·H_{f_0}	PCC+Vit(C)	Probability-based

Table 4.3: Overview of the melody extraction methods evaluated in this chapter. SIMM: Smoothed Instantaneous Mixture Model (source-Filter model), HS: Harmonic Summation, Vit(S): Viterbi decoding on salience function, Vit(C): Viterbi decoding on contours, PCS: Pitch Contour Selection, PCC: Pitch Contour Classification.

by their respective accuracies in both tasks (see Section 4.1)

4.4.2 Experimental setup

We denote our methods based on Pitch Contour Selection as EWS and CBS, which exploit our novel salience functions EW and CB respectively. We now compare these approaches against the state-of-the-art methods exploited in this thesis: Durrieu et al. (2010) (DUR), and Salamon & Gómez (2012) method, which correspond to both SAL (Vamp plugin implementation) and ESS (Essentia implementation). The latter acts as our baseline since it uses the same contour creation implementation as our methods. SAL is evaluated with both the default voicing parameter value ($\nu = 0.2$) and with a manually tuned value (SAL*), which leads to the best results of this approach. The chosen value is $\nu = -1$ for MedleyDB (optimised for MEL1) and $\nu = 1.4$ for Orchset. As introduced in Section 2.5.3 parameter ν controls the amount of contours that are filtered out. The higher the value, the more contours will be kept, and therefore the higher the number of frames estimated as voiced. An overview of all melody extraction methods evaluated in this section and Section 4.5 is provided in Table 4.3.

4.4.3 Results

Figure 4.7 shows the obtained results for all metrics, in both Orchset and MedleyDB with both melody definitions. Table 4.4 , Table 4.5 and 4.6 present numeric results. We observe that CBS achieves the highest overall accuracy in both datasets. The difference in *OA* with EWS is however not significant (dependent t-test, significance level $\alpha = .05$) in any of the datasets. We recall that CBS is based on the salience function CB, which adapts H_{f_0} for contour formation, by combining it with HS. EWS is based on EW, which uses H_{f_0} and an initial estimate of the energy of the melody. Both of them use the same pitch creation algorithm and pitch contour selection for melody decoding.

Method	ν	*VR*	*VFA*	*RPA*	*RCA*	*OA*
EWS	.2	.67 (.14)	.28 (.12)	.62 (.21)	.70 (.16)	.60 (.15)
CBS	.2	.68 (.15)	.29 (.12)	.63 (.21)	.71 (.16)	.61 (.15)
CPV	.2	.71 (.15)	.31 (.13)	.65 (.21)	.73 (.16)	.57 (.14)
ESS	.2	.72 (.13)	.33 (.12)	.55 (.25)	.67 (.19)	.55 (.17)
DUR	-	1.00 (.01)	.95 (.05)	.65 (.21)	.73 (.16)	.36 (.16)
SAL	.2	.78 (.13)	.38 (.14)	.54 (.27)	.68 (.19)	.54 (.17)
SAL*	-1	.57 (.21)	.20 (.12)	.52 (.26)	.68 (.19)	.57 (.18)

Table 4.4: Mean results (and standard deviation) on MedleyDB - MEL1. Parameter ν refers to the voicing threshold used in the methods based on pitch-contour selection. The sign * refers to the results obtained with the best ν.

Method	ν	*VR*	*VFA*	*RPA*	*RCA*	*OA*
EWS	.2	.63 (.13)	.24 (.09)	.61 (.18)	.69 (.14)	.59 (.14)
CBS	.2	.65 (.14)	.24 (.09)	.62 (.19)	.70 (.14)	.60 (.15)
CPV	.2	.68 (.14)	.27 (.11)	.64 (.18)	.72 (.14)	.56 (.13)
ESS	.2	.69 (.12)	.28 (.10)	.53 (.22)	.65 (.17)	.54 (.17)
DUR	-	.99 (.01)	.95 (.06)	.64 (.18)	.72 (.14)	.42 (.14)
SAL	.2	.76 (.12)	.33 (.12)	.52 (.24)	.66 (.17)	.53 (.17)

Table 4.5: Mean results (and standard deviation) on MedleyDB - MEL2. Parameter ν refers to the voicing threshold used in the methods based on pitch-contour selection.

Pitch related accuracies are similar in MedleyDB for both approaches, but the difference in *RPA* is significant on MEL1 ($p = .03$). The difference is greater in Orchset, where EWS obtains higher pitch estimation accuracies (*RPA*, *RCA*). This is due to the fact that EWS uses exclusively H_{f_0} for pitch estimation, while CBS combines this salience function with HS. As we have seen in Chapter 3, HS does not perform as well as H_{f_0} in the orchestral dataset, and thus the pitch estimation accuracy obtained without combining both salience functions (EW) is higher. Voicing related metrics are relatively similar in both methods for MedleyDB, but CBS presents less false alarms in Orchset.

In comparison with the other methods, we first observe that both proposed methods yield a significantly higher *OA* than ESS (baseline), for both datasets and both melody definitions. *OA* is also higher in comparison to the alternative approaches, for both MEL1 and MEL2 on MedleyDB. In the case of Orchset, only DUR yields a higher *OA* than the proposed methods, partially due to a very high recall. Note that DUR always obtains almost perfect recall on all datasets, but also very high false alarm rates, since this method outputs most frames as voiced. The influence of this fact on the overall accuracy depends on the amount of voiced frames of the dataset. Since Orchset mostly contains voiced frames (93.7%), it is beneficial for DUR, in contrast with MedleyDB, which contains full songs with large unvoiced portions, and therefore

Method	ν	*VR*	*VFA*	*RPA*	*RCA*	*OA*
EWS	.2	.52 (.09)	.37 (.21)	.65 (.20)	.78 (.13)	.41 (.15)
CBS	.2	.53 (.10)	.29 (.16)	.58 (.19)	.71 (.14)	.41 (.16)
CPV	.2	.49 (.21)	.43 (.28)	.67 (.20)	.80 (.12)	.34 (.16)
ESS	.2	.51 (.09)	.33 (.18)	.30 (.23)	.55 (.19)	.21 (.18)
DUR	-	1.00 (.00)	.99 (.09)	.66 (.20)	.80 (.12)	.62 (.20)
SAL	.2	.60 (.09)	.40 (.23)	.28 (.25)	.57 (.21)	.23 (.19)
SAL*	1.4	.81 (.07)	.57 (.25)	.30 (.26)	.57 (.21)	.29 (.23)

Table 4.6: Mean results (and standard deviation) on Orchset. Parameter ν refers to the voicing threshold used in the methods based on pitch-contour selection. The sign * refers to the results obtained with the best ν.

false alarms considerably reduce *OA*. CPV obtains lower overall accuracies than the proposed methods. This is especially evident in Orchset, due to the reduced voicing estimation recall.

The proposed approaches achieve a slightly but significantly lower *RPA* in comparison to DUR. This is actually related to the previously observed fact that DUR estimates most frames as voiced. Even though *RPA* is considered a pitch related metric, it is also affected by voicing estimation, since it compares estimated pitches with voiced ground truth pitches. If some of the melody contours are not created, or are erroneously filtered (e.g. due to a lower salience in comparison to the rest of the contours), this will affect both voicing related metrics and pitch related metrics. This is the case for our proposed methods EWS and CBS: while many frames are correctly identified as unvoiced, some contours which correspond to the melody are filtered or simply not created, which decreases pitch related accuracies. However, reducing the voicing false alarm rate helps achieving a better overall accuracy.

SAL and ESS obtain significantly lower pitch related accuracies (*RPA*, *RCA*) than the proposed methods, especially in orchestral material. Given that the only difference between them is the salience function, we can conclude that the accuracy improvement is due to our contributions to the salience function design presented in Section 4.2. This was expected from results in Section 4.2.4, which showed that the our novel salience functions are able to make the melody pitch more salient.

We would like to note that the difference between *RCA* and *RPA* is much higher in SAL than in the proposed methods, especially on Orchset. This shows that the kind of signal representation underneath our salience functions is very effective at reducing the amount of octave errors (Durrieu et al., 2011; Goto, 2004).

As expected, pitch related metrics (*RPA*, *RCA*) are about the same for CPV and DUR (they output the same pitches). This simple combination is already able to significantly improve overall accuracy results on MedleyDB in comparison to all evaluated state-of-the-art approaches except SAL, thanks to the highest pitch estimation accuracy obtained by DUR, and a low *VFA*. However, *OA* results are not as high as with

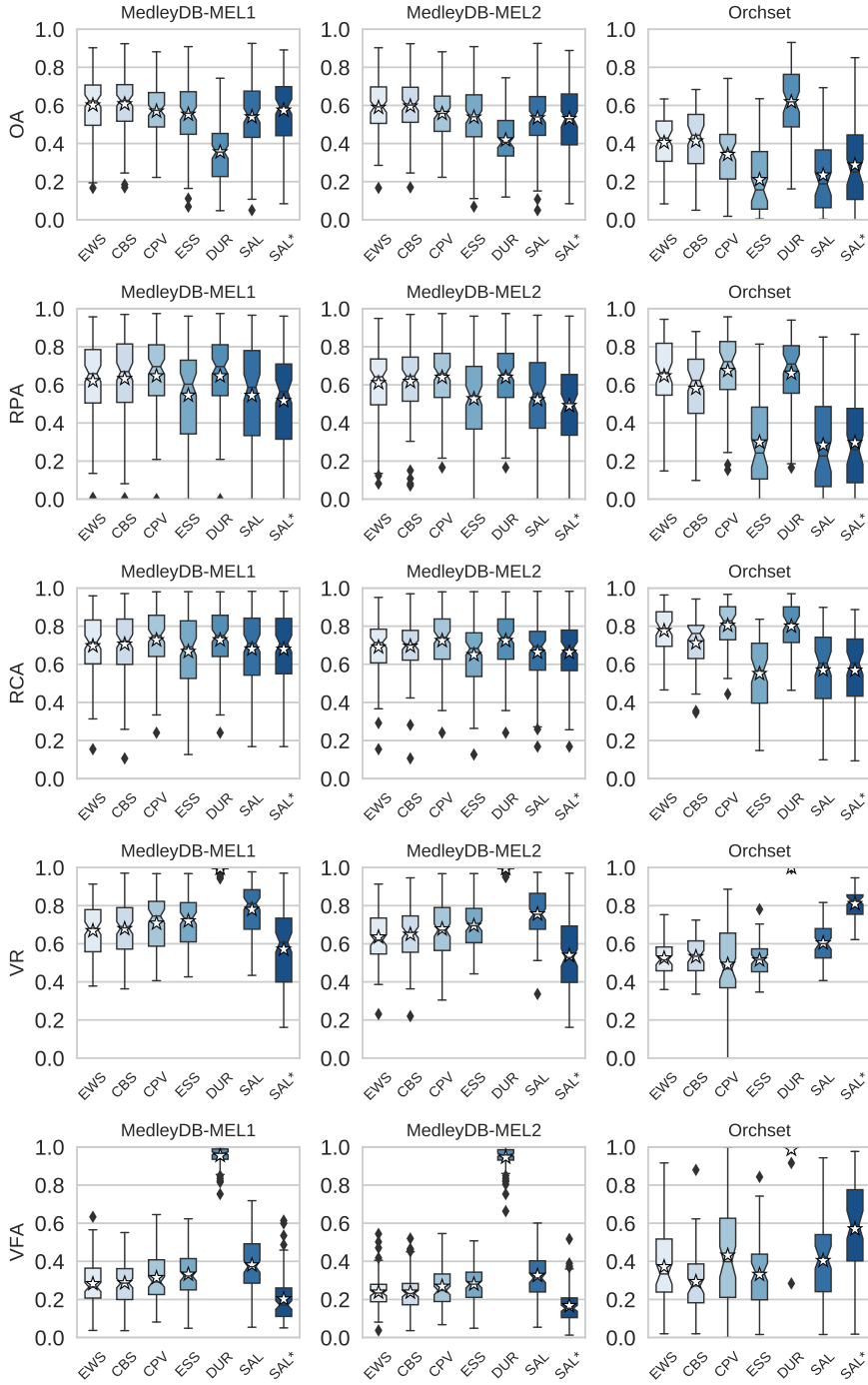

Figure 4.7: Evaluation results for MedleyDB with both MEL1 and MEL2 definitions and Orchset. "SAL*" denotes the results obtained with SAL with $\nu = -1$ for MedleyDB and $\nu = 1.4$ for Orchset. Proposed methods correspond to EWS, CBS and CPV.

DUR on Orchset, due to the lower recall.

The benefits of combining a source-filter model and a pitch contour based tracking method have become evident by now, and each of the proposed combination approaches has its advantages and disadvantages. The main advantage of CPV is its simplicity, and that it obtains as good *RPA* as DUR, which always obtains high scores in all datasets. The main disadvantage is that the contour creation process from SAL does not take advantage of the benefits of the pitch salience from DUR. This is the reason why it becomes important to integrate the source-filter model into the pitch contour creation process, as performed in CBS and CBC. One difficulty of the integration is that the salience function based on a source-filter model from DUR needs to be adapted to the pitch contour creation framework. However, we have seen in this section that this improves overall accuracy in both MedleyDB and Orchset.

4.4.4 Parameter tuning

Previous results can be further improved by adapting melody extraction parameters, initially designed to work with harmonic summation approaches, to the proposed salience functions, datasets, and melody definitions.

In this section, we first analyse the influence of Gaussian filtering (see Figure 4.3) on the complete melody extraction method CBS, by suppressing it from the pitch salience creation process. The effect is quite small on MedleyDB, but it helped improving pitch estimation on Orchset (4% points). This could be due to the small differences in the pitch played by the individual instruments contributing to the melody. As previously observed with the salience function evaluation results, by smoothing H_{f_0} we are able to make more salient the pitches of the notes played by orchestral sections in unison.

There are other parameters which affect different parts of the method. Figure 4.8 shows the effect of the number of iterations when computing H_{f_0} (N_{iter}), which affects the salience function creation; maximum allowed gap in the contour ($tc \in \{50, 75, 100\}$ ms), which affects the contour formation process (see Section 2.5.1), and the voicing tolerance parameter ($v \in \{-1, .2, 1, 1.4\}$), related to the final selection of melody contours (see Section 2.5.3). For the sake of clarity, we only show results from CBS, since the highest overall accuracy was obtained with this method. Results obtained with the rest of evaluated state-of-the-art methods are also presented, including the effect of the number of iterations N_{iter} on Durrieu's approach (DUR).

The best results in vocal music are obtained with few iterations, but complex data (such as instrumental, and especially orchestral music) benefits from a higher number of iterations. In any case, the influence of pitch salience creation parameters is relatively small in comparison to the influence of pitch contour tracking parameters. For instance, *OA* generally increases when the maximum gap between pitches in a contour is decreased from 100 ms to 50 ms in MedleyDB. This may be related to the noise added in unvoiced frames by the SIMM, which can partially be filtered in the

contour creation process. In the case of Orchset, the *OA* is higher when the we allow a larger gap (100ms instead of 50 ms), since this increases the amount of coverage by the created contours, which is especially convenient in this dataset, since it is mainly voiced.

The effect of the voicing parameter (ν) is evident: a higher value increases the voicing threshold and less contours are filtered, which is beneficial in Orchset. Setting a lower threshold is beneficial in MedleyDB with the MEL1 definition, since the amount of voiced frames is smaller. Default peak filtering parameter values (τ_σ, τ_+) provided good results in MedleyDB, but *OA* can be increased up to 60% in Orchset, by increasing τ_σ from .9 to 1.3 with $\nu = 1.4$. This allows a higher difference in salience below the salience mean during pitch contour creation, which is appropriate to deal with the larger dynamic range in classical music.

Regarding instrumentation, *OA* in MedleyDB vocal music is higher than in instrumental, but with the proposed method, we increased it in about 10 and 8 percentage points (pp) over the baseline (ESS) respectively. The improvement is even more evident in Orchset. According to the results, we can conclude that our salience function leads to a better accuracy than HS, for both single instruments and instrument sections.

CBS obtained 25 percentage points (pp) higher *OA* in MedleyDB (with the MEL1 definition, see Figure 4.8) compared to DUR, and slightly worse in Orchset (around 4 pp with the best parameters mentioned). Additionally, CBS generally needs less iterations (N_{iter}) compared to DUR to achieve the best results, which is very positive given the high computational cost of the estimation algorithm. In comparison to the approach by Salamon et al., we obtained 5 and 30 pp higher accuracy in MedleyDB (with the MEL1 definition) and Orchset respectively, using the best voicing parameter for each dataset in both algorithms (CBS and SAL*). This corresponds to about 10% and 100% relative increase, due to the low accuracy of SAL in Orchset.

The selection of parameters has been performed manually, but it could be done automatically by selecting the best performing configuration in a training set. Another possibility is to use a pitch contour classification approach (Bittner et al., 2015), by training a classifier to distinguish between melody and non-melody pitch contours using the proposed salience function based on a source-filter model, as described in Section 4.5.

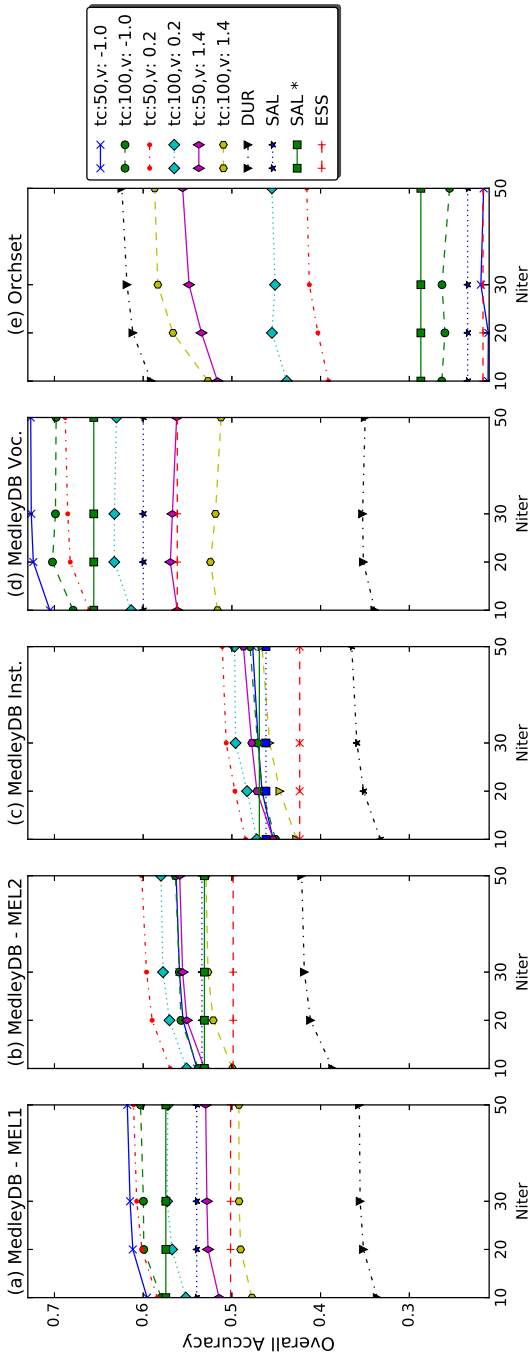

Figure 4.8: Overall accuracy vs. number of iterations, and influence of time continuity (tc), and voicing (v) parameters on the results obtained with CBS. Results for DUR, SAL and the baseline (ESS) given as a reference. "SAL*" denotes that $v = -1$ for MedleyDB and $v = 1.4$ for Orchset (a) MedleyDB (MEL1) definition; (b) MedleyDB (MEL2); (c) MedleyDB (MEL1), instrumental songs; (d) MedleyDB (MEL1), vocal songs; (e) Orchset.

4.4.5 Evaluation in the context of source separation

In this section, we study the effect of our method based on pitch contour selection on a framework for the separation of the lead instrument and accompaniment. The experiments are based on the separation approach by Durrieu et al. (2010), which requires an initial melody estimation stage.

We give the melody extracted with our CBS method as input to the separation algorithm, and compare the results against using the original melody estimate. We use the source separation evaluation framework and metrics described in (Vincent et al., 2006; Emiya et al., 2011): *Source to Distortion Ratio* (SDR), *Source to Interference Ratio* (SIR), and *Source to Artifacts Ratio* (SAR). SDR measures the overall quality of the separation, ISR is related to the spatial reconstruction of the sources, SIR is related to rejection of the interferences, and SAR to the absence of distortions and artifacts. Such evaluation requires that the mix corresponds to the instantaneous mixture of the sources.

In our case, we evaluate the separation of the mix into lead and accompaniment, and thus need to create these tracks, using the stems. We created a subset of the original MedleyDB dataset, which corresponds to the songs without bleed between stems, and which have only one melodic source (according to the metadata associated to each track in MedleyDB). Additionally, the longest file is discarded since during our experiments the separation has not been possible with Durrieu's separation method, due to the excessive use of RAM, which produced memory errors. While it would be possible to run the separation on smaller parts, the results would be different, as the algorithm requires the whole signal to estimate the parameters and compute the separation. This makes a total of 51 files, both vocal and instrumental, which were used for the evaluation. The list of files used in this evaluation is presented in Appendix B.

We evaluate the separation results for three different values of the voicing parameter in our method, and compared them with the original method by Durrieu (DUR). The proposed method leads to an improvement of the overall quality of the separation, for both melody and accompaniment, measured by the SDR metric, as shown in Figure 4.9. Our method obtains a median SDR value for vocals separation of 2.7 with both ($\nu = .2$ and $\nu = 1$), while the median SDR with Durrieu's approach is 1.4. In the case of the accompaniment, the highest SDR is 9.3, which is obtained by our method with $\nu = .2$, while DUR reaches 8.8. Results for other metrics are presented in Figure 4.10, which shows that DUR obtains the lowest SIR on the separation of the melody. This is due to the fact that it tends to estimate most frames as voiced. Similarly our method reduces the SIR with lower values of ν, since it reduces the amount of contours considered as melody. However, in the case of the accompaniment, DUR will have less interference from the melody, since it is performing the separation in most frames.

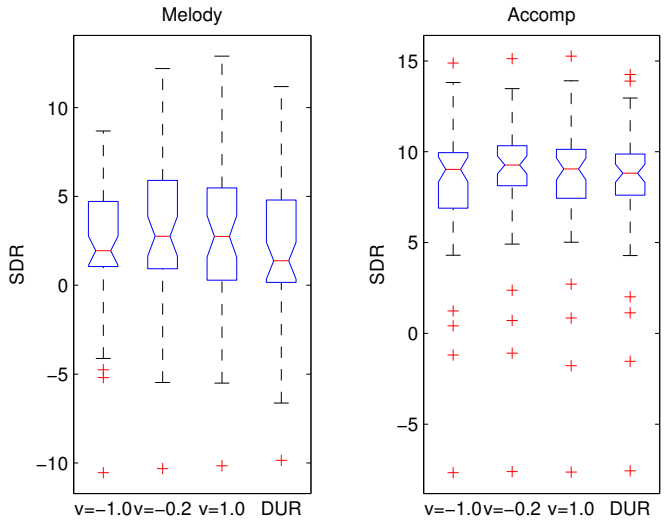

Figure 4.9: SDR source separation metric, when using Durrieu's source separation method with the original melody extraction (DUR), and with our proposed method, with 3 different values of the voicing parameter (ν)

4.5 Melody extraction based on pitch contour classification

4.5.1 Method

We have seen that salience functions based on source-filter models help improving a melody tracking method based on heuristic rules. However, such tracking method has some disadvantages: rules need to be manually defined, they are not valid for any kind of musical data, and it is difficult to add new features which may improve melody extraction accuracy, since additional rules should be created. As introduced in Section 2.5.4, several approaches have been proposed to avoid heuristic rules for contour-based melody tracking (Salamon et al., 2012a; Bittner et al., 2015). Bittner et al. (2015) concluded that see that the use of random forest outperforms the multivariate Gaussians proposed by Salamon et al. (2012a), and additionally, it is scalable to a much larger feature set.

