Flamenco Music Information Retrieval

Automatic Content-Based Description of Flamenco Music Collections

Nadine Kroher

Supervisor: José-Miguel Díaz-Báñez

Department of Applied Mathematics II
University of Seville

Doctoral thesis

July 2018

Acknowledgements

They say it takes a village to raise a child. As it turns out, it took an army of colleagues and collaborators, a supportive bunch of friends and loved ones, a number of highly skilled bureaucracy experts and an incredibly patient and tolerant supervisor to finish this PhD thesis. I thank all of you and it is needless to say that none of this would have been possible without you.

First of all, I would like to express my gratitude to my supervisor José-Miguel Díaz-Báñez whose efforts went far beyond the necessary. Apart from being a great supervisor, he has been extremely helpful and supportive throughout the past years and has helped me understand and appreciate the "flamenco lifestyle".

A large part of this thesis is the result of collaborations and discussions with researchers and aficionados from different research areas, universities and countries. I would like to thank all of you for sharing your expertise, advise and valuable feedback. I also thank my colleagues and the administrative staff at the Department of Applied Mathematics II, who were immensely helpful in resolving all of my academic, bureaucratic and teaching-related questions.

Last but not least, I thank Karin and Juergen for their unconditional support, my friends of the "ONU" for dragging me away from the desk now and then, and of course my partner, for his love, care and eternal patience.

The research conducted in the scope of this thesis has received funding from the projects COFLA2 (Junta de Andalucía, P12- TIC-1362), GALGO (Spanish Ministry of Economy and Competitiveness, MTM2016- 76272-R AEI/FEDER,UE), from the European Union Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 734922 and from the international mobility grant of the University of Seville (plan propio de investigación y transferencia 2017-1.3-A) .

Willen braucht man.
Und Zigaretten.

HELMUT SCHMIDT

Abstract

Flamenco is a rich performance-oriented art music genre from Southern Spain, which attracts a growing community of aficionados around the globe. The constantly increasing number of digitally available flamenco recordings in music archives, video sharing platforms and online music services calls for the development of genre-specific description and analysis methods, capable of automatically indexing and examining these collections in a content-driven manner.

Music Information Retrieval is a multi-disciplinary research area dedicated to the automatic extraction of musical information from audio recordings and scores. Most existing approaches were however developed in the context of popular or classical music and do often not generalise well to non-Western music traditions, in particular when the underlying music theoretical assumptions do not hold for these genres. The specific characteristics and concepts of a music tradition can furthermore imply new computational challenges, for which no suitable methods exist.

This thesis addresses these current shortcomings of Music Information Retrieval by tackling several computational challenge which arise in the context of flamenco music. To this end, a number of contributions to the field are made in form of novel algorithms, comparative evaluations and data-driven studies, directed at various musical dimensions and encompassing several sub-areas of computer science, computational mathematics, statistics, optimisation and computational musicology. A particularity of flamenco, which immensely shapes the work presented in this thesis, is the absence of written scores. Consequently, computational approaches can solely rely on the direct analysis of raw audio recordings or automatically extracted transcriptions, and this restriction generates set of new computational challenges.

A key aspect of flamenco is the presence of reoccurring melodic templates, which are subject to heavy variation during performance. From a computational perspective, we identify three tasks related to this characteristic - melody classification, melody retrieval and melodic template extraction - which are addressed in this thesis. We furthermore approach the task of detecting repeated sung phrases in an unsupervised manner and explore the use of deep learning methods for image-based singer identification

in flamenco videos and structural segmentation of flamenco recordings. Finally, we demonstrate in a data-driven corpus study, how automatic annotations can be mined to discover interesting correlations and gain insights into a largely undocumented genre.

Resumen

El flamenco, un género musical centrado en la improvisación y la espontaneidad, tiene su origen en el sur de España y atrae a una creciente comunidad de aficionados de países de todo el mundo. El aumento constante y la accesibilidad a colecciones digitales de flamenco, en archivos de música y plataformas online, exige el desarrollo de métodos de análisis y descripción computacionales con el fin de indexar y analizar el contenido musical de manera automática.