In this section we propose the use of CB together with a melody tracking method based on pitch contour classification (using a random forest classifier), and study its advantages over the use of HS as salience function. We also analyse the difference between pitch contour selection (PCS) and pitch contour classification (PCC) for melody tracking, comparing four different methods. The first one (CBS) is based on the creation of pitch contours from CB, and employs pitch contour selection as the tracking method. The second one (CBC) combines CB with pitch contour creation from (Salamon & Gómez, 2012) and the contour classification strategy from (Bittner

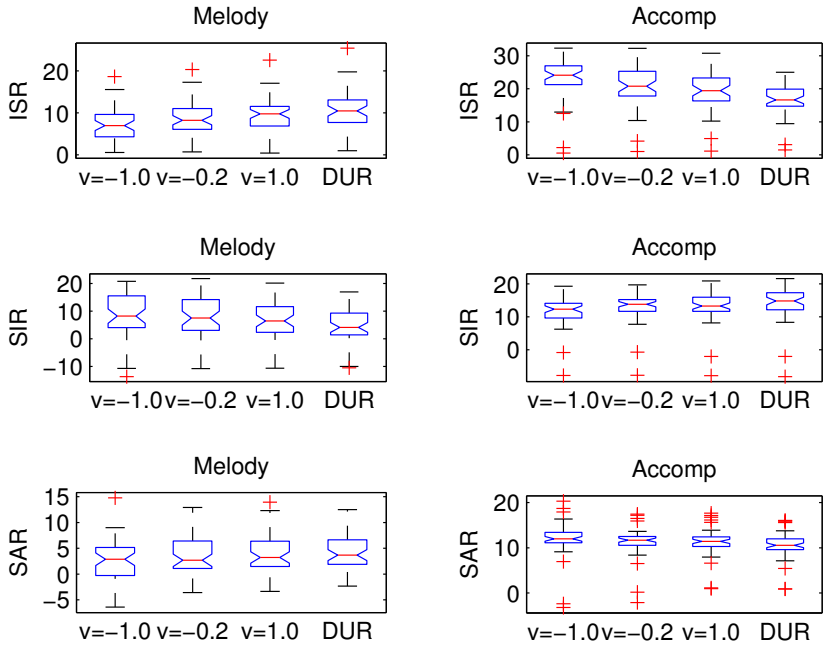

Figure 4.10: ISR, SAR and SIR source separation metrics, when using Durrieu's source separation method. with the original melody extraction (DUR), and with our proposed method, with 3 different voicing parameters

et al., 2015). Apart from the proposed methods (CBS, CBC), we also analyse the results obtained with Salamon & Gómez (2012), and Bittner et al. (2015) (BIT), which is the same approach as CBC but using HS as pitch salience function. Table 4.3 provides an overview of their main building blocks. As in Section 4.4, we include results from original implementation of Salamon & Gómez (2012) in MELODIA (SAL), and the implementation in the Essentia library (ESS), since our methods build on top of the latter.

4.5.2 Experimental setup

The evaluation was carried out using MedleyDB and Orchset datasets, following the standard MIREX evaluation methodology. The proposed combination methods (CBS, CBC), BIT and ESS use the same same set of pitch contours, since in our experiments they were all created with the same implementation (using Essentia), and with the same contour creation parameters and pitch contour characteristics (see Section 4.3). We recall that the pitch contour characteristics employed are: duration, pitch (mean and standard deviation), salience (mean and standard deviation), and total salience.

For the evaluation of classification-based methods (CBC, BIT), we followed Bittner

et al. (2015), and created train/test splits using an "artist-conditional" random partition on MedleyDB. For Orchset we created a "movement-conditional" random partition, meaning that all excerpts from the same movement must be used in the same subset: either for training or for testing. Datasets are randomly split into a training, validation and test sets with roughly 63%, 12%, and 25% of the songs/excerpts in the dataset, respectively. This partitioning was chosen so as to have a training set that is as large as possible while retaining enough data in the validation and test sets for results to be meaningful. In order to account for the variance of the results, we repeat each experiment with five different randomized splits in the case of MedleyDB, and 10 splits in the case of Orchset, given the smaller size of the dataset.

Similarly to previous sections, we set the same frequency limit for all algorithms: $f_{min} = 55$ Hz and $f_{max} = 1760$ Hz. The number of bins per semitone is 10, and the hop size was 128 samples, which corresponds to around 2.9 ms, given a sampling rate of 44100 Hz.

4.5.3 Results

Results for all evaluated algorithms are presented in Figure 4.11. Table 4.7 and Table 4.8 present numeric results for MedleyDB (MEL1 and MEL2 respectively) and Table 4.9 for Orchset.

Method	ν	*VR*	*VFA*	*RPA*	*RCA*	*OA*
CBS	.2	.68 (.15)	.29 (.12)	.63 (.21)	.71 (.16)	.61 (.15)
CBC	-	.73 (.16)	.38 (.17)	.58 (.23)	.63 (.20)	.59 (.15)
ESS	.2	.72 (.13)	.33 (.12)	.55 (.25)	.67 (.19)	.55 (.17)
BIT	-	.83 (.10)	.51 (.14)	.52 (.22)	.63 (.19)	.49 (.15)
SAL	.2	.78 (.13)	.38 (.14)	.54 (.27)	.68 (.19)	.54 (.17)
SAL*	-1.0	.57 (.21)	.20 (.12)	.52 (.26)	.68 (.19)	.57 (.18)

Table 4.7: Comparison of methods based on pitch contour classification and pitch contour selection. Mean results (and standard deviation) on MedleyDB - MEL1. Parameter ν refers to the voicing threshold used in methods based on pitch-contour selection.

Method	ν	*VR*	*VFA*	*RPA*	*RCA*	*OA*
CBS	0.2	.65 (.14)	.24 (.09)	.62 (.19)	.70 (.14)	.60 (.15)
CBC	-	.76 (.13)	.37 (.16)	.58 (.19)	.65 (.16)	.59 (.14)
ESS	0.2	.69 (.12)	.28 (.10)	.53 (.22)	.65 (.17)	.54 (.17)
BIT	-	.80 (.10)	.43 (.14)	.50 (.20)	.61 (.16)	.51 (.14)
SAL	0.2	.76 (.12)	.33 (.12)	.52 (.24)	.66 (.17)	.53 (.17)

Table 4.8: Comparison of methods based on pitch contour classification (CBC, BIT) and pitch contour selection (the rest). Mean results (and standard deviation) on MedleyDB - MEL2. Parameter ν refers to the voicing threshold used in methods based on pitch-contour selection.

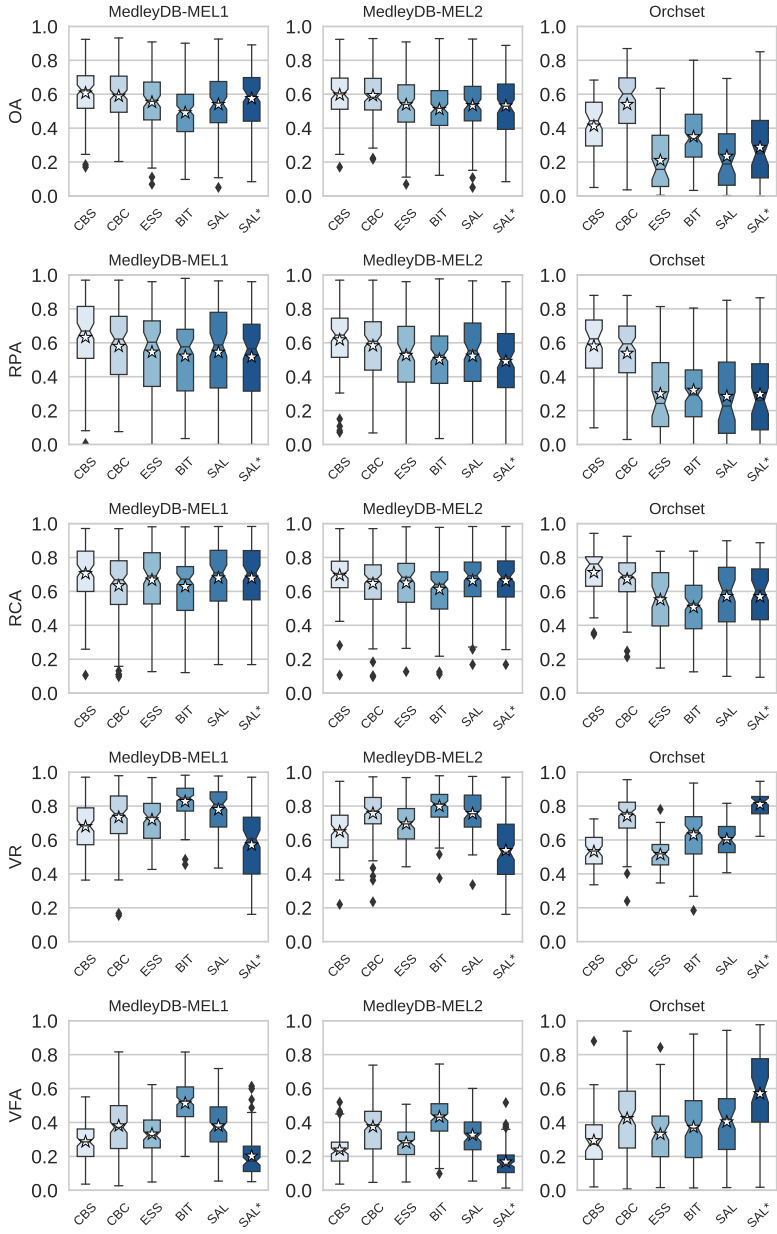

Figure 4.11: Comparison between approaches based on pitch contour selection and pitch contour classification. Results for all metrics, for MedleyDB with both MEL1 and MEL2 definitions and Orchset. "SAL*" denotes the results obtained with SAL with $\nu = -1$ for MedleyDB and $\nu = 1.4$ for Orchset

Method	ν	*VR*	*VFA*	*RPA*	*RCA*	*OA*
CBS	0.2	.53 (.10)	.29 (.16)	.58 (.19)	.71 (.14)	.41 (.16)
CBC	-	.74 (.12)	.42 (.22)	.54 (.20)	.67 (.14)	.54 (.19)
ESS	0.2	.51 (.09)	.33 (.18)	.30 (.23)	.55 (.19)	.21 (.18)
BIT	-	.63 (.16)	.37 (.22)	.32 (.19)	.51 (.16)	.35 (.18)
SAL	0.2	.60 (.09)	.40 (.23)	.28 (.25)	.57 (.21)	.23 (.19)
SAL*	1.4	.81 (.07)	.57 (.25)	.30 (.26)	.57 (.21)	.29 (.23)

Table 4.9: Comparison of methods based on pitch contour classification (CBC, BIT) and pitch contour selection (the rest). Mean results (and standard deviation) on Orchset. Parameter ν refers to the voicing threshold used in methods based on pitch-contour selection.

The first remark is that the proposed classification based method (CBC) yields a statistically significantly (t-test, significance level $\alpha = .05$) higher overall accuracy (*OA*) than SAL for both datasets and both melody definitions. The exception is in MEL1, in the case of SAL* (SAL with a voicing threshold optimised for MEL1): CBC-SAL* ($p = .48$). For the MEL2 definition CBS and CBC yield an *OA* that is significantly higher than all compared approaches. In the case of Orchset, CBC is also significantly better than CBS and the rest of approaches in terms of *OA*.

Salamon's approach (SAL, ESS) and Bittner's (BIT) perform relatively similarly on MedleyDB (the difference in *OA* is only significant on MEL1), but the usefulness of Bittner's classification-based method becomes evident on Orchset: with the same candidate contours as ESS, the *OA* increases considerably. This classification-based method is thus partially able to learn the characteristics of melody contours in orchestral music. As we previously introduced, Orchset is characterized by a higher melodic pitch range compared to most melody extraction datasets which often focus on sung melodies.

Regarding the use of the CB salience function instead of HS, we obtain similar conclusions as in the case of pitch contour selection. CBC provides a higher accuracy in comparison to BIT, showing that the proposed salience function improves melody extraction results also when combined with a pitch contour classification method. Once again, this is particularly evident in orchestral music.

By comparing the results of CBS and CBC we can assess the influence of the pitch tracking strategy, as both methods use the same contours as input, but CBS is based on a rule-based melody tracking (Pitch Contour Selection), and CBC on data-driven approach (Pitch Contour Classification). In MedleyDB, there is no significant difference between both methods in terms of overall accuracy, but the contour classification based method (CBC) has a higher voicing recall for both melody definitions, and CBS presents a lower *VFA*. This agrees with the findings from Bittner et al. (2015) who also compared between both pitch tracking strategies using HS as the salience function. In the case of Orchset, the difference in *OA* is evident between CBS-CBC, since the classification based approach tends to classify most frames as voiced, which is beneficial

when evaluating on this dataset, which is mainly voiced. As we saw in 4.4.4, we can increase the voicing tolerance parameter in CBS to improve *OA* results on Orchset.

Results show higher *RPA* and *RCA* values for CBS in comparison to CBC. However, this does not necessarily mean that the rule-based method has a better pitch estimation than the classification-based method. The reason of the higher value of *RPA* is because CBC does not guess "unvoiced" pitches, while CBS provides pitch estimates with a negative value when the frame is estimated as unvoiced. As introduced in Section 2.7.2, negative pitch estimations also count in the computation of the pitch estimation accuracy. We recall that voicing estimation in the contour classification method (Bittner et al., 2015) is based on filtering pitch contours with a low likelihood of being melodic. It would be straightforward to improve *RPA* and *RCA* results of the classification method, by giving a negative pitch estimate in the frames where contours have a low likelihood of being melodic. Such frames would be guessed as unvoiced, but pitch related accuracies would benefit of the unvoiced guess, in comparison to setting the melody estimate in those frames to 0. In any case, *RPA* is limited by the contour creation process: as we saw in Section 4.3 created contours are not able to cover all the annotated melody for any dataset or melody definition, which limits both the total melody recall and pitch related accuracies.

The most important difference between CBS and CBC is that CBC allows training a model to fit the characteristics of a dataset, avoiding parameter tuning in rule-based approaches. For instance, the set of rules from Salamon & Gómez (2012) used in CBS and SAL are not tuned to orchestral music, which explains why they obtain a lower *OA* on Orchset with the default parameters, in comparison to the data-driven methods (CBC and BIT respectively). Furthermore, if the salience function used is HS, the rule-based method does not reach the overall accuracy obtained with the classification based method (BIT) on Orchset, even with the best voicing parameter choice (SAL*).

4.6 Extended contour characterisation

One of the advantages of data-driven in comparison to heuristics-based approaches is that it is easier to include additional features, since we do not need to manually create rules to consider them. In Section 4.5 we used a Random Forest Classifier to automatically learn rules from a training dataset to discriminate between melodic and non-melodic contours, based on the following set of pitch contour features: duration, pitch (mean and standard deviation), salience (mean and standard deviation), and total salience. We denote this set as the default (DEF) feature combination.

In this section we propose a novel set of pitch contour features (φ) related to timbre, spatial and tonal information. We first introduce each of the features, and then analyse their distributions for melodic and non-melodic pitch contours. Finally, we present the results obtained when including them in a melody extraction method based on pitch

contour classification.

4.6.1 Timbre features

We have so far implicitly exploited timbre information during the estimation of the CB pitch salience function, by means of a source-filter model. However, we claim that the filter matrix computed when estimating the pitch salience function (see Figure 4.2) would also be useful to discriminate melodic from non-melodic contours, since it represents an estimate of the lead instrument's spectral envelope.

As introduced in Section 2.4.4, the lead instrument is modelled as: $\hat{X}_v = X_\Phi \circ X_{f_0}$, where X_{f_0} corresponds to the source, X_Φ to the filter, and the symbol $\circ$ denotes the Hadamard (element-wise) product. The filter matrix X_Φ has a size of FxN (where F corresponds to the number of frequency bins of the spectrogram and N to the number of frames).

We propose a set of timbre features for the characterisation of pitch contours, based on the well-known Mel-Frequency Cepstrum Coefficients (MFCC) (Davis & Mermelstein, 1980; Logan et al., 2000) where instead of using input spectrum as input, we consider the spectral envelope which is learnt in the filter matrix (X_Φ). A 13 coefficient MFCC vector (v_{MFCC}) is computed for each frame i (each column of the filter matrix), as:

$$v_{MFCC}[i] = MFCC(X_\Phi[k,i]) \tag{4.7}$$

where $k = 0,..,F$ and $i \in [i_s, i_e]$, where i_s and i_e correspond to the start and end frames of the target pitch contour. The timbre features for each contour correspond to the mean and standard deviation over the length of the contour ($[i_s, i_e]$) of each of the MFCCs (v^c_{MFCC}), where $c \in [1,13]$. We obtain a total of 26 timbre features:

$$\varphi_{timbre} = [\mu(v^1_{MFCC}), \sigma(v^1_{MFCC}), ..., \mu(v^{13}_{MFCC}), \sigma(v^{13}_{MFCC})] \tag{4.8}$$

Note that since we have a single filter shape per frame, two contours with the same start and end frame have the same timbre features. Therefore, such features are not useful to differentiate between parallel contours, but our hypothesis is that they help differentiating between voiced and unvoiced frames, therefore reducing voicing false-alarms.

4.6.2 Spatial features

One of the tasks of music producers is to combine multiple recorded sounds into one or more channels, which is known as mixing. When producers create stereo or other type of multichannel music signals, it is common that they add artificial attenuations and delays during the mixing process, in order to provide listeners with a sense of source localization (Marxer, 2013). In contrast to other type of signals, sources are commonly static within a music signal. Furthermore, it is also common to find the spatial position of certain sound sources preserved in a large set of music. For

instance, it is very common to find lead vocals positioned in the centre of the stereo field in popular western music, or to find music instruments positioned in a similar distribution as they would be found in a live orchestra performance.

For this reason, spatial information has been used in a variety of MIR tasks, e.g to decompose a music signal into different source contributions (Vinyes et al., 2006; Burred, 2009; Durrieu et al., 2010; Marxer, 2013; Ceron, 2014; Miron et al., 2016), to estimate multiple pitches (Zhang et al., 2012), or to improve musical instrument recognition (Bosch et al., 2012a).

In order to investigate if the localisation on the stereo field is useful to discriminate between melodic and non-melodic pitch contours, we propose two spatial features related with the position of the sources. To do so, we estimate the spatial location of the elements (time-frequency pairs) that form a pitch contour, and then compute the median value for all elements, as well as the standard deviation. The median value is chosen instead of the mean to lower the effect of outliers. The initial step is to compute the difference between the pitch salience functions of left (HS_L) and right (HS_R) channels.

$$\Delta^{HS} = log((HS_R)) - log((HS_L)) \tag{4.9}$$

In this case, we use HS instead of CB, since during the computation of the latter, the salience function based on a source-filter model is normalised by frame before the combination with HS, and therefore HS is equally useful to compute the difference in pitch salience between channels, with a lower computational load. We then compute the median difference in salience ($\widetilde{\Delta_C^{HS}}$) between both channels, in the time-frequency bins of each pitch contour C , as well as the standard deviation ($\sigma(\Delta_C^{HS})$).

$$\widetilde{\Delta_C^{HS}} = \widetilde{\Delta^{HS}}(k,i), where\{k,i\} \in C \tag{4.10}$$

$$\sigma(\Delta_C^{HS}) = \sigma(\Delta^{HS}(k,i)), where\{k,i\} \in C \tag{4.11}$$

where k and i correspond to the frequency bin and frame number respectively. We then map these values into the azimuth domain with the following transform function:

$$az(x) = 360 \cdot \frac{atan(e^x)}{\pi} - 90 \tag{4.12}$$

and finally normalize the azimuth:

$$\hat{az}(x) = az(x)/90, \hat{az} \in [-1,1] \tag{4.13}$$

Each contour C is thus characterised with two spatial features, which correspond to the normalised azimuth value of $\widetilde{\Delta_C^{HS}}$ and $\sigma(\Delta_C^{HS})$.

$$\varphi_{spatial} = [\hat{az}(\widetilde{\Delta_C^{HS}}), \hat{az}(\sigma(\Delta_C^{HS}))] \tag{4.14}$$

4.6.3 Tonal feature

We finally propose a feature related to the degree of fit of a contour in the tonality of the excerpt. Our main motivation is that this feature may help classifying contours which correspond to leading tones, or other passing notes which commonly belong to the melody but are not common in the musical key of the excerpt. To compute it, we first estimate the Harmonic Pitch Class Profile (HPCP) (Gómez, 2006) of the excerpt with hopsize of 1024 frames, and then compute a vector with the mean value of each pitch class across all frames v_P. The computed vector gives an indication of the tonality of the piece, and the amount of presence of all pitch classes. We then create a vector corresponding to the pitch class histogram of the pitches present in a given contour (v_H). Finally, we compute the Pearson correlation (r) between v_P and v_H, which indicates the degree of presence of the notes in a given contour in the whole excerpt.

$$\varphi_{tonal} = r(v_P, v_H) \tag{4.15}$$

4.6.4 Feature distributions

In order to inspect the potential usefulness of the proposed features for discriminating between melodic and non-melodic contours, we analyse their distributions in the set of contours created for all songs in our datasets (MedleyDB and Orchset). Since we only have melody pitch annotations, and no annotations for each contour, we consider a contour to be melodic if more than 50% of its elements overlap with with the annotated melody, or non-melodic otherwise. Figure 4.12 presents the distributions for the vocal excerpts in MedleyDB with the MEL1 definition. Figure 4.13 presents the disributions for MedleyDB with the MEL2 definition, and finally Figure 4.14 shows the distributions for Orchset. The visualised features include the default and novel ones: spatial, tonal and timbre features. We only plot the mean values of the proposed spatial and timbre features, not the standard deviations.

As expected, some of the features proposed in the literature already present noticeable different distributions for melodic and non-melodic contours. For instance, the mean pitch value is useful to discriminate between both classes when we restrict to vocal contours (Figure 4.12), since melodic contours pitch values will be limited to the vocal range. This feature could also be helpful to classify contours with a very low frequency as non-melodic, in both symphonic music (Figure 4.14), or in MedleyDB with the MEL2 definition (Figure 4.13). Salience features (both mean and standard deviation values) also seem useful, especially in MedleyDB. In the case of Orchset, they seem to be less useful, possibly because of the larger dynamic range in this kind of music: as seen in Figure 4.14, the distributions of melodic and non-melodic contours for salience related features are less differentiated that in MedleyDB (Figure 4.12 and Figure 4.13). We should note that the duration and total salience features

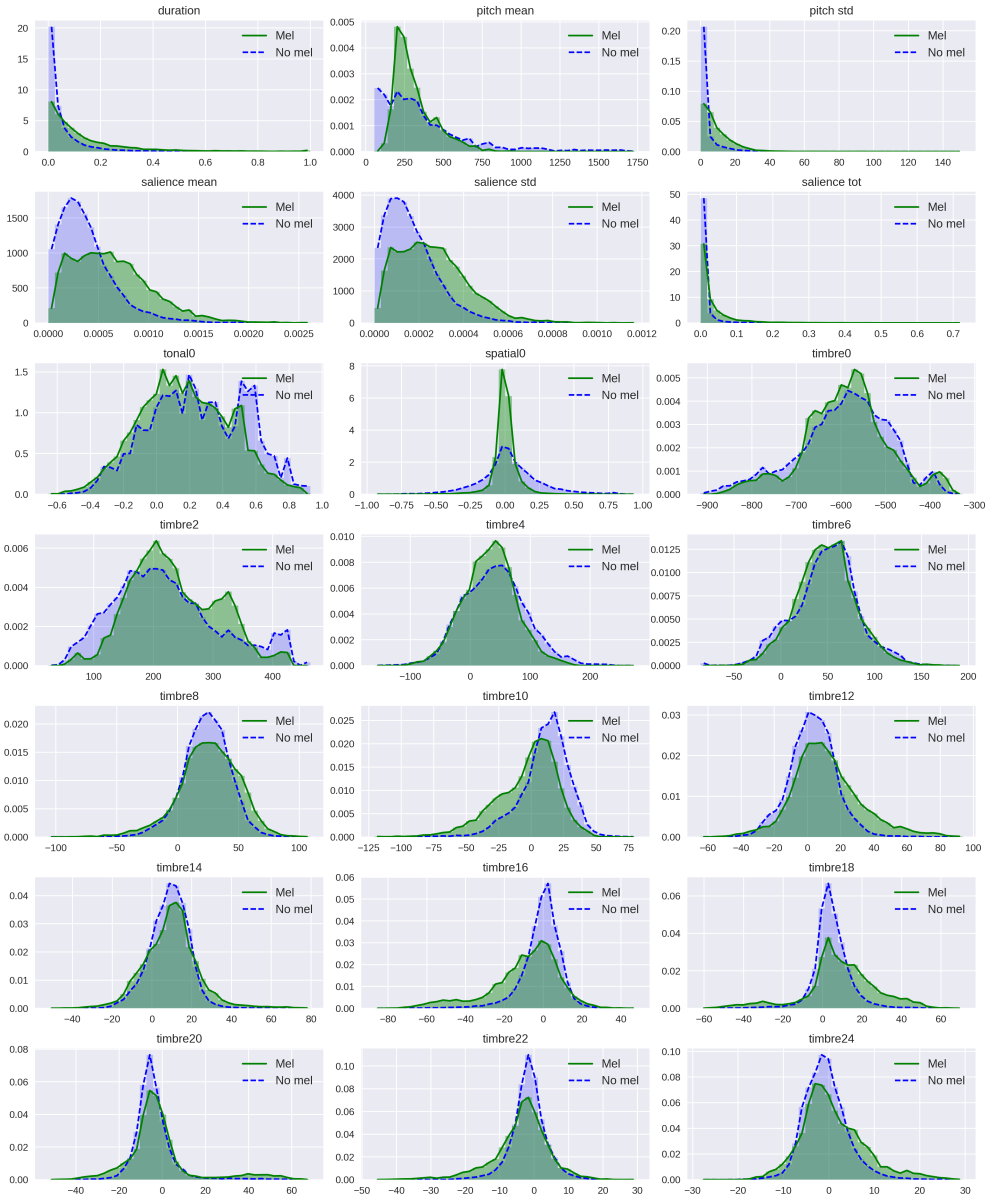

Figure 4.12: Distribution of contour features for melody (green, solid line) and non-melody (blue, dashed line) contours, in MedleyDB with the MEL1 definition for just vocal melodies.

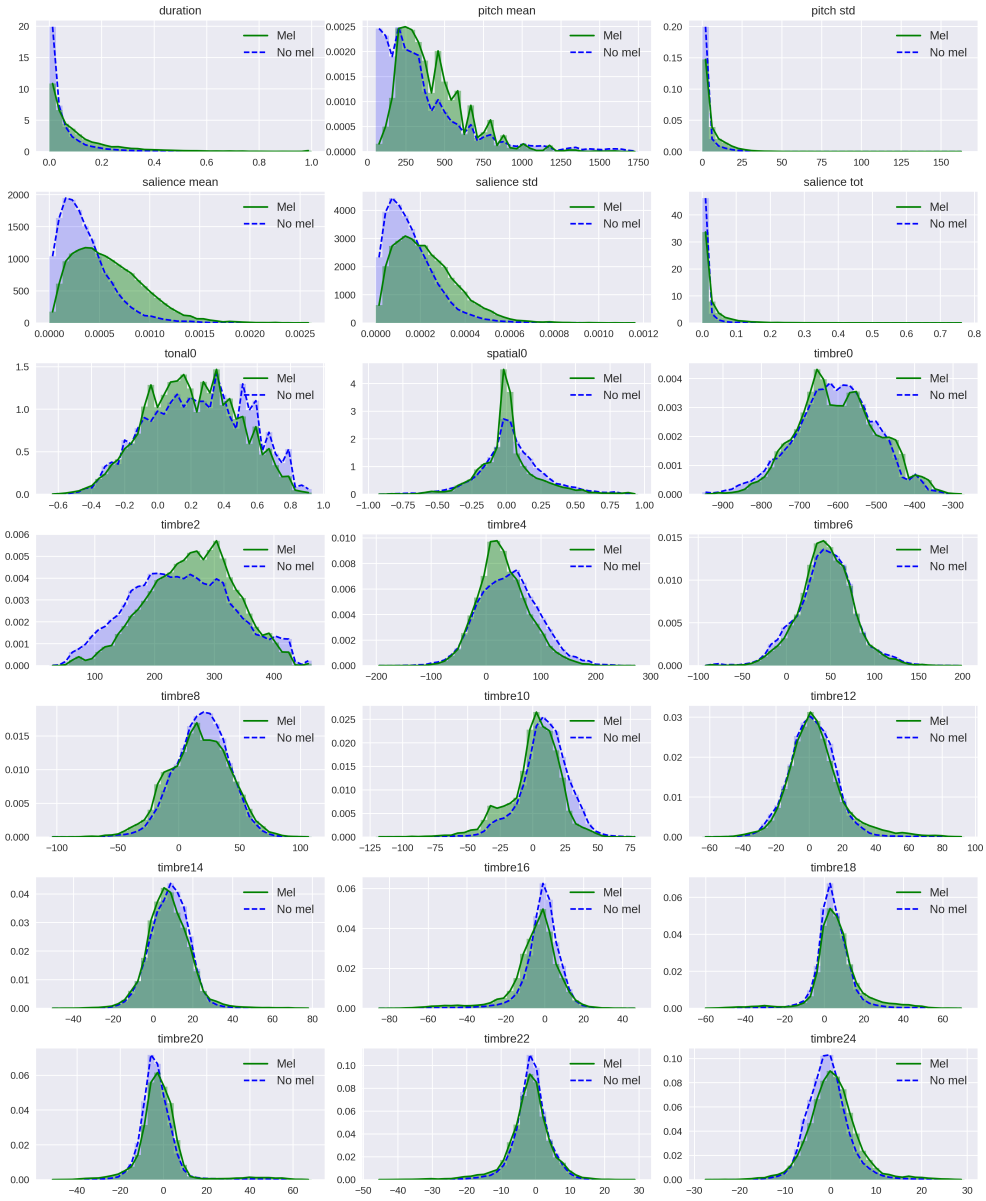

Figure 4.13: Distribution of contour features for melody (green, solid line) and non-melody (blue, dashed line) contours, in MedleyDB with the MEL2 definition.

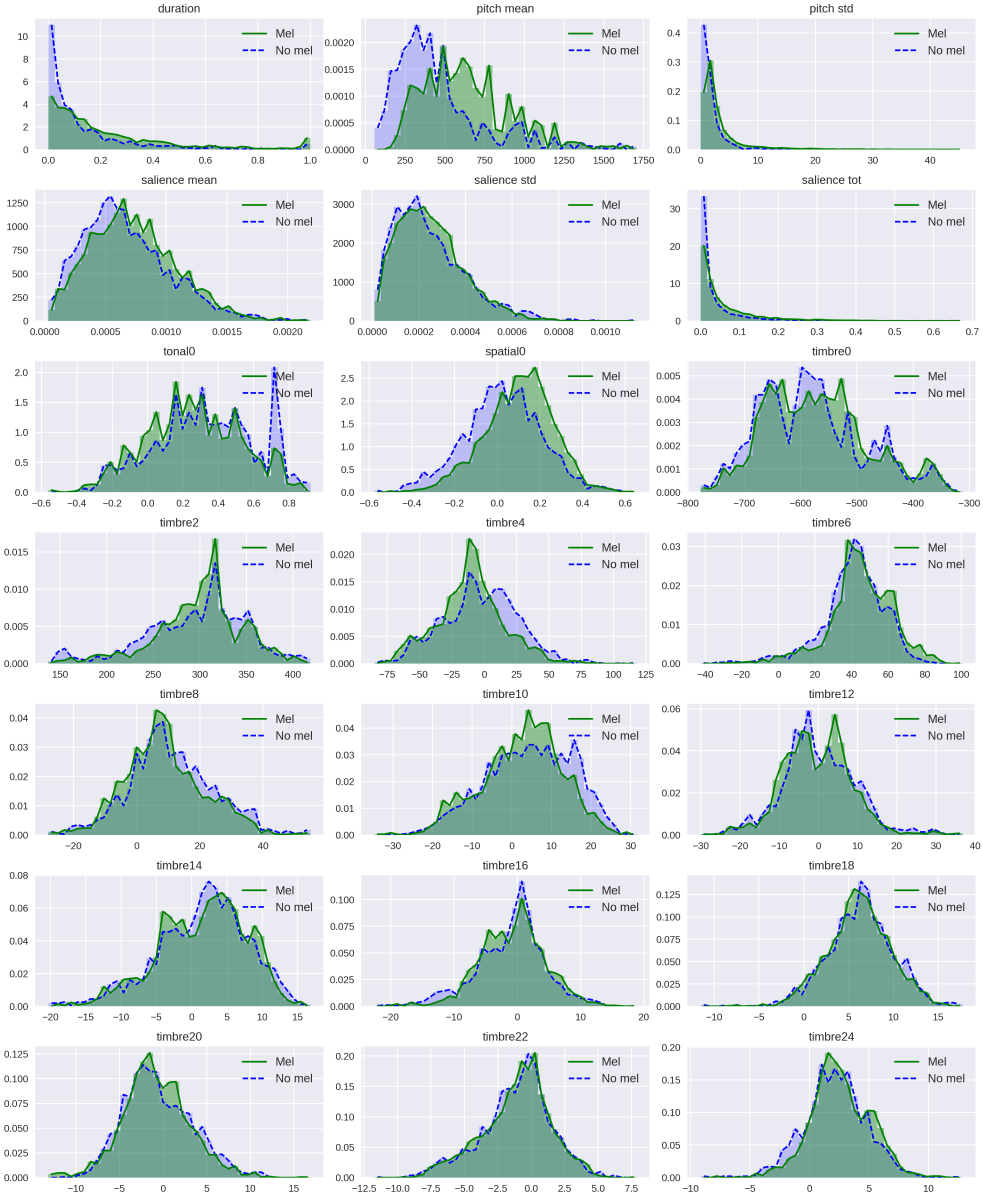

Figure 4.14: Distribution of contour features for melody (green, solid line) and non-melody (blue, dashed line) contours, in Orchset.

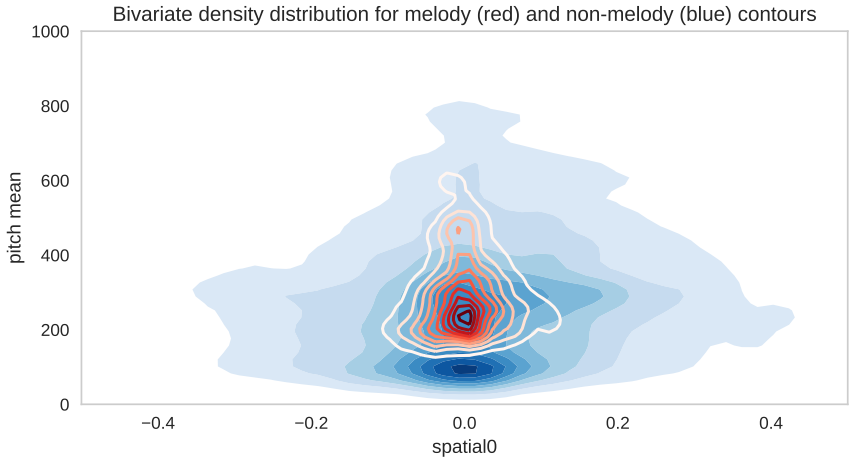

Figure 4.15: Bivariate distribution of the mean contour's pitch and spatial position in MedleyDB with the MEL1 definition for just vocal melodies. Non-melodic: shaded (blue) curves, melodic: unshaded (red) curves.

are normalised per track to remove variance caused by track-level differences (Bittner et al., 2015) as introduced in Section 2.5.4.

With regard to our novel features, we first analyse the distributions of the spatial feature. Figure 4.15 shows the bivariate distributions of the mean pitch and mean spatial location features, for both melodic and non-melodic contours, for vocal excerpts in MedleyDB with the MEL1 definition. We observe that melodic contours tend to have a mean spatial location ("spatial0" feature) which could effectively be exploited for classification: (vocal) melodic contours tend to be more centred than non-melodic contours. This agrees with the knowledge of common music production procedures, as introduced in Section 4.6.2.

Figure 4.16 presents the same distribution for Orchset. In this case, melody contours have a wider range of values for the spatial location in comparison to the vocal contours, but they are centred around a positive value, which corresponds to a location on the left side of the stereo. In this dataset, the melody is commonly played by the first violins (see Section 3.2.1), whose spatial location in the stereo panorama is commonly on the left side, since such recordings tend to be faithful to their position in the orchestral layout (from a listener or conductor perspective). Since non-melodic contours tend to be even more distributed in the stereo, it is also likely that this feature helps the discrimination in Orchset. Melodic contours tend to be more concentrated around the center than non-melodic contours in MedleyDB with the MEL2 (Figure 4.13) definition, but the distribution is wider that in the case of vocal experts with the MEL1 (Figure 4.12). This could be expected, since lead vocals tend to be centered in the stereo panorama while other melodic instruments may be set in other panning positions.

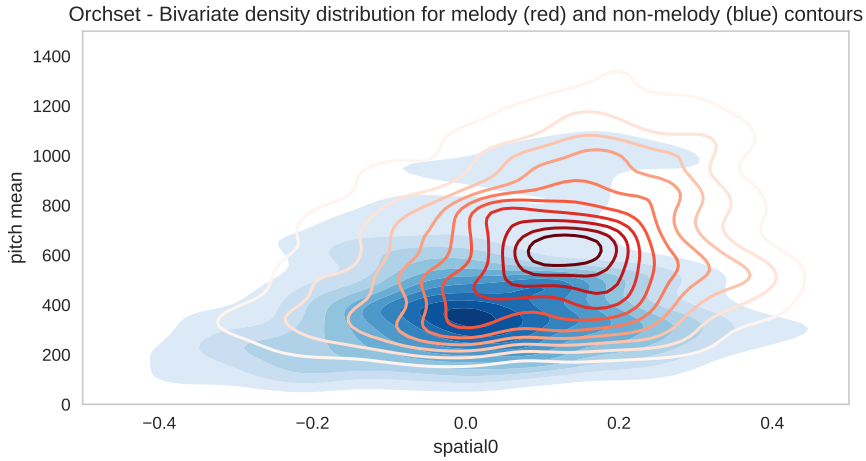

Figure 4.16: Bivariate distribution of the mean contour's pitch and spatial position, in Orchset. Non-melodic: shaded (blue) curves, melodic: unshaded (red) curves.

With regard to the tonal feature, we observe some subtle differences between melodic and non-melodic contour distributions. Melodic contours tend to have a slightly lower value, which indicates that melodic contours "fit" less in the key of the excerpt, since the correlation between chroma profiles is lower. This could be due to the presence of glissandi or passing tones, amongst the melody contours.

Finally, timbre features also seem to have potential for discriminating between melodic and non-melodic contours. As expected, the difference between the distributions is generally higher in the case of MEL1 with only vocal excerpts (Figure 4.12). In this case, timbre features for melodic contours only correspond to the characteristics of vocals. If we do not restrict to vocal melodies, and additionally allow multiple instruments to play the melody in a single excerpt (MEL2), the features for melody contours correspond to the characteristics of multiple instruments. Therefore, the difference between melodic and non-melodic contours (which could correspond to any instrument) is smaller (see Figure 4.12 and 4.13).

4.6.5 Experimental setup

We conduct an evaluation focused on the use of the novel features on both MedleyDB and Orchset. In order to understand their effect on melody extraction approaches, we follow the same data-driven melody tracking approach as in Section 4.5, and compare the results with different feature combinations. Note that these features would also be useful in a heuristics-based approach, but the advantage of using a machine learning classifier is that we do not need to manually create the rules to exploit them.

In order to evaluate the effect of the proposed features, we use the same set of pitch contours as in previous sections (see Section 4.3), and only extend the original set of

Type (number)	Description
Default (6)	Pitch (μ, σ), duration and salience (μ, σ, total)
Tonal (1)	Degree of "fit" in the tonality
Spatial (2)	Panning position of the contour's pitches (median,σ)
Timbre (26)	13 MFCC computed from the filter matrix in contour frames (μ, σ)

Table 4.10: Summary of features computed for each contour

characteristics. Note that spatial features are computed using both audio channels, while the contour formation and characterisation with the rest of features takes place with a mono version (see Section 4.2.3).

We evaluate different feature configurations, starting from the same features as in Section 4.5 (DEF), and additionally including tonal (TON), spatial (SPT) and timbre (TIM) features, as well as all possible combinations between them. The use of all (default and proposed) features is denoted as "ALL". We perform experiments on Orchset, and three different variations on MedleyDB: MEL1 on excerpts with a vocal melody (v-MEL1), MEL1 on all excerpts (all-MEL1), and finally MEL2 on all excerpts (all-MEL2).

We also consider the use of HS as salience function, in order to investigate if the proposed features would also be useful without using a source-filter model. In this case, we do not consider timbre features, since they are computed from the spectral envelope of the filter.

4.6.6 Results

Figure 4.17 shows the Overall Accuracy (*OA*) results for the different feature configurations and experiments. We observe that the use of the proposed features always increases the *OA* when dealing with vocal data and MEL1 definition. If we evaluate on all songs and MEL1 definition, we always improve the results when combining any feature with timbre features, or with timbre features alone. In the case of MEL2, the additional features also help increasing the median *OA*.

The most important increase in overall accuracy is due to the timbre features, which by themselves account for a 5 percentage points increase in the median results on all-MEL2 and all-MEL1. According to a t-test, $\alpha = 0.05$, the increase in *OA* is significant with timbre features, for all-MEL1 and v-MEL1, but the rest of proposed features do not produce a significantly different mean *OA*. Median results are improved in a small amount using tonal or spatial features additionally to the default features. Using two of the additional features always improves the results in comparison to using only the default features in all-MEL2 and all-MEL1, and the best combinations are those in which timbre features are used. Combining all features leads to the best results in v-MEL1, reaching .78 median *OA*. The highest median overall accuracy on all-MEL1 is obtained by the combination of default features with tonal and timbre features,

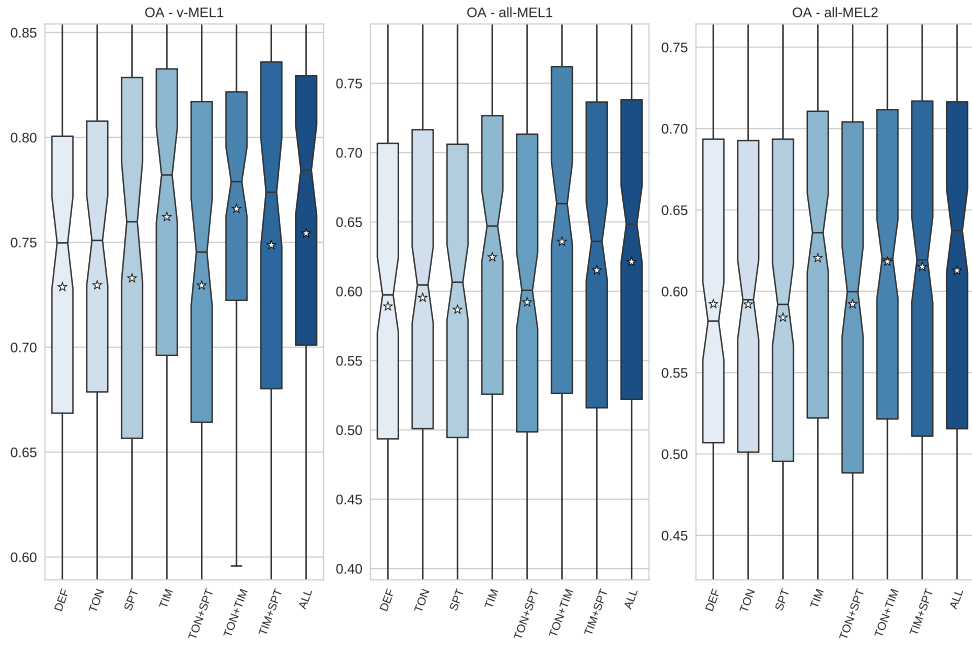

Figure 4.17: Overall Accuracy results in MedleyDB with contours created from CB, for different feature combinations and experiments: MEL1 with only vocal melodies (v-MEL1), MEL1 with all songs (all-MEL1), and MEL2 with all songs (all-MEL2)

achieving 0.66. As observed in figure 4.17, due to the low *OA* values for a few songs, mean values tend to be lower than median values.

Figure 4.19 shows the results for the rest of metrics, which help further understanding the effect of the proposed features on overall accuracy. We observe that the use of timbre features increases voicing recall while significantly ($p = 0.3$) reducing voicing false alarms in MEL1, both contributing to the increase in *OA*. Similar conclusions are obtained with spatial features in the subset of vocal songs, but they have a non-significant effect when considering all excerpts in both MEL1 and MEL2 definitions. The effect of tonal features on voicing detection metrics varies on the experiment, but when they are combined with other features, they always lead to a reduction on false alarms. In the case of MEL2, the largest decrease in *VFA* is produced by the combination of TON+SPT, but it also leads to the highest decrease in *VR*. The highest median *OA* in with the MEL2 definition is obtained by using all features, mainly due to the reduction of *VFA* and only a small reduction of *VR*. All combinations of more than two of the proposed features generally contribute to a decrease in voicing recall in MEL2, but they also contribute to a more drastic decrease of *VFA*, which leads to a higher *OA*.

Note that there is a very high correlation between the effect of using the proposed features on *VR*, and the effect of using them on *RPA* (or *RCA*). As introduced in

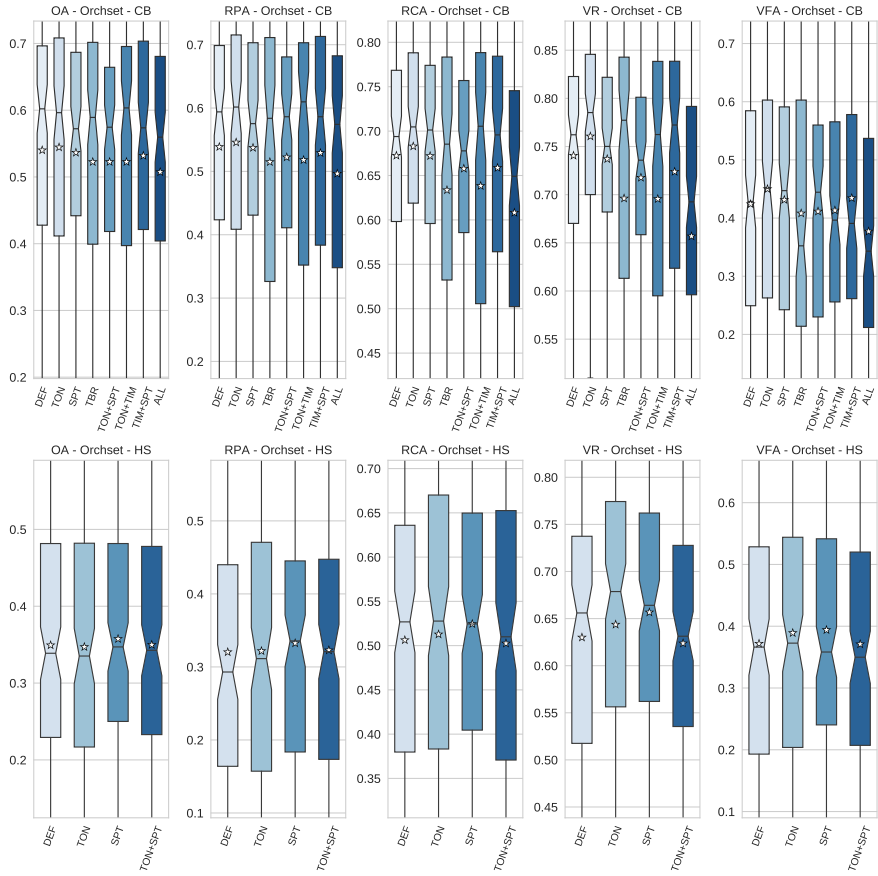

Figure 4.18: Overall Accuracy results in Orchset for contours created with both CB and HS, and different feature combinations

Section 2.7.2, *RPA* computes the proportion of melody frames in the ground truth for which the estimation is considered correct. Since the employed melody decoding method (Bittner et al., 2015) does not provide pitch estimates in unvoiced frames, *RPA* and *RCA* are also affected by *VR*. If voicing recall is decreased, less frames will contribute to the computation of raw pitch accuracy, thus leading to lower values. As also mentioned in Section 4.5, a simple way to increase *RPA* and *RCA* would be to provide negative pitch estimates in the frames where contours are currently filtered (due to a low likelihood of being melodic, as detailed in Section 2.5.4). Note that this would have no effect on voicing related metrics, and would not lead to a decrease in *RPA* or *RCA*, even if all pitch estimates in unvoiced frames are wrong.