Music Information Retrieval (MIR) es un área de investigación multidisciplinaria dedicada a la extracción automática de información musical desde grabaciones de audio y partituras. Sin embargo, la gran mayoría de las herramientas existentes se dirigen a la música clásica y la música popular occidental y, a menudo, no se generalizan bien a las tradiciones musicales no occidentales, particularmente cuando las suposiciones relacionadas con la teoría musical no son válidas para estos géneros. Por otro lado, las características y los conceptos musicales específicos de una tradición musical pueden implicar nuevos desafíos computacionales, para los cuales no existen métodos adecuados.

Esta tesis enfoca estas limitaciones existentes en el área abordando varios desafíos computacionales que surgen en el contexto de la música flamenca. Con este fin, se realizan una serie de contribuciones en forma de algoritmos novedosos, evaluaciones comparativas y estudios basados en datos, dirigidos a varias dimensiones musicales y que abarcan varias subáreas de ingeniería, matemática computacional, estadística, optimización y musicología computacional. Una particularidad del género, que influye enormemente en el trabajo presentado en esta tesis, es la ausencia de partituras para el cante flamenco. En consecuencia, los métodos computacionales deben basarse únicamente en el análisis de grabaciones, o de transcripciones extraídas automáticamente, lo que genera una colección de nuevos problemas computacionales.

Un aspecto clave del flamenco es la presencia de patrones melódicos recurrentes, que esán sujetos a variación y ornamentación durante su interpretación. Desde la perspectiva computacional, identificamos tres tareas relacionadas a esta característica que se abordan en esta tesis: la clasificación por melodía, la búsqueda de secuencias melódicas y la extracción de patrones melódicos. Además, nos acercamos a la tarea

de la detección no supervisada de frases melódicas repetidas y exploramos el uso de métodos de deep learning para la identificación de cantaores en grabaciones de video y la segmentación estructural de grabaciones de audio. Finalmente, demostramos en un estudio de minería de datos, cómo una exploración de anotaciones extraídas de manera automática de un corpus amplio de grabaciones nos ayuda a descubrir correlaciones interesantes y asimilar conocimientos sobre este género mayormente indocumentado.

Table of contents

Chapter 1

Introduction

1.1 Motivation

In recent years, the trend towards constantly growing collections of digitally available audio recordings has expanded beyond the scope of commercial popular and classical music. Digital libraries specialising on folk music and non-Western music traditions, like the Meertens Tune Collection[1], the Irish Traditional Music Archive[2] and the Andalusian Centre for Documentation of Flamenco Music[3], provide researchers, students and music enthusiasts with hundreds of thousands of recordings. The increasing number of available digital items poses new computational challenges for their indexing, exploration, curation and query. Furthermore, initiatives to safeguarding digital records related to cultural heritage have shifted their focus from the physical preservation of the sources to ensuring the localisability of items among overwhelming amounts of available data.

[1] http://www.liederenbank.nl/mtc/
[2] https://www.itma.ie
[3] http://www.centroandaluzdeflamenco.es

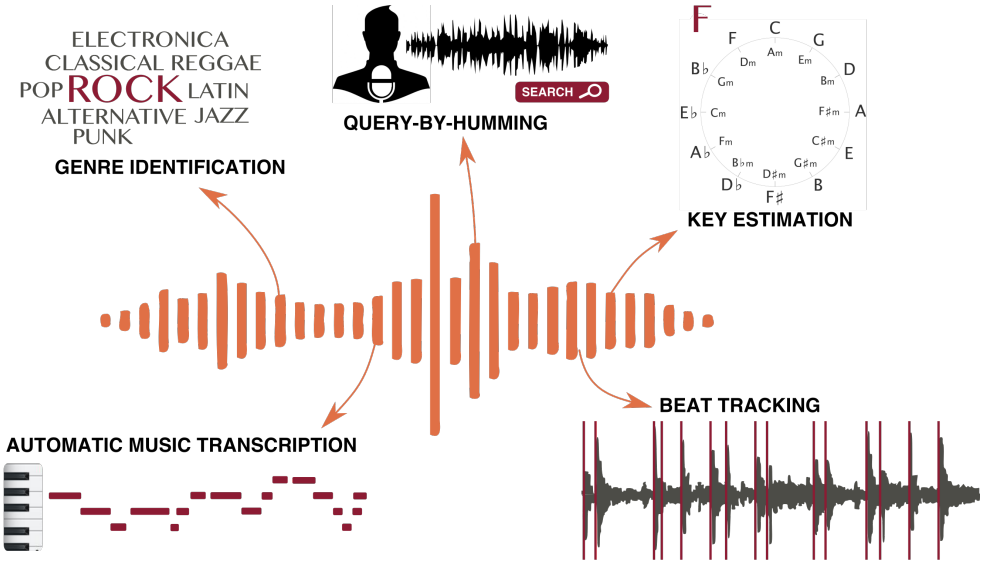


Fig. 1.1 Illustration of examples of MIR tasks.