Figure 4.20 shows the importances of all features given by the classifier, when we use all the features (ALL configuration) in MedleyDB. Figure 4.21 shows the same information but summing the importance of all feature groups. In this case, we have separated the default features into duration, pitch and salience features. Results on

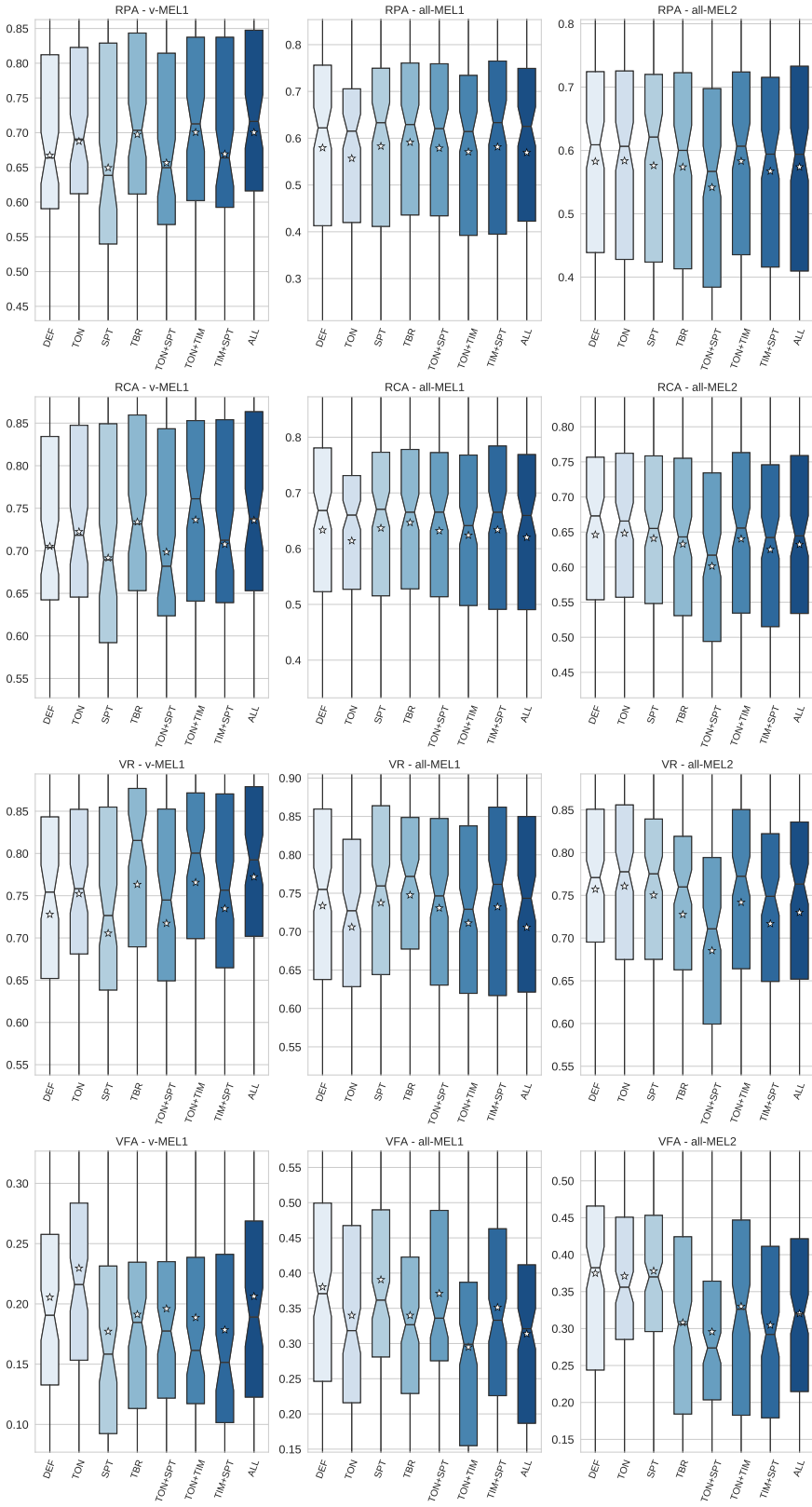

Figure 4.19: Comparison of *RPA*, *RCA*, *VR* and *VFA* results with different feature configurations on MedleyDB and melody definitions: MEL1 with only vocal melodies (v-MEL1),

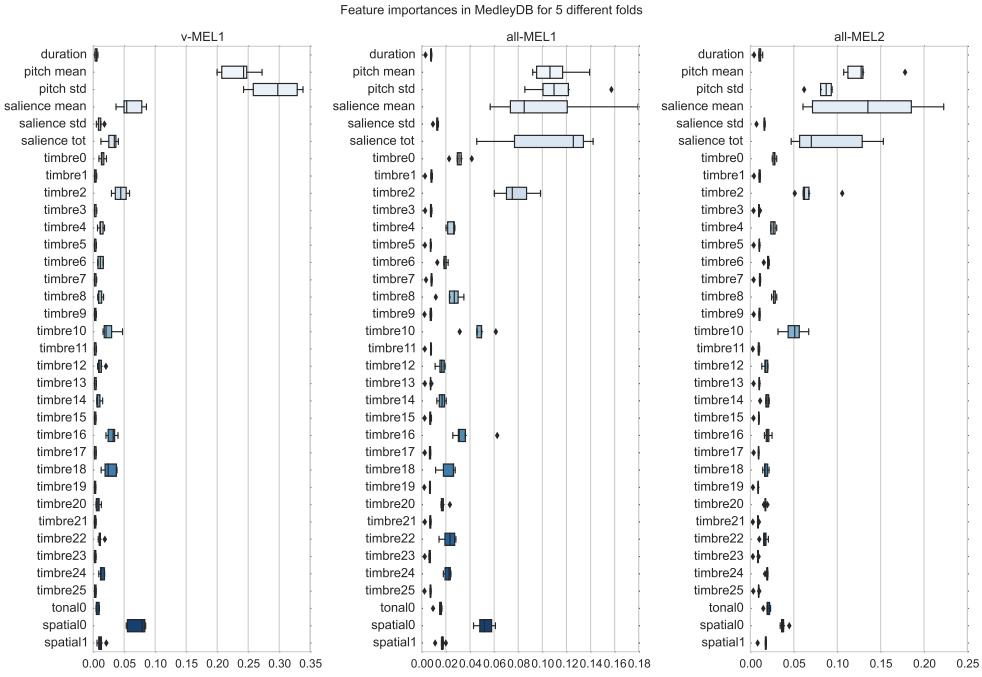

Figure 4.20: Importance of each individual feature for melody contour discrimination in MedleyDB. The boxplots show the results from 5 different folds.

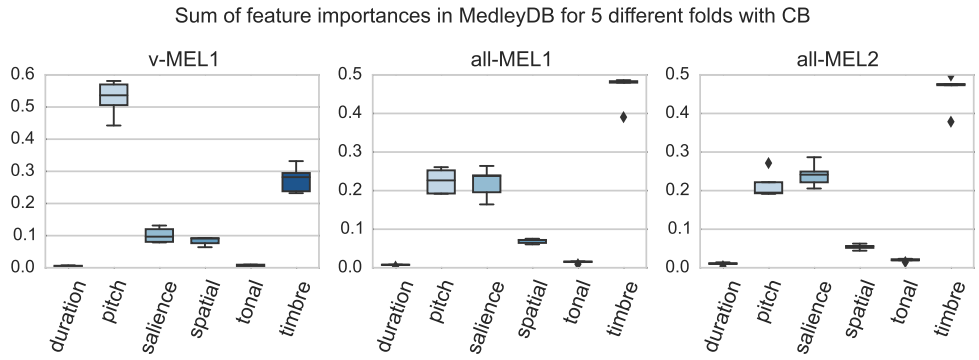

Figure 4.21: Accumulated feature importances in MedleyDB, using CB. Boxplots show results from 5 different folds.

the vocal subset with MEL1 show that the most important features are pitch standard deviation followed by pitch mean. The pitch standard deviation is possibly helping to identify glissandos and vibratos which are commonly found in singing voice. Since melody contours correspond in this case to vocal contours, the mean pitch feature helps distinguishing them from those of other instruments playing notes higher or lower than the singing voice range. Salience mean and spatial0 (which corresponds to spatial mean) are the features with higher relevance after mean pitch. If we consider the aggregated importance in features groups, we see that the whole set of timbre features are actually the second most relevant ones, after pitch features. In the case of all songs and MEL1 definition, salience features gain usefulness over pitch features. Since the classifier can not rely on the voice specific pitch contour features for the discrimination, more importance is given to salience. Spatial features still have some relevance, and timbre features increase their importance as a group, due to the decrease of the usefulness of pitch related features. Finally, in MEL2 we observe that salience features gain more relevance. This definition allows several instruments to play the melody sequentially, and therefore there is a higher variety of pitch related feature values in melodic contours. Isolated timbre features have little relevance on their own, but if we sum the importances of the whole set, the group has the higher total importance. This is probably simply due to the fact that this is by far the largest group (26 features).

In the case of Orchset, Figure 4.18 shows that our novel features do not bring improvements in *OA*, compared to the default features when using CB as salience function. There are several explanations for this. In the first place, melodic contour features in the training set of some random splits may not properly represent melodic contour features in the test set, due to the relatively small size of the dataset (at least, in comparison to MedleyDB). Secondly, some of the feature combinations produce a decrease in recall, which is especially harmful in this dataset since it is mainly voiced. Finally, the additional features do not present large differences in the distributions for melodic and non-melodic contours. Some of the default features are more discriminative on this dataset, when we use CB as salience function: the most relevant is pitch mean, reaching 21% median importance over 10 folds. The next ones are pitch contour duration (6%), some timbre features (maximum 6%), and spatial features (4%). Salience features obtain even lower importance: since they are correlated with the amount of energy of the melody, and orchestral music presents a large dynamic range, melodic contours also have a wide range of salience values. As already seen in figure 4.14, salience related features in Orchset present a very high overlap between the distributions of melodic and non-melodic contours. The best performing configurations in terms of median *OA* are TON and TON+TIM, which also increase the median *VR* in comparison to DEF. Using all available features leads to a decrease of voicing recall, but also to a reduction of the number of voicing false alarms.

As introduced in 4.6.1, our timbre features rely on the estimation of the lead instrument spectral envelope, which is computed together with the H_{f_0} pitch salience func-

tion in the source-filter decomposition (Durrieu et al., 2011). As we have seen, the use of this salience function and our timbre features generally improves melody extraction results. One of the drawbacks is the complexity, as the estimation algorithm requires many computationally heavy iterative steps. For this reason, we also investigate the usefulness of the proposed spatial and tonal features when we use a salience function based on harmonic summation instead of CB (based on H_{f_0}). Figure 4.22 presents the results on MedleyDB, where the DEF results correspond to those by BIT in Section 4.5 (we recall that BIT is based on HS instead of CB, and uses the default features). We observe that the tonal feature improves *OA* in all configurations. Spatial features also increase the median *OA* values in all configurations, but the mean value is not increased in v-MEL1. The combination of tonal, spatial and default features (TON+SPT) also increases median *OA* results in comparison using the default features alone. In the case of Orchset, Figure 4.18 shows that, similarly as in the case of using CB as salience function, none of the feature combinations using the proposed features bring improvements in overall accuracy. In this dataset, the largest increase in melody extraction accuracy in comparison to previous approaches based on pitch contours (BIT or SAL), is due to the use of the proposed salience function (CB) instead of HS.

We also analysed feature importances when using a salience function based on harmonic summation, instead of the proposed CB salience function, on the SPT+TON configuration. Results in MEL2 showed that the classifier gives more importance to salience features when we use CB to create the contours, and relies more on pitch features when using HS. This again shows the benefits of using the proposed salience function, since apart of leading to the creation of a set of contours with a higher melody coverage (see Section 4.3), it also leads to more discriminative salience features.

4.7 Multiple melodic lines estimation

As introduced in Section 2.3, many efforts in MIR have been devoted to melody extraction and multiple fundamental frequency estimation. However, very little research has focused on the estimation of multiple melodic lines, even though it is relevant for applications in which we do not need to estimate all present pitches, but focus on multiple sources playing melodic content simultaneously. Some music examples can be found in symphonies, fugues, certain types of jazz and even popular music.

In this section we propose a method for estimating multiple melodic lines from a music signal based on pitch contour characterisation, using unsupervised methods for pitch salience estimation and a supervised method for melody decoding. The first step is the computation of the combined pitch salience function proposed in Section 4.2 (CB). We then form and characterise pitch contours with the proposed timbre, spatial and tonal features, as well as default features (see Section 4.6). We then propose a supervised method for melody tracking based on pitch contour classification (see

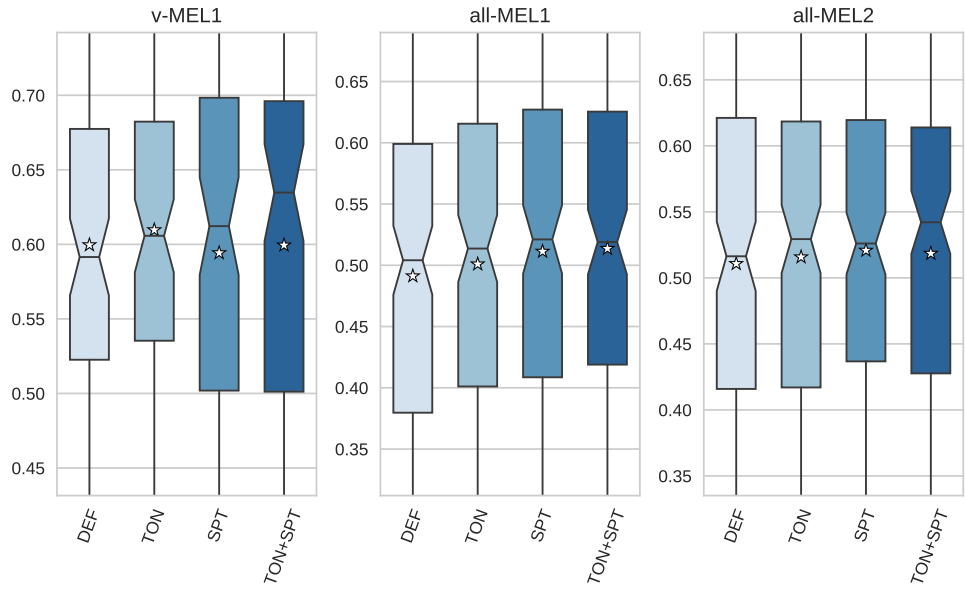

Figure 4.22: Overall Accuracy results in MedleyDB for contours created with HS, for different feature combinations and experiments: MEL1 with only vocal melodies (v-MEL1), MEL1 with all songs (all-MEL1), and MEL2 with all songs (all-MEL2)

Section 2.5.4), together with a novel method for joint decoding of multiple lines (see Figure 4.23). This method additionally exploits knowledge obtained from analysing transitions between contours in a training set.

In comparison to previous approaches (Bittner et al., 2015; Durrieu et al., 2010), we do not just consider pitch and salience when performing decoding, but also the rest of features. Another important difference is that we do not just assume and encourage continuity in time-feature space (e.g time-pitch space), but actually learn if there is such continuity, and model it from training data. Furthermore, we do not only model transitions between melody contours, but also model transitions involving non-melodic contours. This allows our decoding method to encourage transitions which are more likely to be produced among melodic contours. We perform the evaluation on the MedleyDB dataset, which also includes annotations for multiple melodic lines (MEL3 definition).

4.7.1 Contour labelling

Our method for multiple melodic line estimation is based on pitch contour classification, using a Random-Forest Classifier, similarly to the proposed CBC method from Section 4.5, and Bittner et al. (2015). To train it, we first need to label contours in the training set as being melodic or non-melodic, based on the amount of their overlap with the ground truth annotation. Bittner's approach and the proposed method CBC

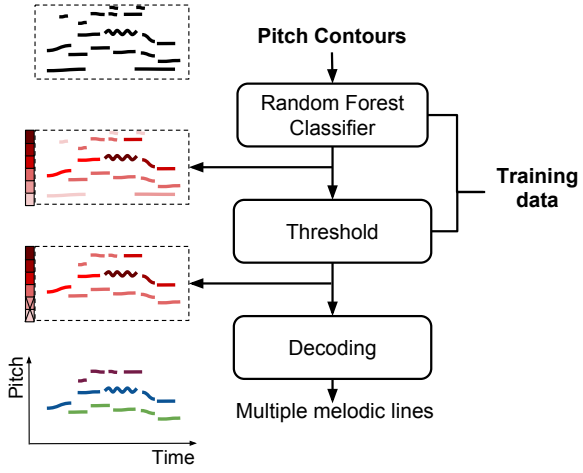

Figure 4.23: Decoding multiple melodic lines based on pitch contour characterisation

compute the overlap as the Overall Accuracy measure when using the contour pitches as estimation and the annotated pitches as ground truth (see Section 2.5.4). In our case, since the ground truth contains multiple annotated pitches per frame, we need a different measure. To compute the overlap, we first measure the amount of frames in which the contour's pitch is within a quarter tone from any of the annotated pitches. The overlap for a given contour (O_{MEL3}) is computed by dividing the previous value by the length (in frames) of the contour. A given contour is considered to be melodic if more than half of its frames overlap with the annotated melody ($O_{MEL3} > \alpha^{th} = 0.5$).

4.7.2 Contour transition modelling

A common assumption made by most melody extraction algorithms when performing melody decoding is the smoothness of the melody in the time-pitch space (Bittner et al., 2015; Salamon & Gómez, 2012; Durrieu et al., 2010). However, similar smoothness could be found in other features as well, which could be exploited for decoding melodic lines. In Section 4.6, we analysed the distribution of several features for both melodic and non-melodic contours. In this section we now analyse the transitions between contours within a melodic line, in terms of the difference between the initial contour features and final contour features.

Since we want to characterise transitions between contours within a single melodic line but our annotations contain multiple lines, we select one from the ground truth as follows. For a given song, we start from the extracted set of contours, and compute the amount of overlap with each of the annotated melodic lines. In this case, the overlap is computed using the Overall Accuracy measure, as the ground truth is a single melody line (for further details see Section 2.5.4). The selected line is the one with the maximum overlap with our set of contours.

After selecting the melodic line, we label contours as melodic if the overlap with the selected melody line is higher than 0.5, or non-melodic otherwise. Note that in this subsection the definition of a melodic contour is thus different from Section 4.7.1 or following sections, since we here use a single melodic line as reference. We then identify pairs of pitch contours which are melodic and partially overlapping in time. The differences between the features for each pair of contours are stored, and we compute them for the whole set of songs in the training set. We also create pairs of overlapping contours in which at least one of them is not melodic, and then compute and store the differences between feature values.

Transitions between melody contours present different distributions in comparison to the rest of transitions, especially in the case of the mean pitch feature: the difference in pitch mean is smaller, which corresponds to the common assumption of smoothness in time-pitch space. There is also a noticeable difference between the two types of transitions in spatial features: transitions between melody contours tend to have a smaller difference in the estimated spatial position. This correlates with the fact that the instrument that plays a melody line tends to keep the same spatial position.

In order to use this data-derived knowledge within a melody extraction framework, we propose to model transitions from training data and use them to encourage transitions between melodic contours, as described in Section 4.7.3. We propose using a normal distribution ($\mathcal{N}(\mu^{\varphi}_{mel}, (\sigma^{\varphi}_{mel})^2)$) to model the differences for each feature (φ) when transitioning between melodic contours, where μ^{φ}_{mel} and σ^{φ}_{mel} represent the mean and standard deviation of the feature differences. We also model the rest of transitions with normal distributions: $\mathcal{N}(\mu^{\varphi}_{rest}, (\sigma^{\varphi}_{rest})^2)$, where μ^{φ}_{rest} and σ^{φ}_{rest} represent the mean and standard deviation of feature differences in the rest of transitions. Another possibility to be explored as future work is to train a classifier to discriminate the type of transitions, similarly to what we do for discriminating melodic from non-melodic contours.

From the training dataset we also compute the probability of changing of contour in subsequent frames, within a melody line. This is computed as the ratio of the amount of transitions between melody contours, to the total number of melody contour frames.

4.7.3 Contour classification and multiple pitch decoding

After assigning a class to each of the contours from the training set (following Section 4.7.1), we train the Random Forest Classifier, following the same procedure as in Bittner et al. (2015) and our classification based melody extraction method (CBC) (see Section 2.5.4). After training the classifier, for each song in the validation and test set we compute the probability of each contour being melodic, similarly to Bittner et al. (2015). Before melody decoding, we also filter out the contours whose melodic likelihood is below a certain value (see Figure 4.23). As introduced in Section 2.5.4, this value corresponds to the threshold which maximises the class weighted F1 meas-

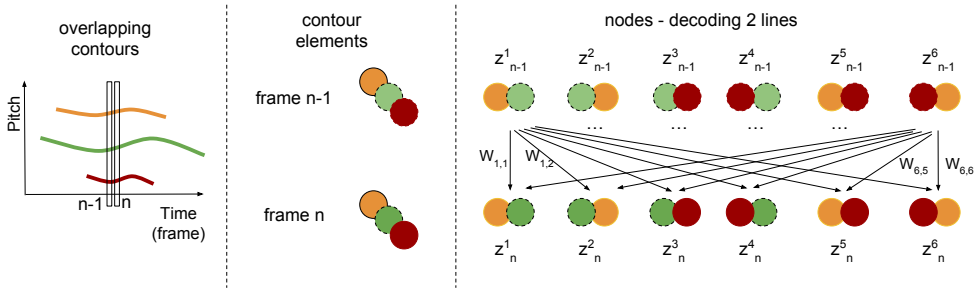

Figure 4.24: Creation of nodes from contour elements, and transition weights $W_{i,j}$ between two nodes in consecutive frames.

ure on the validation set. The remaining contours will then be decoded into (N_L) melodic lines.

First, pitch contours which do not overlap in time with other contours are assigned to the first melody line. Pitch contours with some overlap are then stacked into groups. In each group of contours, we perform a decoding to obtain the most likely sequences of pitches corresponding to the melodic lines. In order to decode multiple lines (N_L), we propose a joint multiple-line decoding within each group, unlike the approach by Bittner et al. (2015), which decodes a single melody line. To do so, we create "nodes", which represent an ordered combination of (N_L) contour elements, and then Viterbi decoding is performed in the node-time space. The number of nodes in a given frame depends on the maximum amount of overlapping contours in that frame (N^c), and the maximum number of lines we want to track. The nodes in a frame represent all possible ordered combinations of N^c pitch candidates in groups of N_L. The total number of nodes in a frame (N^Z) corresponds to the permutations $P(N^c, N_L) = N^Z = N^c!/(N^c - N_L)!$. E.g. for a frame with 4 concurrent contours ($N^c = 4$), if we decode 2 lines, the number of nodes is: $N^Z = 4!/(4-2)! = 12$.

In order to perform the decoding, we need to compute the likelihoods of the nodes, the prior probabilities, and the transition matrices between nodes. The likelihood of each node is computed as the product of the likelihoods associated to each of the elements forming the node, which are equal to the estimated probability of their containing contour being melodic. The prior probability is also the product of the prior probabilities associated to each of the pitches, which is set uniformly.

To create the node transition matrix in a given frame we compute transition weights ($W_{i,j}$) between each of the nodes to the nodes of the previous frame. The size of this matrix thus changes depending on the amount of nodes in each frame. Figure 4.24 shows the nodes when decoding $N_L = 2$, and there are three overlapping contours.

To compute the transition weight between a node z^k_{n-1} and node z^l_n from frames $n-1$ and n respectively, we compute the product of the transition weights of the elements from the nodes. Transition weights between individual node elements cor-

respond to the product of two different weights. The first weight is associated to the (log)difference between both elements pitches. The second weight is associated to the transition weights between the contours that contain the elements, and these are related to (all) contour features. Both weights are computed using the transition models introduced in Section 4.7.2.

We compute a transition matrix for each feature, with a weight for the transition from contour i to contour j computed as:

$$W_{i,j}^{\varphi} = \frac{\mathcal{N}(\Delta_{i,j}^{\varphi} | \mu_{mel}^{\varphi}, (\sigma_{mel}^{\varphi})^2)}{\mathcal{N}(\Delta_{i,j}^{\varphi} | \mu_{rest}^{\varphi}, (\sigma_{rest}^{\varphi})^2)} \tag{4.16}$$

where $\Delta_{i,j}^{\varphi}$ corresponds to the difference in the feature φ, and where $W_{i,j}^{\varphi}$ corresponds to the transition weight for a given feature, both between contours i and j. Note that such weight is equal to one if the both distributions are the same ($\mu_{mel}^{\varphi} = \mu_{rest}^{\varphi}, \sigma_{mel}^{\varphi} = \sigma_{rest}^{\varphi}$). The weight is higher than one if the feature value difference $\Delta_{i,j}^{\varphi}$ is more likely to correspond to a transition between melodic contours, or lower than one if it is more likely to correspond to other transitions.

The weights of the global transition matrix are computed as the product of the weights due to all features:

$$W_{i,j} = \prod_{\varphi \in \Psi} W_{i,j}^{\varphi} \tag{4.17}$$

where Ψ corresponds to the set of all features.

Additionally, we also consider a weight equal to the probability of changing between contours, which is learnt from the training data. In short, the transition matrix favours transitions which are more likely to occur between melodic contours, using information about multiple features obtained from training data.

After computing all likelihoods and normalising the transition matrices between nodes, we compute the cumulated likelihood. Backtracking finds the most likely path in the node-frame space, which is then transformed into a total of N_L melodic lines (f_0 sequences) ordered by total cumulated likelihood. Note that in practice the computations take place in the log scale, in order to avoid rounding errors due to the very small values in linear scale.

4.7.4 Experimental setup

We evaluate the proposed method using MedleyDB with the MEL3 definition, since it contains annotations of multiple melodic lines. We consider two different configurations of the proposed method: the first (CBM) with the same contour creation parameters as in Section 4.3, which was also used in the following sections ($\tau_\sigma = 0.9$,

$\tau_+ = 0.9$). Additionally, we evaluate another configuration (+CBM), with more lenient peak filtering (prior to contour formation), using the following parameter values: $\tau_\sigma = 1.6$, $\tau_+ = 0.5$). We recall that the decrease in τ_+ allows filtering out less peaks in each frame, and the increase in τ_σ allows a higher difference in salience below the salience mean. Therefore, both of them contribute to the creation of a larger amount of contours. This is useful to increase the recall, which is especially convenient with the MEL3 definition, since it allows the presence of multiple melody pitches at the same instant.

We compare the performance of the proposed method with other three methods. The first method (HSM) is a variation of CMB, which uses harmonic summation instead of the proposed combined salience function. For CBM and HSM, we evaluate the results obtained with several values of N_L (maximum number of lines to be decoded). We denote each variation with the number of lines at the end of the name abbreviation, e.g. CBMN3 denotes the use of CBM with $N_L = 3$. The second method with which we compare is the multipitch approach by Duan et al. (2010) (MP-DUA) and the third is Benetos & Weyde (2014) MIREX 2014 submission (MP-BEN14), but slightly modified by its original author to output values on a 20 cent grid, instead of semitone-quantized pitch values. Note that MP-BEN14 is based on the use of instrument-specific spectral templates, which have not been adapted to the instruments present in MedleyDB (for instance there are no templates for singing voice).

The evaluation is conducted on five train/test splits using an "artist-conditional" random partition on the 108 songs that include melody annotations on MedleyDB. We use the same distribution of songs among training, validation and testing as in Section 4.5, with roughly 63%, 12%, and 25% each.

4.7.5 Results

Figure 4.25 shows the precision (Prec), recall (Rec), and accuracy (Acc) of the evaluated methods, computed from the joint results on the testing set in five different random splits.

The proposed method (CBM) obtains the highest accuracy amongst the evaluated methods. HSM ranks second, and then MP-BEN and MP-DUA achieve the lowest accuracy results. Note that MP-BEN and MP-DUA are multipitch estimation methods, and therefore they aim at estimating all present pitches in a musical audio signal. It was therefore expected that they would not obtain high accuracies in our dataset, since MEL3 only contains annotations of the pitches produced by melodic instruments, and the precision of multipitch methods would be low. However, we would expect these methods to obtain a higher recall than any other methods. Results show that CBM obtains a lower recall than MP-DUA, and very similar recall in comparison to MP-BEN. However, the configuration +CBM (which creates a higher amount of contours) achieves a higher recall than both MP-BEN and MP-DUA. More importantly, this configuration also obtains a much higher precision in comparison to

MP-BEN and MP-DUA.

As expected, CBM obtains a higher precision than +CBM, particularly when $N_L = 1$. As also expected, increasing the amount of decoded lines decreases the precision and increases the recall in CBM, +CBM and HSM methods. Note that the increase of recall when we increase the number of decoded lines (from $N_L = 1$ to $N_L = 3$) is smaller in CBM than in the case of +CBM. The number of created contours which overlap in time is smaller in CBM, and therefore increasing the amount of decoded lines has less effect in the recall, since we actually reach the maximum amount of concurrent pitches. All evaluated variants of CBM obtain similar accuracy, with a mean value around 0.4, and a slightly lower median value.

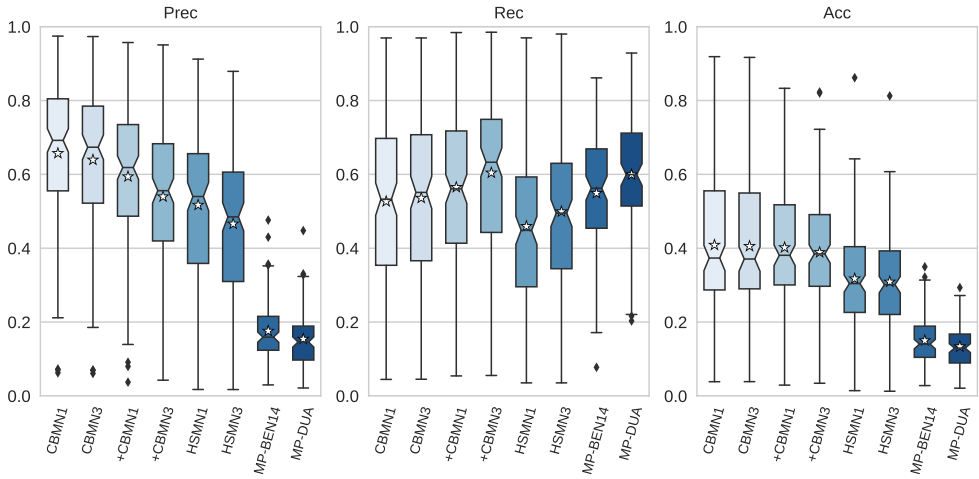

Figure 4.25: Precision, Recall and Accuracy results for MEL3 definition

We further study the kind of errors produced by the algorithms, by analysing Figure 4.26. First, the lowest total error (E_{tot}) is obtained by CBM, followed by +CBM and HSM. Much higher error values are obtained by MP-BEN, and MP-DUA. E_{subs} measures the substitution errors: the number of ground-truth f_0 values for each frame that were not estimated, but other incorrect f_0 values were returned instead. As also expected, the highest values are obtained in this case by MP-BEN and then MP-DUA. The lowest E_{subs} is obtained by CBM, specifically CBMN3. E_{miss} measures the number of missed errors: ground-truth f_0 values that were missed by the algorithm, but no other f_0 estimates were returned. Lowest values for this kind of error are obtained by MP-DUA, MP-BEN, and then +CBM. The reason why MP-DUA and MP-BEN obtain such low values is because if they do not output an annotated f_0 value at a given frame, they usually give some other f_0 estimate as output, which counts as a substitution error. Finally E_{fa} measures the false alarms: the extra f_0 estimates that are not substitutes. As also expected, MP-DUA and MP-BEN obtain the highest number of errors, since they are multipitch algorithms and therefore estimate more pitches than the melodic. Also as expected, +CBM obtains the highest errors amongst

the proposed methods, especially when decoding a maximum of 3 lines.

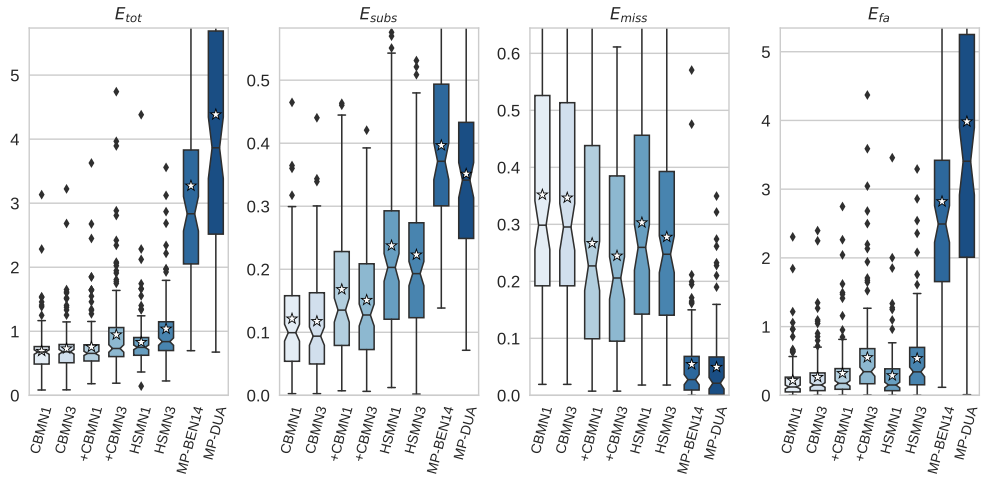

Figure 4.26: Error Results in MedleyDB for the MEL3 definition (multiple line estimation)

Finally, Figure 4.27 shows the chroma accuracy of the methods (ignoring octave errors). Note that the difference between the normal and the chroma measures are due to octave errors. The best chroma accuracy is still obtained by +CBMN1, which increases around 2 percentage points in comparison to the accuracy measure, meaning there are few octave errors. We observe that MP-DUA and MP-BEN present a higher number of octave errors than our proposed methods, since the difference between Ch-Rec and Rec is much larger (19 and 4 percentage points in the case of MP-DUA and +CBMN3 respectively). Total error rankings when we ignore octave errors ($Ch - E_{tot}$) are similar to E_{tot}, and the lowest error is also obtained by CBM.

4.8 Conclusions

In this chapter we have analysed the benefits of exploiting knowledge derived from data for melody extraction. We have presented and evaluated a set of melody extraction methods which integrate a source-filter model within a pitch contour based melody extraction framework, increasingly exploiting available data for melody decoding. The evaluation is conducted on both vocal and instrumental music, coming from two different datasets: MedleyDB and Orchset. These datasets cover a wide range of genres: pop, rock, jazz, opera, and symphonic music, as well as different melody definitions.

At the beginning of this chapter, we hypothesised that the combination of source-filter models and pitch-contour-based melody tracking would lead to improvements in melody extraction accuracy. To investigate this, we first adapted a pitch salience function based on a source-filter model for the formation of pitch contours. Pitch

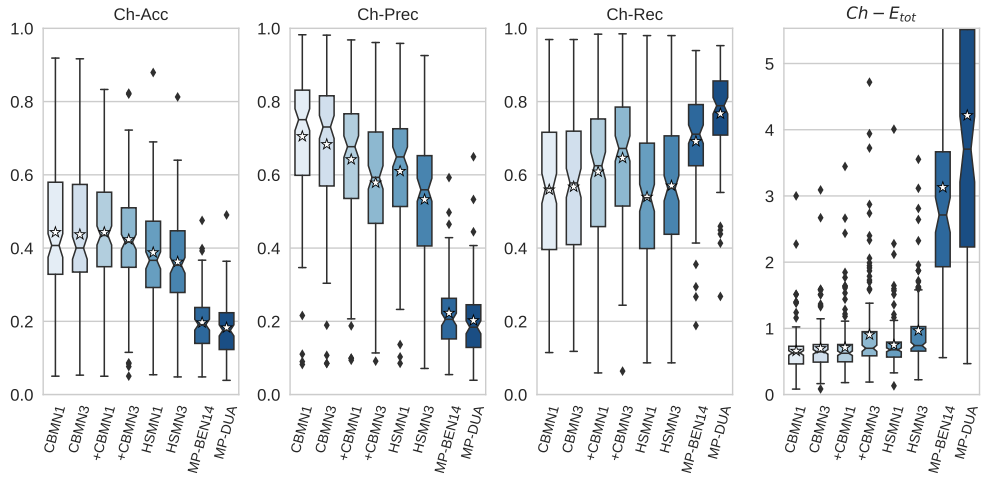

Figure 4.27: Results in MedleyDB for chroma related metrics, with the MEL3 definition

contour formation is based on finding peaks from the pitch salience function, and grouping the most salient ones, using time and pitch continuity principles. The underlying assumption is that salience peaks corresponding to the melody are more salient than non-melody peaks, and that there is a relatively small deviation between their saliences. However, the pitch salience function obtained with a source-filter model presents a large range of values. To adapt this salience function for pitch contour formation, we first propose its combination (CB) with a salience function based on harmonic summation (HS). Our second proposal is based on weighting the salience function with an estimate of the energy of the melody source. The evaluation showed that the proposed methods are able to increase the predominance of the melody over the rest of pitches, in comparison to other pitch salience functions. One of the contributions of this chapter is the proposal of novel metrics for pitch salience evaluation, which consider not just pitch estimation in voiced frames, but also the behaviour in unvoiced frames.

The proposed salience functions were then used within melody extraction methods based on pitch contour selection (using heuristic rules). Results showed that the combined salience functions lead to improvements in both overall accuracy and pitch estimation accuracy, and it also reduces octave errors in comparison to using harmonic summation.

We also studied the effect of several parameters for pitch salience estimation using a source-filter model, as well as parameters for pitch contour creation and pitch contour selection. Pitch contour creation parameters were found to influence melody extraction results more than those used for pitch salience estimation. The voicing threshold used for the final pitch contour selection also had a very important role in overall melody extraction accuracy: the higher the voicing parameter, the higher amount of voiced estimations. This is adequate for the Orchset dataset which is mainly voiced,

but harmful for MedleyDB especially with the MEL1 definition, due to the large amount of unvoiced portions. We also evaluated this melody extraction method in the context of source separation, using a subset of songs from MedleyDB, which only have one melodic source. To achieve this, we gave as input to Durrieu's separation algorithm (Durrieu et al., 2010) the output of the proposed melody extraction method, and compared it against using its own melody estimate. Results showed an increase of separation quality, measured by the *Source to Distortion Ratio*.

We then proposed the use of CB for melody extraction using pitch contour classification. Evaluation results also showed that CB outperforms harmonic summation in this case. The main benefits of using pitch contour classification was found when evaluating the results on the symphonic music dataset: it is able to adapt to the characteristics of this dataset, which is mainly voiced. Also, the rules used in the heuristic method are tailored to other kinds of music, and may not be applicable to the symphonic music context, as detailed in Chapter 3. Another advantage of classification based methods is that it is much more straightforward to include additional features, since we do not need to manually create rules to consider them. We thus selected a melody decoding framework based on pitch contour classification to analyse the effect of using an additional set of features. We propose the characterisation of pitch contours using timbre, spatial and tonal features, additionally to the previously used features (related to contour length, pitch and salience). Results showed that combining the proposed features with the previously used features (related to contour pitch, salience and duration) generally reduces voicing false alarms in both datasets, and improves overall accuracy in MedleyDB with all melody definitions. The highest increase in overall accuracy is due to timbre features, thanks to a better voicing detection. However, spatial features are also helpful when identifying vocal melodies, and in combination with timbre and tonal features (especially with the MEL1 definition in MedleyDB).

Finally, we proposed a joint multiple-line decoding method also based on pitch contours as an intermediate pitch representation. We exploit the previously introduced features for contour classification, and uses transition models to favour transitions between melodic contours. After filtering out contours with a low melody likelihood, the sequence of melody pitches is decoded from the remaining contours. Multiple pitches are tracked using Viterbi decoding on a time-node space, where a node represents an ordered combinations of pitch candidates. The evaluation is conducted on MedleyDB with the MEL3 definition, which allows multiple simultaneous melodic lines. Results show a much higher accuracy and precision than two state-of-the-art multipitch estimation methods, while achieving similar recall. Better results are also obtained in this case when using CB as pitch salience function instead of HS.

We have thus seen that supervised and unsupervised learning methods, as well as the proposed features, allow improving the state-of-the-art in melody extraction, which validates our second and third main hypotheses (presented in Section 1.3). An interesting future research direction is to substitute the unsupervised pitch salience estimation by a supervised method such as Deep Neural Networks (DNN). A recently

proposed approach by Rigaud & Radenen (2016) in which the pitch salience is estimated by a DNN shows promising results, even though it is only evaluated on vocal music. We foresee the use of Neural Networks for pitch salience estimation on a wider range of music data, and propose its integration within a melody extraction framework based on pitch contour characterisation. One of the challenges is to deal with the relatively small size of the datasets in comparison to other disciplines, and the cost of creating annotated data. It would also be interesting to automatically learn contour creation parameters from data, since they have an important effect on the amount and shape of the created contours, and thus on melody extraction accuracy.

Chapter 5
Conclusions

5.1 Introduction

In the introduction of this dissertation we presented our main research question: *"can melody extraction algorithms benefit from modelling the context of the data to be analysed?"*. From this research question we derived three hypotheses, which we have investigated in Chapter 3 and Chapter 4.

One hypothesis was that melody extraction algorithms are generally focused on simple vocal data and may not generalise well to other, more complex musical contexts. To validate it, in Chapter 3 we evaluated state-of-the-art pitch estimation methods on a novel symphonic music dataset (Orchset) which presented different characteristics than those of standard melody extraction datasets. First of all, the definition of melody used in this dataset is less restrictive than in previously ones. Second, the melody is not played by a single instrument but by multiple instruments or orchestra sections in unison. In addition, instrumental sections can alternate in playing the melody. Third, the dataset is musically and acoustically complex, including a higher spectral density and more frequent overlap between sources. Evaluation results of a selection of state-of-the-art pitch estimation algorithms revealed that symphonic music is a very challenging material for melody extraction, and most methods present lower accuracies in this material than in vocal music, which has traditionally been the focus. We also presented an analysis of agreement when estimating the melody, and studied the correlation of both pitch estimation accuracy and mutual agreement, with musical characteristics from the annotated melodies. Note density presented the largest (negative) correlation with melody extraction accuracy. From this evaluation, we first concluded that the most common approach for pitch salience estimation is not appropriate for this kind of data. In addition, some approaches perform a suboptimal melody pitch tracking, which further affects melody extraction accuracy on symphonic music signals. These results validate our hypothesis, and as a consequence of this evaluation, we found out that the most accurate melody extraction method in this dataset employs a source-filter model for pitch salience computation. This method models the char-

acteristics of the music signals under analysis in an unsupervised fashion, implicitly learning the spectral shape of the lead instrument.

Another hypothesis was that supervised and unsupervised learning from data would allow advancing the state-of-the-art in melody extraction. To validate this hypothesis, in Chapter 4 we proposed a set of data-driven methods which build upon previous state-of-the-art approaches (Durrieu et al., 2010; Salamon & Gómez, 2012; Bittner et al., 2015). Their evaluation showed improvements on melody extraction overall accuracy, for a wide range of music material. Our first contribution was to incorporate a source-filter model into a melody extraction framework based on pitch contour selection by means of heuristic rules. Our method benefits from unsupervised learning at salience function creation, and provides substantial improvements in comparison to the original method, based on harmonic summation (Salamon & Gómez, 2012) on both MedleyDB and Orchset. However, a drawback of this approach is that it is not able to learn the characteristics of melodic contours from a set of training data, and that it is complex to add new features since we should manually create the rules for melody tracking. Our second contribution consists in the combination of the proposed salience function and a melody tracking method based on pitch contour classification, by means of a Random Forest Classifier. This method achieves substantial improvements over an alternative approach based on harmonic summation (Bittner et al., 2015). In comparison to our first contribution, this method was better able to adapt to the characteristics of any given dataset, since it did not employ any fixed heuristic rule for melody tracking. This was especially beneficial on symphonic music, which is substantially different from the data for which the heuristic rules are tailored. Another positive aspect of a classification-based approach is that it is much more straightforward to include additional features, since we do not need to manually create rules to consider them.

The remaining hypothesis was that features related to timbre, tonality, and spatial information would be useful for improving melody extraction algorithms. In Chapter 4 we proposed the use of such information to create novel features for the characterisation of pitch contours, in order to complement the (duration, pitch and salience-related) features previously proposed by Salamon & Gómez (2012). Our features led to the decrease of voicing false alarms, and generally improved overall melody extraction accuracy (especially on MedleyDB) when used within a classification-based melody tracking approach. Finally, we extended melody extraction methods to estimate several melodic lines by means of joint multiple-line decoding. This approach was based on the combined salience function, the proposed set of features, and pitch contour classification. Melodic lines were decoded using the Viterbi algorithm on a time-node space, where a node represents an ordered combination of candidate pitches. Data-driven pitch contour transition models were used to favour transitions which are more likely to involve melodic contours, and discourage transitions which were likely to involve non-melody contours. This approach obtains higher accuracies than state-of-the-art multiple pitch estimation methods when evaluated on MedleyDB MEL3

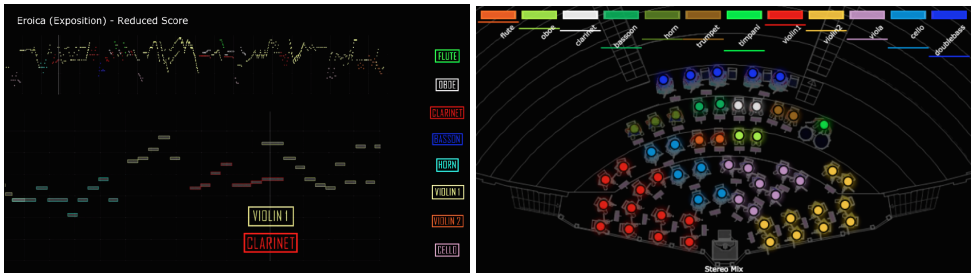

Figure 5.1: Screenshots of the reduced piano roll representation (left) and orchestral layout visualisation (right)

definition, which contains the annotation of all melodic instruments.

In the following sections we present a set of prototype applications where the work from this thesis has been integrated (Section 5.2), and we review the most relevant contributions of this dissertation (Section 5.3), including the symphonic music dataset and source code of the proposed methods [47]. Finally, we provide a description of future work directions (Section 5.4).

5.2 Prototype applications

In the context of the PHENICX project (Gómez et al., 2013), we have provided listeners with visualisations of musical information, in order to create novel orchestral classical concert experiences. One of the main motivations is that symphonic music embraces a wealth of musical information, which may not be easily perceived or understood by general audiences. In symphonic music, it is common that musical scores are available in a digital form, but they are too complex to understand for people without formal musical training. In this project, we have used piano roll representations in order to simplify musical notation, only considering pitch (y-axis), time (x-axis) and instrument related information. It is however still complicated to understand piano rolls with complete symphonic music scores, due to the large number of instruments and the considerable overlap between voices, since this results in overloaded images. For this reason we proposed a simplified version of the complete piano roll, showing only the main melodic line (Martorell et al., 2015), as shown in Figure 5.1 (left). We also used an orchestral layout visualisation, with different colours per instrumental section, and different intensity depending on the individual dynamics. Additionally to the instrumentation, physical layout and dynamics information, it also depicts simplified pitch information for each instrumental section, as shown in Figure 5.1 right.

[47] http://www.mtg.upf.edu/node/3737

This dissertation has contributed with the automatic creation of a "simplified musical score", which is similar to the previously mentioned reduced piano roll, displaying the melody, as well as information about the instrument(s) playing it. The difference with a piano roll is that we represent f_0 estimations, and therefore fine-grained pitch information such as vibratos or glissandi can be visualised.

Assuming that music scores can be aligned to the audio either manually or automatically (Dixon & Widmer, 2005; Niedermayer & Widmer, 2010; Miron et al., 2015; Carabias-Orti et al., 2015; Miron et al., 2016), it is possible to use this additional information to identify the melody and the instrument(s) (sections) playing it with more accuracy. Two prototypes have been developed, which propose the visualisation of melodic information in the context of symphonic music, using the melody extraction method presented in Section 4.4. They additionally take advantage of an aligned music score to refine the estimation, and in the case of *meloVizz* (Section 5.2.2) also to compute the probability of each of the instruments to be playing the melody. The melody is extracted without using the score, but a mask is applied to the estimation, deleting any melody estimates which are not present in the time-frequency positions of the notes present in the score.

5.2.1 PHENICX prototype

As a main outcome of the PHENICX project[48,49], we developed a prototype which integrates different technology for extending and enriching classical music concerts[50]. In particular, the prototypes integrates functionalities to be used before, during and after the concert event. Before the event, the user can, for instance, get program notes in advance or prepare with an earlier recording. During the event, the user can access to curated text guides, follow the score, tag or share their preferred moments. After the concert, the user can access all concert recordings and material, providing audio and video orchestra focus (incorporating audio sound source separation technologies), personal tags and comments, instant video sharing, comparison of different recordings or structure insights.

One of the features of the prototype was the display of a synchronized music score, as well as a "simplified version" using the automatically detected predominant melody of a piece. The application allows the visualisation of the main melody of line while watching and listening to the recorded concert, as shown in Figure 5.2.

[48] Academic website: http://phenicx.upf.edu/
[49] Product website: http://phenicx.com/
[50] Prototype: http://phenicx.prototype.videodock.com/

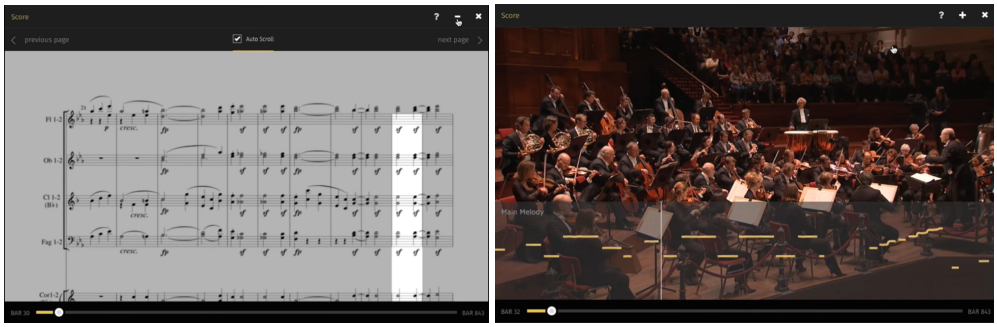

Figure 5.2: Screenshots of the PHENICX prototype, showing time aligned information from the first movement of Beethoven Symphony 3 (Eroica). On the left side, score from the woodwind section (bars 21 to 31). On the right side, estimated melody (bars 28 to 38) over a video recording of the orchestra.

5.2.2 Melody visualisation (*meloVizz*)

In the context of the PHENICX project, we also developed *MeloVizz*[51], a web-based prototype for the visualization of melodic information as well as an estimation of the instrument/s playing the melody.

The probability (P_n^i) that an instruments i is playing the melody in a frame n is estimated with a very basic method, to be improved as future work. First, we perform a frame-wise comparison of the estimated melody pitches against the instrument pitches derived from the score. If they lie within a quarter-tone range, we set the value of A_n^i to one, otherwise to 0. The estimation of the probability P_n^i is computed by Gaussian filtering A_n^i along the time axis for smoothing.

The tool allows playing the analysed musical piece, and following the estimated melody in a piano-roll canvas (see Figure 5.3). A scrolling curve shows pitch values (y-axis), while time is represented horizontally (x-axis). A different colour is used for each instrument (or section), allowing the user to easily visualise which instrument is predominant at each time. Additionally, the intensity of each colour is variable, and is mapped to the estimated probability of the instrument (P_n^i). A vertical line refers to the current playing time, and pitches estimated in a short time window around it are displayed, both in the past and future. Variable size text labels are additionally displayed at the top, showing the instrument/s that contribute to the melody. The name of the instrument considered predominant is displayed in its corresponding colour.

Figure 5.3 (left) shows estimated melody pitches in the first movement of Beethoven's Eroica. In this example, we observe the vibrato from the flute at current playing time, which is the only instrument contributing to the melody. Figure 5.3 (right) shows that both clarinet and violin contribute to the estimated melody, but the violin is considered predominant.

[51] http://repovizz.upf.edu/phenicx/melovizz

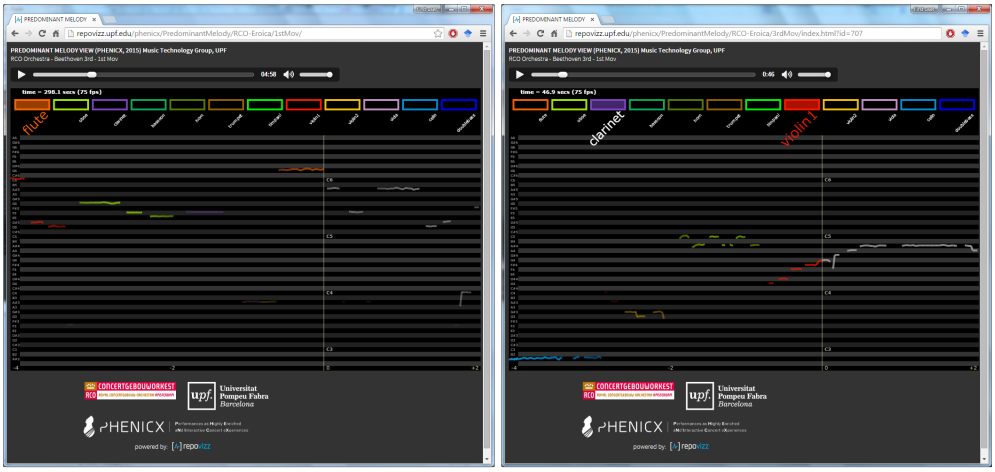

Figure 5.3: Screenshots of *meloVizz*. The images correspond to estimations from the first (left) and second (right) movements of Beethoven's 3rd Symphony (Eroica), recorded by the Royal Concertgebouw Orchestra.

The prototype is implemented in HTML5 to be run in any modern browser. It gathers data through AJAX calls from the online repovizz repository (Mayor et al., 2013) using its RESTful API. In order to visualise a given piece, data files with estimated melody pitches and instrument probabilities need to be previously computed, and uploaded into a repovizz datapack.

5.3 Summary of contributions

5.3.1 Extending the scope of melody extraction

Until very recently, most research in the field had focused on a restrictive melody definition, allowing only a single source to be playing the melody in the whole piece. Furthermore, most research has in practice focused on vocal melody extraction. This thesis is an effort to consider a more varied, complex and realistic set of data in melody extraction research. On the one hand, we have created a dataset which focuses on symphonic music, which had never been considered in the literature. On the other hand, we have used MedleyDB as a source of realistic musical data, since it comprises many complete songs on a varied genre set, including both vocal and instrumental melodies. Furthermore, we consider several definitions of melody, which could be useful for different kinds of applications.

5.3.2 Annotation process and analysis of agreement

In Section 3.2 we proposed a methodology for dataset creation and annotation, in the context of symphonic music. Given the difficulty of annotating the melody within such a musical context, we proposed a methodology based on asking subjects to sing

or hum along the music. After a manual agreement analysis, we annotated the notes that the participants considered as melody, considering both their singing recordings, the musical content of the audio excerpts, as well as the degree of confidence with their singing.

In Section 3.3 we also conducted an automated analysis of agreement, which included the study of correlations between musical and signal-related factors with the agreement of both humans and algorithms. Musical factors were derived from the annotated melody (e.g. melodic range, melodic density, etc.), and the factor derived from the audio signal was the degree of predominance of the melody over the accompaniment. Results showed that the most accurate algorithms on this dataset were those which were less correlated with the degree of predominance of the melody.

5.3.3 Datasets

As previously introduced, we have published several datasets during the course of this thesis. The first and most relevant dataset for this work is *Orchset*[52], which is intended to be used for the evaluation of melody extraction algorithms. As described in Section 3.2, this collection contains 64 audio excerpts focused on symphonic music, with their corresponding annotation of the melody. This dataset has been employed in the audio melody extraction task in MIREX evaluation exchange, and it has now become publicly available (Bosch et al., 2016b). Melody is defined here as "*the single (monophonic) pitch sequence that a listener might reproduce if asked to whistle or hum a piece of polyphonic music*". The dataset creation comprised several tasks: excerpts selection, recording sessions of people singing along with the excerpts, analysis of the recordings and melody annotation. The public release of this dataset contributed to the win of the "*Maria de Maeztu Research reproducibility award*" for PhD activities, which aims to promote the discussion and implementation of mechanisms that foster the reproducibility of research.

We also collaborated in the creation of the *PHENICX-Anechoic dataset*[53], which was published and employed for the evaluation of score-informed source separation in (Miron et al., 2016), and consists of four passages of symphonic music from the Classical and Romantic periods. The original audio recordings (Pätynen et al., 2008) contained a higher number of instruments, but in order to have a consistent evaluation between pieces, we selected the following instruments: violin, viola, cello, double bass, oboe, flute, clarinet, horn, trumpet and bassoon. We then created a ground truth score, by manually annotating the notes played by each of the instruments. This dataset is thus also useful for tasks such as score following, multipitch estimation, transcription or instrument detection.

An additional dataset published during the course of this thesis is *IRMAS*[54]: A Dataset

[52] http://mtg.upf.edu/download/datasets/orchset
[53] http://mtg.upf.edu/download/datasets/phenicx-anechoic
[54] http://mtg.upf.edu/download/datasets/irmas/

for Instrument Recognition in Musical Audio Signals, which was based on a dataset created during the PhD thesis of Ferdinand Fuhrmann (Fuhrmann, 2012), and modified and adapted for the evaluation of the article (Bosch et al., 2012a). IRMAS is intended to be used for training and testing automatic instrument recognition methods, in a varied set of professionally produced music recordings. The dataset includes a total of 6705 excerpts for training, and 2874 excerpts for testing. The instruments considered are: cello, clarinet, flute, acoustic guitar, electric guitar, organ, piano, saxophone, trumpet, violin, and human singing voice. This dataset is not exploited in this dissertation.

5.3.4 Evaluation metrics

We have proposed a set of metrics for the evaluation of pitch salience functions and melody extraction methods. Pitch salience function evaluation in the context of melody extraction had only focused on the estimation of the melody pitch. In Section 4.2.3 we proposed metrics related to measuring the salience of melody pitches in comparison to the salience in unvoiced frames, which is an important indicator of the goodness of a pitch salience function, especially when used for the creation of pitch contours. In Section 3.4.4 we also proposed several metrics especially relevant in the context of symphonic music, which allow gaining further knowledge from melody extraction methods, in terms of the smoothness of the extracted melody. This could be relevant for real applications such as visualisation, source separation and transcription.

5.3.5 State of the art evaluation

In Chapter 3 we analysed the performance of state-of-the-art pitch estimation methods in the context of melody extraction on symphonic music. The evaluation studied multipitch estimation methods, pitch salience functions and melody extraction methods, in order to understand where the limitations arise when dealing with such complex data. We have also analysed the correlation between melody extraction accuracy and musical characteristics. Specifically, we focused on characteristics of the melody to be analysed, as well as properties of the signal such as the energetic predominance of the melody over the accompaniment. We also proposed a novel set of metrics to gain more knowledge from melody extraction algorithms, related to the smoothness of the melody contour.

In Chapter 4 we also evaluated melody extraction methods in a more varied set of data, including genres such as pop, rock, jazz, country, and symphonic music. Additionally, we considered multiple melody definitions, and studied the benefits of data-driven approaches in comparison to rule-based ones for adapting to different kinds of data.

5.3.6 Novel methods

We have also proposed the use of unsupervised and supervised methods for melody pitch detection, as well as the combination of several pitch salience functions with melody tracking based on pitch contours characterisation. The main contributions are:

Factorisation of pitch estimation methods based on pitch contours. In Chapter 4 we have presented several melody extraction methods based on the use of different salience functions, with different melody decoding algorithms, all based on pitch contours as a mid-level representation. The factorisation of the different steps in the proposed framework allows the experimentation with different combinations of methods. The proposed methods are implemented in open source software[55] built upon other libraries[56,57,58], which will hopefully contribute to further improvements and better research in the topic. The public release of our code contributed to the win of the "*Maria de Maeztu Research reproducibility award*" for PhD activities.

We also contributed (Bittner et al., 2017) to the conception of an open source library called `motif` [59], which is built around the factorization paradigm used in this thesis. This library contains implementations of several contour extraction and contour classification methods that can be applied to any pitch estimation task. The library is built to make it easy to add new methods and to experiment with different combinations of methods.

Improving melody-oriented pitch salience functions with source-filter models. In Section 4.2, we propose the adaptation of pitch salience functions based on source-filter models for pitch contour creation. We first proposed the combination (CB) with a salience function based on harmonic summation. Our second proposal was based on weighting the salience function with an estimate of the energy of the melody source. Both methods also employed Gaussian filtering for smoothing. Results showed that the proposed salience functions are able to increase the salience of the melody pitch over the rest of pitches, in comparison to those obtained with SIMM or harmonic summation. We also showed that the proposed methods assign lower salience to unvoiced frames, which is a desirable property for a salience function used for pitch contour creation.

Combining source-filter models and pitch contour selection. In Section 4.4 we proposed a method which combines a source-filter model with pitch contour selection.

[55] https://github.com/juanjobosch/SourceFilterContoursMelody
[56] https://github.com/MTG/essentia
[57] https://github.com/rabitt/contour_classification
[58] https://github.com/wslihgt/separateLeadStereo
[59] http://www.github.com/rabitt/motif

This method obtained a remarkably higher overall accuracy in a symphonic music dataset, in comparison to other approaches in MIREX 2015 and 2016. This is also the case in comparison to the algorithm MELODIA which represents the state of the art in the field. Additionally, the proposed melody extraction method obtained best or second best overall accuracy in MIREX 2015 evaluation campaign in datasets containing jazz, pop, rock, R&B and Indian classical music. More importantly, in larger and more realistic datasets such as MedleyDB, it has also proved to perform better than the state of the art.

Combining source-filter models and pitch contour classification. In Section 4.5 we proposed a method which combines a source-filter model with a melody decoding method based on pitch contour classification. The main difference with the method based on heuristics is that it learns the characteristics of the data to be analysed using a Random Forest Classifier. Decoding takes place using the Viterbi algorithm, where the likelihoods are the estimated probabilities of each contour as being melodic. The main advantage of this method in comparison to pitch contour selection is that the heuristic rules used are here substituted by a classification algorithm, which is able to learn from the data to be modelled, and we do not need to manually tune the parameters to improve the melody extraction, e.g. the voicing detection threshold. A further advantage of this method is that it allows an easier integration of new pitch contour features.

Use of timbre, spatial and tonal features for pitch contour classification and melody decoding. In Section 4.6 we propose a novel approach for the characterisation of pitch contours with features related to timbre, tonality and spatial position in the stereo panorama. Such characteristics are first used to model melodic contours in the dataset to be analysed, and then exploited for the classification of pitch contours as being melodic or not. Using the proposed additional features for melody extraction with the same classification method leads to less false alarms, and higher overall accuracy in a wide range of music data.

Multiple line decoding. In Section 4.7 we proposed a method for decoding multiple melodic lines, based on the previously introduced set of features, and a pitch contour classification framework. Instead of iteratively decoding the melodic lines, we perform a joint multiple-line decoding. To do so, we create "nodes", which represent an ordered combination of candidate pitches. Decoding is performed in the node-time space, using Viterbi decoding, and uses data-driven models to favour transitions between melodic contours. This approach is trained and evaluated on MedleyDB with MEL3 definition, which contains the annotation multiple melodic lines. The proposed approach achieves similar or higher recall, and a much higher precision and accuracy in comparison to state-of-the-art multipitch estimation approaches.

Informed methods for melody pitch estimation. In Section 3.6 we conducted some investigations on the use of timbre-informed methods for melody pitch estimation, based on the use of instrument specific spectral templates in a Probabilistic Latent Component Analysis (PLCA) framework. We also studied the expansion of the set of spectral templates, in an unsupervised fashion, by analysing the music signal under analysis. We have also started to explore the use of score information as prior knowledge in a prototype application, using a very simple method for score-informed melody extraction and melody instrument identification.

5.3.7 Applications

Even though it has not been the main focus of this thesis, we have also explored the use of the proposed methods in practical applications, where the developed algorithms have been integrated and exploited. In Section 4.4.5 we explored the use of a melody extraction approach within a source separation method, aiming to separate melody from accompaniment. The use of the proposed approach led to improvements in the separation quality in comparison to using the original melody extraction method.

In Section 5.2 we presented a web-based melody visualization prototype, in the context of symphonic classical music. The estimated melody is presented in a piano-roll, as well as the estimation of the instruments which play it. This application integrates musical scores in order to improve the melody extraction and perform instrument detection.

5.4 Future perspectives

In this section, we introduce some future perspectives related to our research. In addition, we also mention some current directions and ongoing work related to this dissertation.

5.4.1 Towards multiple pitch streaming

We have ongoing work related to the streaming of multiple pitches into different melodic lines. This is based on the multipitch decoding method presented in Section 4.7, and on transition models to maximize transition likelihoods between neighbouring contours without temporal overlap.

After Viterbi decoding on each group of contours, we re-evaluate the assignment of the pitch sequences to melody lines, favouring transitions which are more likely to occur within a melody line. We are exploring the use of transition models (see Section 4.7.2) for the assignment of additional weights which would, for instance, increase pitch continuity within all decoded melody lines. As we introduced in Section 4.6, spatial features are potentially helpful for the assignment of pitches to melody lines,

since musical instruments commonly stay in a constant stereo panning position in the whole recording.

5.4.2 Automatic estimation of instrument activations

Relatively few methods in the literature attempt to perform multipitch streaming, and they commonly rely on features related to the sources that produce each of the pitches or musical notes (Duan et al., 2014; Kirchhoff et al., 2013). In Section 4.6 we proposed the characterisation of pitch contours with timbre-related features, computed from the lead instrument spectral envelope which is learnt in the source-filter decomposition. Results showed that such features are helpful to discriminate melodic from non-melodic contours, which improved voicing detection and thus overall accuracy. However, one drawback of such feature is that the values are the same for all pitches active at a given instant, and thus it is not useful to discriminate between pitch contours with a complete overlap in time. It would be thus interesting to use other features to characterise each contour with timbre information related to the instrument which produced that sequence of pitches. For instance, it is possible to automatically estimate the probability that a given musical instrument is playing a given note, using supervised methods such as PLCA (Benetos, 2012).

We conducted a preliminary test on Orchset, to investigate if it would be possible to automatically guess the instrument which plays the melody in a given symphonic music excerpt. To do so, we complemented the set of contour characteristics, with additional "instrument activation" features, related to the probability of each contour being generated by each of the orchestral instruments considered in Section 3.6. This probability was computed from the instrument specific activation matrix obtained with PLCA. We then measured the overlap of each contour with the ground truth annotations, filtered all contours with an overlap lower than 50%, and annotated the remaining contours as melodic. We then used Orchset annotations with the instrument family (or families) which play the melody in each excerpt to perform three tests, selecting the excerpts in which the melody was played by only 1) strings, 2) brass or 3) woodwind sections. In each of the tests, we computed the mean values of the instrument activation features on the melodic contours, and saw if the results were coherent with the annotated instrument family. In the case of strings, violin and viola obtained the highest values, which is correct. In the case of brass, it was oboe and trumpet, which is partially correct. Finally, in the case of woodwinds, the highest mean activations were obtained by clarinet, flute and oboe, which is also correct. Even though these results are encouraging, we recall that the amount of excerpts per instrument family playing the melody is not homogeneous, so further investigations are needed with a larger dataset, especially with more brass and woodwind examples (see Section 3.2.1).

Further work deals with the integration of such timbre features in our melody extraction framework in order to improve timbre-based multi pitch streaming and to allow

the assignment of pitches to different melody lines, according to the instrument that plays them. Such approach could be also used for the visualisation prototype introduced in Section 5.2.2, instead of relying on an aligned score. While music scores are commonly available for most of the symphonic repertoire, they may correspond to different instrument arrangements or interpretations. Even with a score which matches the given audio recording, the automatic alignment at a note level is a very complicated task (Miron et al., 2015). This is particularly true in symphonic music due to the presence of slow and soft passages, and possible structural differences due to repetitions (Grachten et al., 2013).

5.4.3 Learning more from data

We have seen that both unsupervised and supervised learning methods are useful for melody extraction. In the proposed framework, the method employed for pitch salience estimation models the data under analysis to produce a melody-oriented pitch representation, but it is unsupervised. An interesting future research direction is to explore a supervised method for pitch salience estimation, such as Deep Neural Networks (DNN). An advantage of using a supervised method is that it would be possible to train it with different melody definitions. As we have seen, one of the benefits of the proposed pitch salience is that it tends to be very sparse, and the melody pitch is much more salient than the rest of pitches. The drawback is that it is task-specific, and it would be beneficial to use a different pitch salience function for multipitch estimation. The use of a supervised method for pitch salience estimation would even allow to learn a model to estimate all present pitches, when appropriately trained. One of the main challenges is to deal with the relatively small size of the datasets in comparison to other disciplines, and the cost of creating annotated data. We expect data augmentation and meaningful data synthesis to become key factors for improving data-driven melody extraction (and MIR in general), specially in combination with multi-track datasets, since they provide much more flexibility in the transformations and combinations.

Pitch contours have proven to be a useful mid-level representation, and therefore we foresee their use in combination of deep neural networks as a supervised method for the computation of pitch salience. However, the formation of pitch contours from the peaks of the pitch salience function is not a data-driven process. As we have seen, the parameters used during contour creation have a very important effect on the amount and shape of the created contours, and thus on the melody extraction voicing recall. It would thus be interesting to also learn the ideal contour creation parameters from training data. A good example is the creation of pitch contours in orchestral classical music, which features a higher dynamic range than in other genres. In that case, we have seen that it is better to allow more variance in the salience peaks, thus creating a higher amount of contours, which is ideal for the symphonic music dataset.

Further work also deals with learning the best configuration within a PLCA frame-

work for melody pitch estimation, such as the parameters for pre- and post-filtering as described in Section 3.6.

5.4.4 Multimodal melody extraction

We foresee that the use of additional modalities could help the automatic extraction of the melody. A first example would be the use of a symbolic representation of the piece, which could be used for score-informed audio melody extraction, given that the score is available in a digital format and is properly aligned to the piece under analysis (either manually or automatically). Other modalities such as video could be explored, which would give information about the instruments being played, or help estimating the notes, onsets and offsets.

Appendix A

Publications by the Author

We here provide a list of publications by the author related with the thesis work.

Peer-reviewed journals

- **Bosch, J.**, Marxer, R., and Gómez, E., (2016). Evaluation and combination of pitch estimation methods for melody extraction in symphonic classical music. *Journal of New Music Research*, 45(2), 101–117.

- Miron, M., Carabias-Orti, J.J., **Bosch, J.**, Gómez, E. and Janer, J. (2016) Score-Informed Source Separation for Multichannel Orchestral Recordings. *Journal of Electrical and Computer Engineering*, doi:10.1155/2016/8363507.

Peer-reviewed conferences

- Bittner, R. M., Salamon, J., **Bosch, J.** and Bello, J. (2017). Pitch Contours as a Mid-Level Representation for Music Informatics [accepted], In *Proceedings of AES International Conference on Semantic Audio*.

- **Bosch, J.** and Gómez, E., (2016) Melody extraction based on a source-filter model using pitch contour selection, In *Proceedings of Sound and Music Computing Conference (SMC)*, pp. 67-74

- **Bosch, J.**, Bittner, R. M., Salamon, J. and Gómez, E. (2016) A Comparison of Melody Extraction Methods Based on Source-Filter Modelling, In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 571-577

- **Bosch, J.**. and Gómez, E. (2014) Melody extraction in symphonic classical music: a comparative study of mutual agreement between humans and algorithms, In *Proceedings of the Conference on Interdisciplinary Musicology – CIM14*.

- Liem, C.C., Van Der Sterren, R., van Tilburg, M., Sarasúa, A., **Bosch, J.**, Janer, J., Melenhorst, M., Gómez, E. and Hanjalic, A. (2013) Innovating the classical music experience in the PHENICX project: use cases and initial user feedback. In *Proceedings of the 1st International Workshop on Interactive Content Consumption (WSICC) at EuroITV*.

- **Bosch, J.** (2013) Automatic Melodic and Structural Analysis of Music Material for Enriched Concert Related Experiences, In *Proceedings of the ACM international conference on Multimedia*, pp. 1067-1070

- **Bosch, J.**, Janer, J., Fuhrmann, F. and Herrera, P. (2012) A Comparison of Sound Segregation Techniques for Predominant Instrument Recognition in Musical Audio Signals, In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. pp. 559-564

- **Bosch, J.** , Kondo, K., Marxer, R. and Janer, J. (2012) Score-informed and timbre independent lead instrument separation in real-world scenarios. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, IEEE, pp. 2417-2421

Other contributions to conferences

- **Bosch, J.**, Gómez, E., Melody extraction for MIREX 2016, Music Information Retrieval Evaluation eXchange (MIREX), 2016

- **Bosch, J.**, Gómez, E., Melody extraction by means of a source-filter model and pitch contour characterization (MIREX 2015), Music Information Retrieval Evaluation eXchange (MIREX), 2015

- **Bosch, J.**, Mayor O., Gómez E., Melovizz: A Web-based tool for Score-Informed Melody Extraction Visualization, 2015. In Late-Breaking Demo Session of the 16th International Society for Music Information Retrieval Conference.

Appendix B
Experiment details

MedleyDB files used for source separation experiment (Section 4.4.5)

AimeeNorwich_Child.wav
AlexanderRoss_VelvetCurtain.wav
Auctioneer_OurFutureFaces.wav
AvaLuna_Waterduct.wav
BrandonWebster_DontHearAThing.wav
BrandonWebster_YesSirICanFly.wav
ClaraBerryAndWooldog_AirTraffic.wav
ClaraBerryAndWooldog_Boys.wav
ClaraBerryAndWooldog_Stella.wav
ClaraBerryAndWooldog_WaltzForMyVictims.wav
Creepoid_OldTree.wav
DreamersOfTheGhetto_HeavyLove.wav
EthanHein_1930sSynthAndUprightBass.wav
EthanHein_BluesForNofi.wav
EthanHein_GirlOnABridge.wav
HeladoNegro_MitadDelMundo.wav
HezekiahJones_BorrowedHeart.wav
HopAlong_SisterCities.wav
KarimDouaidy_Hopscotch.wav
LizNelson_Coldwar.wav
LizNelson_Rainfall.wav
Meaxic_TakeAStep.wav
Meaxic_YouListen.wav
MusicDelta_80sRock.wav

MusicDelta_Country1.wav
MusicDelta_Country2.wav
MusicDelta_Disco.wav
MusicDelta_FreeJazz.wav
MusicDelta_Gospel.wav
MusicDelta_Grunge.wav
MusicDelta_Hendrix.wav
MusicDelta_InTheHalloftheMountainKing.wav
MusicDelta_Pachelbel.wav
MusicDelta_Punk.wav
MusicDelta_Reggae.wav
MusicDelta_Rock.wav
MusicDelta_Rockabilly.wav
MusicDelta_Shadows.wav
MusicDelta_SpeedMetal.wav
MusicDelta_Vivaldi.wav
MusicDelta_Zeppelin.wav
NightPanther_Fire.wav
PortStWillow_StayEven.wav
SecretMountains_HighHorse.wav
Snowmine_Curfews.wav
StevenClark_Bounty.wav
StrandOfOaks_Spacestation.wav
SweetLights_YouLetMeDown.wav
TheDistricts_Vermont.wav
TheScarletBrand_LesFleursDuMal.wav
TheSoSoGlos_Emergency.wav

Appendix C
Glossary

C.1 Acronyms

CC	Chroma Continuity
MMA	Mean Mutual Agreement
OA	Overall Accuracy
OJ	Octave Jumps ratio
RCA	Raw Chroma Accuracy
RPA	Raw Pitch Accuracy
VFA	Voicing False Alarm rate
VR	Voicing Recall
WRCA	Weighted Raw Chroma Accuracy
WRPA	Weighted Raw Pitch Accuracy
AME	Audio Melody Extraction
API	Application Programming Interface
ASA	Auditory Scene Analysis
BEN	Multipitch estimation method, based on SI-PLCA (Benetos & Dixon, 2011)
BEN14	Multipitch estimation method, based on SI-PLCA used in MIREX 2014 (Benetos & Weyde, 2014)
BIT	Melody extraction method based on PCC (Bittner et al., 2015)
CAN	Pitch salience function (FChT) (Cancela et al., 2010)
CB	Pitch salience function based on the combination of a source-filter model and harmonic summation
CBC	Melody extraction method based on the salience function CB and pitch contour classification
CBM	Method based on CB for the estimation of multiple melodic lines
CBS	Melody extraction method based on the salience function CB and pitch contour selection
COMB	Multipitch estimation method based on the combination of salience functions peaks

CPV	Combination of Pitch and Voicing
CQT	Constant-Q Transform
DEF	Default pitch contour features: pitch (mean and deviation), salience (mean, standard deviation and sum), duration
DNN	Deep Neural Network
DRE	Melody and multipitch estimation method, Spectral peaks comparison and streaming rules (Dressler, 2012b,a)
DTW	Dynamic Time Warping
DUA	Multipitch estimation method based on Maximum Likelihood principles (Duan et al., 2010)
DUR	Melody extraction method based on a source-filter model (Durrieu et al., 2010)
ELF	Equal-Loudness Filters
EM	Expectation Maximisation
ERB	Equivalent Rectangular Bandwidth
ES	Matrix used for energy weighting
ESS	Melody extraction method by Salamon & Gómez (2012) as implemented in Essentia
EW	Pitch salience function based on a source-filter model, weighted by its frame-by-frame energy (combined with ES)
EWS	Melody extraction method based on the salience function EW and pitch contour selection
FChT	Fan Chirp Transform
FUE	Melody extraction method based on PLCA on the CQT (Fuentes et al., 2012)
GT	Generalisability Theory
HMM	Hidden Markov Model
HPCP	harmonic pitch-class profiles
HPSS	Harmonic/Percussive Source Separation
HS	Salience function based on Harmonic Summation
HSM	Method based on HS for the estimation of multiple melodic lines
IF	Instantaneous Frequency
MAR	Pitch salience function (Tikhonov regularisation) (Marxer, 2013)
MEL1	Melody 1 definition in MedleyDB
MEL2	Melody 2 definition in MedleyDB
MEL3	Melody 3 definition in MedleyDB
MFCC	Mel-frequency cepstral coefficient
MIR	Music Information Research
MRFFT	Multi-Resolution Fast Fourier Transform
NMF	Non-negative Matrix Factorisation
PCC	Pitch Contour Classification
PCS	Pitch Contour Selection
PDD	Probability Density function Discontiguity

PHENICX	Performances as Highly Enriched and Interactive Concert eXperiences
PLCA	Probabilistic Latent Component Analysis
RCOMB	Multipitch estimation method based on the combination of salience functions peaks, with neighbourhood refinement
RESTful	representational state transfer
RVS	Relative Voiced Salience
S/F	Source-Filter model
SAL	Melody extraction method based on PCS with salience function based on Harmonic Summation (Salamon & Gómez, 2012)
SIMM	Smooth Instantaneous Mixture Model
SIPLCA	Shift-Invariant Probabilistic Latent Component Analysis
SMS	Spectral Modeling Synthesis
SPT	Combination of default + spatial-related pitch contour features
STFT	Short Time Fourier Transform
SVD	Singing Voice Detection
SVM	Support Vector Machines
TIM	Combination of default + timbre-related pitch contour features
TON	Combination of default + tonal-related pitch contour features
TR	Tikhonov Regularisation

Bibliography

ANSI (S1.1-1994) (1994). American National Standard Acoustical Terminology. Standard, Standards Secretariat - Acoustical Society of America. [Cited on page 7.]

Arora, V. & Behera, L. (2013). On-line melody extraction from polyphonic audio using harmonic cluster tracking. *IEEE Transactions on Audio, Speech, and Language Processing*, *21*(3), 520–530. [Cited on pages 24 and 36.]

Arora, V. & Behera, L. (2015). Multiple f0 estimation and source clustering of polyphonic music audio using plca and hmrfs. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *23*(2), 278–287. [Cited on page 35.]

Bachem, A. (1950). Tone height and tone chroma as two different pitch qualities. *Acta Psychologica*, *7*, 80–88. [Cited on page 3.]

Baines, A. & Temperley, N. (). Pitch. the oxford companion to music, oxford music online. http://www.oxfordmusiconline.com/subscriber/article/opr/t114/e5199. (Date last accessed Mar 2017). [Cited on page 3.]

Bay, M., Ehmann, A. F., beauchamp, J. W., Smaragdis, P., & Downie, J. S. (2012). Second fiddle is important too: pitch tracking individual voices in polyphonic music. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 319–324. [Cited on page 35.]

Bay, M., Ehmann, A. F., & Downie, J. S. (2009). Evaluation of multiple-f0 estimation and tracking systems. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 315–320. [Cited on page 39.]

Benetos, E. (2012). *Automatic transcription of polyphonic music exploiting temporal evolution*. Ph.D. thesis, School of Electronic Engineering and Computer Science, Queen Mary University of London, UK. [Cited on pages 7, 10, 21, and 152.]

Benetos, E., Badeau, R., Weyde, T., & Richard, G. (2014). Template adaptation for improving automatic music transcription. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 175–180. [Cited on pages 27, 35, 82, 83, and 84.]

Benetos, E. & Dixon, S. (2011). Multiple-instrument polyphonic music transcription using a convolutive probabilistic model. In *Proceedings of the Sound and Music Computing Conference (SMC)*, pp. 19–24. [Cited on pages 23, 24, 26, 27, 37, 47, 65, 79, and 159.]

Benetos, E. & Dixon, S. (2012). A shift-invariant latent variable model for automatic music transcription. *Computer Music Journal*, *36*(4), 81–94. [Cited on page 27.]

Benetos, E. & Dixon, S. (2013). Multiple-instrument polyphonic music transcription using a temporally-constrained shift-invariant model. *Journal of the Acoustical Society of America*, *133*(3), 1727–1741. [Cited on pages 27 and 35.]

Benetos, E. & Weyde, T. (2014). Multiple-f0 estimation and note tracking for mirex 2014 using a variable-q transform. *Music Information Retrieval Evaluation eXchange (MIREX)*. [Cited on pages 134 and 159.]

Benetos, E. & Weyde, T. (2015a). An efficient temporally-constrained probabilistic model for multiple-instrument music transcription. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 701–707. [Cited on pages 27, 79, and 83.]

Benetos, E. & Weyde, T. (2015b). Multiple-f0 estimation and note tracking for mirex 2015 using a sound state-based spectrogram factorization model. In *Music Information Retrieval Evaluation eXchange (MIREX)*. [Cited on page 27.]

Berg-Kirkpatrick, T., Andreas, J., & Klein, D. (2014). Unsupervised transcription of piano music. In *Advances in Neural Information Processing Systems Conference*, pp. 1538–1546. [Cited on page 23.]

Bertin, N., Badeau, R., & Vincent, E. (2010). Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription. *IEEE Transactions on Audio, Speech, and Language Processing*, *18*(3), 538–549. [Cited on page 23.]

Bittner, R., Salamon, J., Essid, S., & Bello, J. (2015). Melody extraction by contour classification. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 500–506. [Cited on pages 19, 20, 22, 31, 32, 33, 36, 88, 98, 100, 105, 108, 109, 112, 113, 120, 124, 129, 130, 131, 132, 142, and 159.]

Bittner, R., Salamon, J., Tierney, M., Mauch, M., Cannam, C., & Bello, J. (2014). Medleydb: a multitrack dataset for annotation-intensive mir research. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 155–160. [Cited on pages 2, 19, 38, 43, and 48.]

Bittner, R. M., Salamon, J., Bosch, J., & Bello, J. (2017). Pitch contours as a mid-level representation for music informatics [accepted]. In *Proceedings of the AES International Conference on Semantic Audio*. [Cited on page 149.]

Böck, S. & Schedl, M. (2012). Polyphonic piano note transcription with recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 121–124. [Cited on pages 23 and 30.]

Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J. R., & Serra, X. (2013). Essentia: An audio analysis library for music information retrieval. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 493–498. [Cited on pages 20, 24, and 90.]

Bosch, J., Bittner, R. M., Salamon, J., & Gómez, E. (2016a). A comparison of melody extraction methods based on source-filter modelling. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 571–577. [Cited on page 21.]

Bosch, J. & Gómez, E. (2015). Melody extraction by means of a source-filter model and pitch contour characterization (MIREX 2015). In *Music Information Retrieval Evaluation eXchange (MIREX)*. [Cited on page 10.]

Bosch, J. & Gómez, E. (2016). Melody extraction based on a source-filter model using pitch contour selection. In *Proceedings of the Sound and Music Computing Conference (SMC)*, pp. 67–74. [Cited on page 21.]

Bosch, J., Janer, J., Fuhrmann, F., & Herrera, P. (2012a). A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 559–564. [Cited on pages 7, 48, 115, and 148.]

Bosch, J., Kondo, K., Marxer, R., & Janer, J. (2012b). Score-informed and timbre independent lead instrument separation in real-world scenarios. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pp. 2417–2421. [Cited on page 2.]

Bosch, J., Marxer, R., & Gómez, E. (2016b). Evaluation and combination of pitch estimation methods for melody extraction in symphonic classical music. *Journal of New Music Research*, *45*(2), 101–117. [Cited on pages 11, 31, and 147.]

Bosch, J., Mayor, O., & Gómez, E. (2015). Melovizz: A web-based tool for score-informed melody extraction visualization. In *In Late-Breaking Demo Session of the 16th International Society for Music Information Retrieval Conference*. [Cited on page 2.]

Bouvier, D., Obin, N., Liuni, M., & Roebel, A. (2016). A source/filter model with adaptive constraints for nmf-based speech separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 131–135. IEEE. [Cited on page 27.]

Bregman, A. (1994). *Auditory scene analysis: The perceptual organization of sound*. MIT press. [Cited on page 14.]

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32. [Cited on page 34.]

Bronkhorst, A. W. (2000). The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions. *Acta Acustica united with Acustica*, *86*(1), 117–128. [Cited on page 15.]

Burred, J. J. (2009). *From sparse models to timbre learning: new methods for musical source separation*. Ph.D. thesis, Technical University of Berlin. [Cited on page 115.]

Caetano, M. & Rodet, X. (2012). A source-filter model for musical instrument sound transformation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 137–140. IEEE. [Cited on page 27.]

Canadas-Quesada, F. J., Vera-Candeas, P., Ruiz-Reyes, N., Carabias-Orti, J., & Cabanas-Molero, P. (2014). Percussive/harmonic sound separation by non-negative matrix factorization with smoothness/sparseness constraints. *EURASIP Journal on Audio, Speech, and Music Processing*, *2014*(1), 1–17. [Cited on page 15.]

Cancela, P. (2008). Tracking melody in polyphonic audio. *Music Information Retrieval Evaluation eXchange (MIREX)*. [Cited on page 30.]

Cancela, P., López, E., & Rocamora, M. (2010). Fan chirp transform for music representation. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pp. 1–8. [Cited on pages 20, 24, 25, 30, 47, 64, and 159.]

Carabias-Orti, J., Virtanen, T., Vera-Candeas, P., Ruiz-Reyes, N., & Cañadas-Quesada, F. (2011). Musical instrument sound multi-excitation model for non-negative spectrogram factorization. *IEEE Journal of selected Topics in Signal Processing*, *5*(6), 1144–1158. [Cited on pages 23, 26, and 35.]

Carabias-Orti, J. J., Rodríguez-Serrano, F. J., Vera-Candeas, P., Ruiz-Reyes, N., & Cañadas-Quesada, F. J. (2015). An audio to score alignment framework using spectral factorization and dynamic time warping. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 742–748. [Cited on page 144.]

Cemgil, A. T., Kappen, H. J., & Barber, D. (2006). A generative model for music transcription. *IEEE Transactions on Audio, Speech, and Language Processing*, *14*(2), 679–694. [Cited on page 21.]

Ceron, E. C. (2014). *Pitch-informed solo and accompaniment separation*. Ph.D. thesis, Universität Ilmenau. [Cited on page 115.]

Chan, T.-S., Yeh, T.-C., Fan, Z.-C., Chen, H.-W., Su, L., Yang, Y.-H., & Jang, R. (2015). Vocal activity informed singing voice separation with the ikala dataset. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 718–722. IEEE. [Cited on pages 15 and 43.]

Cheng, T., Dixon, S., & Mauch, M. (2014). A comparison of extended source-filter models for musical signal reconstruction. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pp. 1–7. [Cited on page 27.]

Cogliati, A. & Duan, Z. (2015). Piano music transcription with fast convolutional sparse coding. In *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. [Cited on page 23.]

Cont, A., Dubnov, S., & Wessel, D. (2007). Realtime multiple-pitch and multiple-instrument recognition for music signals using sparse non-negative constraints. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*. [Cited on page 35.]

Cooke, M., Hershey, J. R., & Rennie, S. J. (2010). Monaural speech separation and recognition challenge. *Computer Speech & Language*, *24*(1), 1–15. [Cited on page 21.]

Dannenberg, R. B., Birmingham, W. P., Pardo, B., Hu, N., Meek, C., & Tzanetakis, G. (2007). A comparative evaluation of search techniques for query-by-humming using the musart testbed. *Journal of the American Society for Information Science and Technology*, *58*(5), 687–701. [Cited on page 2.]

Davis, S. & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *28*(4), 357–366. [Cited on pages 36 and 114.]

Davy, M. & Godsill, S. (2003). Bayesian harmonic models for musical signal analysis. *Bayesian Statistics*, *7*, 105–124. [Cited on page 21.]

De Cheveigné, A. (1998). Cancellation model of pitch perception. *Journal of the Acoustical Society of America*, *103*(3), 1261–1271. [Cited on page 17.]

De Cheveigné, A. (2006). Multiple f0 estimation. In D. L. Wang & G. J. Brown (Eds.) *Computational Auditory Scene Analysis, Algorithms and Applications*, pp. 45–79. IEEE Press/Wiley. [Cited on page 16.]

De Cheveigné, A. & Kawahara, H. (2002). Yin, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America*, *111*(4), 1917–1930. [Cited on page 17.]

Dessein, A., Cont, A., & Lemaitre, G. (2010). Real-time polyphonic music transcription with non-negative matrix factorization and beta-divergence. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 489–494. [Cited on page 23.]

Dixon, S. & Widmer, G. (2005). Match: A music alignment tool chest. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 492–497. [Cited on page 144.]

Dobson, M. C. (2010). New audiences for classical music: The experiences of non-attenders at live orchestral concerts. *Journal of New Music Research*, *39*(2), 111–124. [Cited on page 9.]

Doval, B. & Rodet, X. (1993). Fundamental frequency estimation and tracking using maximum likelihood harmonic matching and hmms. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, pp. 221–224. [Cited on page 17.]

Downie, J. S. (2008). The music information retrieval evaluation exchange (2005-2007): A window into music information retrieval research. *Acoustical Science and Technology*, *29*(4), 247–255. [Cited on page 4.]

Dressler, K. (2011). An auditory streaming approach for melody extraction from polyphonic music. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 19–24. [Cited on pages 25 and 35.]

Dressler, K. (2012a). Multiple fundamental frequency extraction for MIREX 2012. In *Music Information Retrieval Evaluation eXchange (MIREX)*. [Cited on pages 23, 24, 35, 37, 47, 64, 65, and 160.]

Dressler, K. (2012b). Towards Computational Auditory Scene Analysis: Melody Extraction from Polyphonic Music. In *Proceedings of the International Symposium on Computer Music Multidisciplinary Research (CMMR)*, pp. 319–334. [Cited on pages 18, 22, 24, 36, 42, 47, 51, 55, 64, and 160.]

Drobisch, M. (1846). Uber die mathematische bestimmung der musikalischen intervalle. *Fürstlich, Jablonowskischen Gesellschaft der Wissenschaften. Leipzig: Weidmann'sche Buchlandlung*. [Cited on page 3.]

Duan, Z., Han, J., & Pardo, B. (2014). Multi-pitch streaming of harmonic sound mixtures. *IEEE Transactions on Audio, Speech, and Language Processing*, *22*(1), 138–150. [Cited on pages 7, 35, 41, and 152.]

Duan, Z., Pardo, B., & Zhang, C. (2010). Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions. *IEEE Transactions on Audio, Speech, and Language Processing*, *18*(8), 2121–2133. [Cited on pages 23, 24, 37, 47, 65, 66, 70, 72, 134, and 160.]

Duan, Z. & Temperley, D. (2014). Note-level music transcription by maximum likelihood sampling. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. [Cited on page 23.]

Duan, Z., Zhang, Y., Zhang, C., & Shi, Z. (2008). Unsupervised single-channel music source separation by average harmonic structure modeling. *IEEE Transactions on Audio, Speech, and Language Processing*, *16*(4), 766–778. [Cited on page 21.]

Durrieu, J., David, B., & Richard, G. (2011). A musically motivated mid-level representation for pitch estimation and musical audio source separation. *IEEE Journal of Selected Topics in Signal Processing*, *5*(6), 1180–1191. [Cited on pages 20, 24, 26, 28, 47, 64, 72, 73, 88, 89, 90, 93, 102, and 128.]

Durrieu, J., Richard, G., David, B., & Févotte, C. (2010). Source/filter model for unsupervised main melody extraction from polyphonic audio signals. *IEEE Transactions on Audio, Speech, and Language Processing*, *18*(3), 564–575. [Cited on pages 2, 5, 7, 14, 15, 18, 19, 20, 22, 27, 28, 30, 36, 42, 47, 64, 72, 74, 76, 87, 90, 92, 99, 100, 107, 115, 129, 130, 138, 142, and 160.]

Eerola, T. & North, A. (2000). Expectancy-based model of melodic complexity. In *Proceedings of the International Conference on Music Perception and Cognition*. [Cited on page 50.]

Eerola, T. & Toiviainen, P. (2004). *MIDI Toolbox: MATLAB Tools for Music Research*. University of Jyväskylä. [Cited on page 49.]

Ellis, D. & Poliner, G. (2006). Classification-based melody transcription. *Machine Learning*, *65*(2-3), 439–456. [Cited on page 18.]

Elowsson, A. & Friberg, A. (2014). Polyphonic transcription with deep layered learning. In *Music Information Retrieval Evaluation eXchange (MIREX)*. [Cited on page 30.]

Emiya, V., Badeau, R., & David, B. (2010). Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE Transactions on Audio, Speech, and Language Processing*, *18*(6), 1643–1654. [Cited on page 23.]

Emiya, V., Vincent, E., Harlander, N., & Hohmann, V. (2011). Subjective and Objective Quality Assessment of Audio Source Separation. *IEEE Transactions on Audio, Speech, and Language Processing*, *19*(7), 2046–2057. [Cited on page 107.]

Ewert, S., Plumbley, M., & Sandler, M. (2015). A dynamic programming variant of non-negative matrix deconvolution for the transcription of struck string instruments. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 569–573. [Cited on page 23.]

Fan, Z. C., Jang, J. S. R., & Lu, C. L. (2016). Singing voice separation and pitch extraction from monaural polyphonic audio music via dnn and adaptive pitch tracking. In *Proceedings of the IEEE Second International Conference on Multimedia Big Data (BigMM)*, pp. 178–185. [Cited on page 19.]

Flexer, A. (2014). On inter-rater agreement in audio music similarity. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 245–250. [Cited on page 48.]

Forney, G. D. (1973). The Viterbi algorithm. *IEEE*, *61*(3), 268–278. [Cited on pages 17, 27, and 34.]

Friberg, A. & Ahlbäck, S. (2009). Recognition of the main melody in a polyphonic symbolic score using perceptual knowledge. *Journal of New Music Research*, *38*(2), 155–169. [Cited on page 51.]

Fuentes, B., Badeau, R., & Richard, G. (2013). Harmonic adaptive latent component analysis of audio and application to music transcription. *IEEE Transactions on Audio, Speech, and Language Processing*, *21*(9), 1854–1866. [Cited on page 23.]

Fuentes, B., Liutkus, A., Badeau, R., & Richard, G. (2012). Probabilistic model for main melody extraction using constant-Q transform. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5357–5360. IEEE. [Cited on pages 20, 22, 24, 26, 30, 31, 36, 42, 47, 64, and 160.]

Fuhrmann, F. (2012). *Automatic musical instrument recognition from polyphonic music audio signals*. Ph.D. thesis, Universitat Pompeu Fabra. [Cited on pages 7 and 148.]

Ghias, A., Logan, J., Chamberlin, D., & Smith, B. C. (1995). Query by humming: musical information retrieval in an audio database. In *Proceedings of the ACM international conference on Multimedia*, pp. 231–236. ACM. [Cited on page 2.]

Gold, B. & Rabiner, L. (1969). Parallel processing techniques for estimating pitch periods of speech in the time domain. *The Journal of the Acoustical Society of America*, *46*(2B), 442–448. [Cited on page 17.]

Gómez, E. (2006). *Tonal description of music audio signals*. Ph.D. thesis, Universitat Pompeu Fabra. [Cited on page 116.]

Gómez, E. & Bonada, J. (2013). Towards computer-assisted flamenco transcription: An experimental comparison of automatic transcription algorithms as applied to a cappella singing. *Computer Music Journal*, *37*(2), 73–90. [Cited on page 2.]

Gómez, E., Grachten, M., Hanjalic, A., Janer, J., Jorda, S., Julia, C. F., Liem, C., Martorell, A., Schedl, M., & Widmer, G. (2013). Phenicx: Performances as highly enriched and interactive concert experiences. In *Proceedings of the Sound and Music Computing Conference (SMC)*. [Cited on pages 2, 8, and 143.]

Goto, M. (2004). A Real-Time Music-Scene-Description System: Predominant-F0 Estimation for Detecting Melody and Bass Lines in Real-World Audio Signals. *Speech Communication*, *43*(4), 311–329. [Cited on pages 14, 18, 22, and 102.]

Goto, M. (2006). Aist annotation for the rwc music database. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 359–360. [Cited on page 43.]

Goto, M. & Hayamizu, S. (1999). A real-time music scene description system: Detecting melody and bass lines in audio signals. In *Working Notes of the IJCAI-99 Workshop on Computational Auditory Scene Analysis*, pp. 31–40. [Cited on page 4.]

Grachten, M., Gasser, M., Arzt, A., & Widmer, G. (2013). Automatic alignment of music performances with structural differences. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 607–612. [Cited on page 153.]

Grindlay, G. & Ellis, D. (2011). Transcribing multi-instrument polyphonic music with hierarchical eigeninstruments. *IEEE Journal of Selected Topics in Signal Processing*, *5*(6), 1159–1169. [Cited on page 35.]

Gulati, S., Serra, J., Ishwar, V., & Serra, X. (2016a). Discovering rāga motifs by characterizing communities in networks of melodic patterns. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 286–290. IEEE. [Cited on page 2.]

Gulati, S., Serrà, J., Ishwar, V., Sentürk, S., & Serra, X. (2016b). Phrase-based raga recognition using vector space modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 66–70. [Cited on page 2.]

Han, J. & Chen, C.-W. (2011). Improving melody extraction using probabilistic latent component analysis. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 33–36. IEEE. [Cited on page 21.]

Han, K. & Wang, D. (2014). Neural network based pitch tracking in very noisy speech. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *22*(12), 2158–2168. [Cited on page 35.]

Heittola, T., Klapuri, A., & Virtanen, T. (2009). Musical instrument recognition in polyphonic audio using source-filter model for sound separation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 327–332. [Cited on page 27.]

Hermes, D. J. (1988). Measurement of pitch by subharmonic summation. *The journal of the acoustical society of America*, *83*(1), 257–264. [Cited on page 25.]

Holzapfel, A., Davies, M., Zapata, J., Oliveira, J., & Gouyon, F. (2012). Selective sampling for beat tracking evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*(9), 2539–2548. [Cited on page 65.]

Hsu, C. & Jang, J. (2010). Singing pitch extraction by voice vibrato/tremolo estimation and instrument partial deletion. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 525–530. [Cited on pages 18, 19, and 36.]

Hsu, C. L., Wang, D., & Jang, J. S. R. (2011). A trend estimation algorithm for singing pitch detection in musical recordings. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 393–396. [Cited on page 19.]

Huron, D. (1989). Voice denumerability in polyphonic music of homogeneous timbres. *Music Perception*, *6*(4), 361–382. [Cited on page 37.]

Huron, D. (2001). Tone and voice: A derivation of the rules of voice-leading from perceptual principles. *Music Perception*, *19*(1), 1–64. [Cited on page 50.]

Hyer, B. (2016). Tonality. grove music online, oxford music online. http://www.oxfordmusiconline.com/subscriber/article/grove/music/28102. (Date last accessed Dec 2016). [Cited on page 6.]

Jiang, D.-n., Zhang, W., Shen, L., & Cai, L. (2005). Prosody analysis and modeling for emotional speech synthesis. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 281–284. [Cited on page 21.]

Kameoka, H., Nishimoto, T., & Sagayama, S. (2007). A multipitch analyzer based on harmonic temporal structured clustering. *IEEE Transactions on Audio, Speech, and Language Processing*, *15*(3), 982–994. [Cited on page 23.]

Kirchhoff, H., Dixon, S., & Klapuri, A. (2013). Multiple instrument tracking based on reconstruction error, pitch continuity and instrument activity. In *Proceedings of the International Symposium on Computer Music Multidisciplinary Research (CMMR)*, pp. 894–903. [Cited on pages 35 and 152.]

Klapuri, A. (2000). Qualitative and quantitative aspects in the design of periodicity estimation algorithms. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pp. 1–4. IEEE. [Cited on page 17.]

Klapuri, A. (2003). Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *IEEE Transactions on Speech and Audio Processing*, *11*(6), 804–816. [Cited on pages 17, 23, and 37.]

Klapuri, A. (2006). Multiple fundamental frequency estimation by summing harmonic amplitudes. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 216–221. [Cited on page 24.]

Klapuri, A., Davy, M. et al. (2006). *Signal processing methods for music transcription*, vol. 1. Springer. [Cited on page 21.]

Koduri, G. K., Serrà, J., & Serra, X. (2012). Characterization of intonation in carnatic music by parametrizing pitch histograms. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 199–204. [Cited on page 2.]

Kullback, S. & Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, *22*(1), 79–86. [Cited on page 26.]

Kum, S., Oh, C., & Nam, J. (2016). Melody extraction on vocal segments using multi-column deep neural neworks. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 819–825. [Cited on pages 19, 20, and 22.]

Lahat, M., Niederjohn, R., & Krubsack, D. (1987). A spectral autocorrelation method for measurement of the fundamental frequency of noise-corrupted speech. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *35*(6), 741–750. [Cited on page 17.]

Lee, B. S. & Ellis, D. P. (2012). Noise robust pitch tracking by subband autocorrelation classification. In *Proceedings of Interspeech*, pp. 707–710. [Cited on page 35.]

Lee, D. D. & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*(6755), 788–791. [Cited on page 26.]

Lee, D. D. & Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pp. 556–562. [Cited on page 26.]

Leglaive, S., Hennequin, R., & Badeau, R. (2015). Singing voice detection with deep recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 121–125. IEEE. [Cited on page 36.]

Lehner, B., Widmer, G., & Bock, S. (2015). A low-latency, real-time-capable singing voice detection method with lstm recurrent neural networks. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pp. 21–25. IEEE. [Cited on page 36.]

Lehner, B., Widmer, G., & Sonnleitner, R. (2014). On the reduction of false positives in singing voice detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7480–7484. IEEE. [Cited on page 36.]

Leman, M. (1995). *Music and schema theory: cognitive foundations of systematic musicology*. Springer. [Cited on page 50.]

Levitin, D. J. (1999). Memory for musical attributes. In P. R. Cook (Ed.) *Music, Cognition, and Computerized Sound*, pp. 209–227. MIT Press. [Cited on page 4.]

Logan, B. et al. (2000). Mel frequency cepstral coefficients for music modeling. In *Proceedings of the International Symposium on Music Information Retrieval)*. [Cited on pages 36 and 114.]

Maher, R. C. & Beauchamp, J. W. (1994). Fundamental frequency estimation of musical signals using a two-way mismatch procedure. *The Journal of the Acoustical Society of America*, *95*(4), 2254–2263. [Cited on page 17.]

Marolt, M. (2004a). A connectionist approach to automatic transcription of polyphonic piano music. *IEEE Transactions on Multimedia*, *6*(3), 439–449. [Cited on pages 21, 23, and 30.]

Marolt, M. (2004b). On finding melodic lines in audio recordings. In *Proceedings of the International Conference on Digital Audio Effects*. [Cited on pages 18 and 22.]

Marolt, M. (2005). Audio melody extraction based on timbral similarity of melodic fragments. In *Proceedings of the EUROCON*, vol. 2, pp. 1288–1291. IEEE. [Cited on page 18.]

Martorell, A., Melenhorst, M., Gómez, E., Mayor, O., & Widmer, G. (2015). Off-line music visualization technology (PHENICX-D-WP6-150115-D6.1-OfflineMusicVisualizationTechnology-v1). Tech. rep., Universitat Pompeu Fabra. [Cited on page 143.]

Marxer, R. (2013). *Audio source separation for music in low-latency and high-latency scenarios*. Ph.D. thesis, Universitat Pompeu Fabra, Barcelona. [Cited on pages 2, 16, 21, 24, 26, 28, 30, 31, 47, 114, 115, and 160.]

Mauch, M. & Dixon, S. (2014). Pyin: A fundamental frequency estimator using probabilistic threshold distributions. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 659–663. [Cited on pages 17 and 54.]

Mauch, M., Fujihara, H., Yoshii, K., & Goto, M. (2011). Timbre and melody features for the recognition of vocal activity and instrumental solos in polyphonic music. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 233–238. [Cited on page 36.]

Mayor, O., Llimona, Q., Marchini, M., Papiotis, P., & Maestre, E. (2013). repovizz: a framework for remote storage, browsing, annotation, and exchange of multi-modal data. In *Proceedings of the ACM Multimedia*, pp. 415–416. [Cited on page 146.]

McFee, B., Humphrey, E. J., & Urbano, J. (2016). A plan for sustainable mir evaluation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 285–291. [Cited on page 14.]

Medan, Y., Yair, E., & Chazan, D. (1991). Super resolution pitch determination of speech signals. *IEEE transactions on signal processing*, *39*(1), 40–48. [Cited on page 17.]

Meddis, R. & O'Mard, L. (1997). A unitary model of pitch perception. *The Journal of the Acoustical Society of America*, *102*(3), 1811–1820. [Cited on page 17.]

Melenhorst, M. S. & Liem, C. C. (2015). Put the concert attendee in the spotlight. a user-centered design and development approach for classical concert applications. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 800–806. [Cited on page 9.]

Mesaros, A., Virtanen, T., & Klapuri, A. (2007). Singer identification in polyphonic music using vocal separation and pattern recognition methods. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 375–378. [Cited on page 2.]

Miron, M., Carabias, J., & Janer, J. (2015). Improving score-informed source separation for classical music through note refinement. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. [Cited on pages 144 and 153.]

Miron, M., Carabias-Orti, J. J., Bosch, J. J., Gómez, E., & Janer, J. (2016). Score-informed source separation for multichannel orchestral recordings. *Journal of Electrical and Computer Engineering*, *2016*. [Cited on pages 11, 115, 144, and 147.]

Müller, M. (2007). *Dynamic Time Warping*, pp. 69–84. Berlin, Heidelberg: Springer. [Cited on page 55.]

Mysore, G. & Smaragdis, P. (2009). Relative pitch estimation of multiple instruments. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 313–316. [Cited on page 35.]

Nam, J., , Ngiam, J., Lee, H., & Slaney, M. (2011). A classification-based polyphonic piano transcription approach using learned feature representations. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 175–180. [Cited on page 30.]

Ni, Y., McVicar, M., Santos-Rodriguez, R., & De Bie, T. (2013). Understanding effects of subjectivity in measuring chord estimation accuracy. *IEEE Transactions on Audio, Speech, and Language Processing*, *21*(12), 2607–2615. [Cited on page 48.]

Niedermayer, B. & Widmer, G. (2010). A multi-pass algorithm for accurate audio-to-score alignment. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 417–422. [Cited on page 144.]

Noll, A. M. (1967). Cepstrum Pitch Determination. *Journal of the Acoustical Society of America*, *41*(2), 293. [Cited on pages 7 and 17.]

Ozerov, A., Philippe, P., Bimbot, F., & Gribonval, R. (2007). Adaptation of bayesian models for single-channel source separation and its application to voice/music separation in popular songs. *IEEE Transactions on Audio, Speech, and Language Processing*, *15*(5), 1564–1578. [Cited on page 18.]

Ozerov, A., Vincent, E., & Bimbot, F. (2012). A general flexible framework for the handling of prior information in audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*(4), 1118–1133. [Cited on page 15.]

P. Grosche, P., Schuller, B., Müller, M., & Rigoll, G. (2012). Automatic transcription of recorded music. *Acta Acustica united with Acustica*, *98*(2), 199–215. [Cited on page 23.]

Paiva, R., Mendes, T., & Cardoso, A. (2006). Melody detection in polyphonic musical signals: Exploiting perceptual rules, note salience, and melodic smoothness. *Computer Music Journal*, *30*(4), 80–98. [Cited on pages 4, 18, 22, and 30.]

Pätynen, J., Pulkki, V., & Lokki, T. (2008). Anechoic recording system for symphony orchestra. *Acta Acustica united with Acustica*, *94*(6), 856–865. [Cited on pages 11 and 147.]

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830. [Cited on page 34.]

Pertusa, A. & Iñesta, J. M. (2005). Polyphonic monotimbral music transcription using dynamic networks. *Pattern Recognition Letters*, *26*(12), 1809–1818. [Cited on page 30.]

Piston, W. (1948). *Harmony*. Norton. [Cited on page 6.]

Piszczalski, M. & Galler, B. A. (1979). Predicting musical pitch from component frequency ratios. *The Journal of the Acoustical Society of America*, *66*(3), 710–720. [Cited on page 17.]

Poliner, G., Ellis, D., Ehmann, A., Gómez, E., Streich, S., & Ong, B. (2007). Melody transcription from music audio: Approaches and evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, *15*(4), 1247–1256. [Cited on pages 2, 5, 15, and 48.]

Poliner, G. E. & Ellis, D. P. W. (2005). A classification approach to melody transcription. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 161–166. [Cited on pages 19, 20, and 22.]

Rao, V., Gupta, C., & Rao, P. (2013). Context-aware features for singing voice detection in polyphonic music. In *Proceedings of the 9th International Conference on Adaptive Multimedia Retrieval: Large-scale Multimedia Retrieval and Evaluation (AMR'11)*, pp. 43–57. Springer-Verlag. [Cited on page 36.]

Rao, V. & Rao, P. (2009). Improving polyphonic melody extraction by dynamic programming based dual f0 tracking. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*. [Cited on page 34.]

Rigaud, F. & Radenen, M. (2016). Singing voice melody transcription using deep neural networks. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 737–743. [Cited on pages 19, 22, 28, 36, and 139.]

Ringer, A. L. (2017). Melody. grove music online, oxford music online. http://www.oxfordmusiconline.com/subscriber/article/grove/music/18357. (Date last accessed Jan 2017). [Cited on page 3.]

Rizo, D., Ponce, P. J., Pérez-Sancho, C., Pertusa, A., & Iñesta, J. (2006). A pattern recognition approach for melody track selection in midi files. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 61–66. [Cited on page 51.]

Roose, H. (2008). Many-voiced or unisono? an inquiry into motives for attendance and aesthetic dispositions of the audience attending classical concerts. *Acta Sociologica*, *51*(3), 237–253. [Cited on page 9.]

Ross, M., Shaffer, H., Cohen, A., Freudberg, R., & Manley, H. (1974). Average magnitude difference function pitch extractor. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *22*(5), 353–362. [Cited on page 17.]

Ryynänen, M. & Klapuri, A. (2008). Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, *32*(3), 72–86. [Cited on pages 22, 31, and 36.]

Ryynanen, M. P. & Klapuri, A. (2005). Polyphonic music transcription using note event modeling. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 319–322. IEEE. [Cited on page 23.]

Sakoe, H. & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, *26*(1), 43–49. [Cited on page 55.]

Salamon, J. (2013). *Melody Extraction from Polyphonic Music Signals*. Ph.D. thesis, Universitat Pompeu Fabra. [Cited on page 5.]

Salamon, J. & Gómez, E. (2012). Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio. Speech. Lang. Processing*, *20*(6), 1759–1770. [Cited on pages 5, 14, 18, 19, 20, 22, 24, 25, 28, 30, 31, 32, 33, 36, 42, 47, 54, 55, 64, 87, 88, 89, 90, 91, 93, 94, 97, 98, 99, 100, 108, 109, 113, 130, 142, 160, and 161.]

Salamon, J., Gómez, E., & Bonada, J. (2011). Sinusoid extraction and salience function design for predominant melody estimation. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pp. 73–80. [Cited on pages 25, 37, and 93.]

Salamon, J., Gómez, E., Ellis, D., & Richard, G. (2014). Melody Extraction from Polyphonic Music Signals: Approaches, applications, and challenges. *IEEE Signal Processing Magazine*, *31*, 118–134. [Cited on pages 1, 2, 4, 18, 42, 43, and 73.]

Salamon, J., Peeters, G., & Röbel, A. (2012a). Statistical characterisation of melodic pitch contours and its application for melody extraction. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 187–192. [Cited on pages 32, 33, 98, and 108.]

Salamon, J., Rocha, B., & Gómez, E. (2012b). Musical genre classification using melody features extracted from polyphonic music signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 81–84. [Cited on page 32.]

Salamon, J., Serra, J., & Gómez, E. (2013). Tonal representations for music retrieval: from version identification to query-by-humming. *International Journal of Multimedia Information Retrieval*, *2*(1), 45–58. [Cited on pages 2 and 32.]

Salamon, J. & Urbano, J. (2012). Current Challenges in the Evaluation of Predominant Melody Extraction Algorithms. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 289–294. [Cited on pages 42 and 79.]

Schedl, M., Gómez, E., & Urbano, J. (2014). Music information retrieval: Recent developments and applications. *Foundations and Trends in Information Retrieval*, *8*, 127–261. [Cited on page 16.]

Schlüter, J. & Grill, T. (2015). Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 121–126. [Cited on page 20.]

Selfridge-Field, E. (1998). Conceptual and representational issues in melodic comparison. *Computing in musicology: a directory of research*, *11*(1), 3–64. [Cited on pages 1 and 48.]

Sigtia, S., Benetos, E., & Dixon, S. (2016). An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *24*(5), 927–939. [Cited on page 30.]

Simon, L. S. & Vincent, E. (2012). A general framework for online audio source separation. In *Proceedings of the International conference on Latent Variable Analysis and Signal Separation*, pp. 397–404. Springer. [Cited on page 16.]

Simonton, D. (1984). Melodic structure and note transition probabilities: A content analysis of 15,618 classical themes. *Psychology of Music*, *12*(1), 3–16. [Cited on page 50.]

Smaragdis, P. (2004). Discovering auditory objects through non-negativity constraints. In *Proceedings of the Statistical and Perceptual Audio Processing (SAPA)*. [Cited on page 26.]

Smaragdis, P. & Brown, J. (2003). Non-negative matrix factorization for polyphonic music transcription. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 177–180. IEEE. [Cited on page 26.]

Smaragdis, P., Raj, B., & Shashanka, M. (2006). A probabilistic latent variable model for acoustic modeling. *Advances in models for acoustic processing, NIPS*, *148*. [Cited on page 26.]

Smith, J., Burgoyne, J., Fujinaga, J., De Roure, D., & Downie, J. (2011). Design and creation of a large-scale database of structural annotations. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, vol. 11, pp. 555–560. [Cited on page 48.]

Solomon, L. (2017). Melody,glossary of technical musical terms. http://web.archive.org/web/20080207010024. (Date last accessed Jan 2017). [Cited on page 4.]

Su, L. & Yang, Y.-H. (2015). Combining spectral and temporal representations for multipitch estimation of polyphonic music. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, *23*(10), 1600–1612. [Cited on pages 16 and 23.]

Tachibana, H., Ono, T., Ono, N., & Sagayama, S. (2010). Melody line estimation in homophonic music audio signals based on temporal-variability of melodic source. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 425–428. IEEE. [Cited on pages 18 and 36.]

Talkin, D. (1995). A robust algorithm for pitch tracking (RAPT). *Speech coding and synthesis*, *495*, 518. [Cited on page 17.]

Terhardt, E. (1977). The two-component theory of musical consonance. *Psychophysics and physiology of hearing*, pp. 381–390. [Cited on page 6.]

Terhardt, E. (1979). Calculating virtual pitch. *Hearing research*, *1*(2), 155–182. [Cited on page 17.]

Terhardt, E., Stoll, G., & Seewann, M. (1982). Algorithm for extraction of pitch and pitch salience from complex tonal signals. *The Journal of the Acoustical Society of America*, *71*(3), 679–688. [Cited on page 17.]

Tolonen, T. & Karjalainen, M. (2000). A computationally efficient multipitch analysis model. *IEEE Transactions on Speech and Audio Processing*, *8*(6), 708–716. [Cited on page 21.]

Urbano, J., Marrero, M., & Martín, D. (2013). On the measurement of test collection reliability. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pp. 393–402. New York, NY, USA: ACM. [Cited on pages 78 and 79.]

Verma, P. & Schafer, R. W. (2016). Frequency estimation from waveforms using multi-layered neural networks. In *Proceedings of Interspeech*, pp. 2165–2169. [Cited on pages 19 and 22.]

Vincent, E., Gribonval, R., & Fevotte, C. (2006). Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, *14*(4), 1462–1469. [Cited on page 107.]

Vinyes, M., Bonada, J., & Loscos, A. (2006). Demixing commercial music productions via human-assisted time-frequency masking. In *Proceedings of the Audio Engineering Society 120th Convention*. [Cited on pages 16 and 115.]

Virtanen, T. (2007). Monaural Sound Source Separation by Nonnegative Matrix Factorization With Temporal Continuity and Sparseness Criteria. *IEEE Transactions on Audio, Speech, and Language Processing*, *15*(3), 1066–1074. [Cited on page 26.]

Von Hippel, P. (2000). Redefining pitch proximity: Tessitura and mobility as constraints on melodic intervals. *Music Perception*, pp. 315–327. [Cited on page 50.]

Walmsley, P. J., Godsill, S. J., & Rayner, P. J. (1999). Bayesian modelling of harmonic signals for polyphonic music tracking. In *Proceedings of the Cambridge Music Processing Colloquium*, vol. 30. [Cited on page 21.]

Wang, C.-C., Fan, Z.-C., Jang, J.-S. R., & Yoshii, K. (2016). Mirex2016: Audio melody extraction using neural network. *Music Information Retrieval Evaluation eXchange (MIREX)*. [Cited on page 42.]

Wu, J., Vincent, E., Raczyński, S., Nishimoto, T., Ono, N., & Sagayama, S. (2011). Polyphonic pitch estimation and instrument identification by joint modeling of sustained and attack sounds. *IEEE Journal of Selected Topics in Signal Processing*, *5*(6), 1124–1132. [Cited on page 35.]

Yeh, C. (2008). *Multiple fundamental frequency estimation of polyphonic recordings*. Ph.D. thesis, Université Lille 1. [Cited on page 21.]

Yeh, C., Röbel, A., & Rodet, X. (2010). Multiple fundamental frequency estimation and polyphony inference of polyphonic music signals. *IEEE Transactions on Audio, Speech, and Language Processing*, *18*(6), 1116–1126. [Cited on page 37.]

Yeh, T., Wu, M., Jang, J., Chang, W., & Liao, I. (2012). A hybrid approach to singing pitch extraction based on trend estimation and hidden markov models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 457–460. IEEE. [Cited on pages 19 and 22.]

Zapata, J. R., Davies, M., & Gómez, E. (2014). Multi-feature beat tracking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *22*, 816 – 825. [Cited on pages 53, 55, and 65.]

Zatorre, R. J., Evans, A. C., & Meyer, E. (1994). Neural mechanisms underlying melodic perception and memory for pitch. *Journal of Neuroscience*, *14*(4), 1908–1919. [Cited on page 4.]

Zhang, J. X., Christensen, M. G., Jensen, S. H., & Moonen, M. (2012). Joint doa and multi-pitch estimation based on subspace techniques. *EURASIP Journal on Advances in Signal Processing*, *2012*(1), 1. [Cited on page 115.]