Since the 1990s, the Music Information Retrieval (MIR) community is dedicated to the development of algorithms which can extract high-level information from audio recordings and musical scores [50]. Such systems are fundamental to the creation of tools which automatically index large music collections and enable users to explore their content in efficient and semantically meaningful ways. Commonly addressed MIR tasks include, but are not limited to, beat tracking [112], automatic music transcription [8], key estimation [154], query-by-humming [66] and genre classification [206] (Figure 1.1).

While many MIR methods have shown to yield reliable performance on real-world data and several have made their way into commercial applications, most approaches have been developed in the context of Western popular and classical music and do not necessarily generalise to other, in particular non-Western, genres. These limitations are mainly due to musical assumptions, on which the algorithms rely, but which do not hold for other music traditions. An example is automatic music transcription, the task of approximating a score-like representation from an audio signal. While state-of-the-art systems usually target pitch values quantised to the equal tempered scale, many non-Western music traditions, for example Indian Art Music or Turkish Makam, follow different tuning systems. Similarly, key estimation systems usually target major and minor tonality, the modes typically encountered in popular and classical music. However, non-Western music traditions may be based on different tonal organisations. Another example is the well-studied problem of musical genre classification. While it has been shown in [176], that flamenco can be automatically

identified among other music genres, the somewhat related problem of automatically classifying flamenco styles remains unresolved [100]. In addition, methods which rely on a symbolic representation, are often not applicable to oral music traditions where scores are generally not available.

On the other hand, it should be mentioned that genre-specific methods can often yield higher performance and improved reliability compared to generic systems when particular characteristics of a tradition, i.e. limited instrumentation, are appropriately formalised. In [8] it is for example mentioned, that genre-specific music transcription systems do generally yield better results compared to generic methods. The particular characteristics of a genre may furthermore give rise to new computational challenges. An example, which applies to several genres, including flamenco [101], Carnatic [164], Arab-andalusian [99] and Sizhu [200] music, is the existence of reoccurring melodic templates which are of essential importance for a meaningful systematic classification of recordings. Similarly, the concept of tune families in European folk music [31] refers to groups of songs which are assumed to originate from common ancestral melodies. Consequently, the classification of recordings according to their underlying melody is essential to the automatic organisation of such collections. Another example is found in Indian art music, where the tuning of vocals and instruments depends on the tonic of the performed piece. Consequently, the automatic identification of the tonic is an important research topic in the field, since its is fundamental to a number of related MIR tasks [77], such as transcription and pattern detection.

In addition to their use in organisation end exploration of digital collections, genre-specific MIR-based tools are of major importance for musicological studies [213]. Automatic analysis methods enable musicologists to scale descriptive studies from a few manually analysed examples to large collections [10, 28]. MIR tools provide the means to empirically validate musicological hypotheses on large corpora with minimal manual intervention, and the formulation of computational models itself can furthermore give rise to new hypotheses and, in this way, directly contribute the discovery of knowledge [71].

During recent years, the gap between mainstream and non-Western music traditions has received growing acknowledgement in the MIR community and a number of initiatives and research projects target the development of genre-specific systems [182]: The objective of the *CompMusic*[4] project was the development of MIR tools for Carnatic, Hindustani, Makam, Jingju and Arab-Andalusian music. Several methods

[4]http://compmusic.upf.edu

developed in the scope of the project have been implemented in an online platform[5]. The *WITCHCRAFT*[6] project focused on the content-based retrieval of folk song melodies and established an online melody-based search engine for a large collection of Dutch folk songs[7].

This thesis has been conducted in the context of the COFLA research project[8] and its main objective is the development and evaluation of genre-specific MIR methods for flamenco music. Due to its improvisational and expressive nature, its unique musical characteristics, and the fact that the genre is largely undocumented, flamenco poses a number of interesting computational challenges. At the same time, its growing popularity, constant evolution and increasing presence in digital media calls for the development genre-specific computational tools to manage existing digital content, to allow a broad range of users access to the genre and to contribute to its preservation.

Below, a short introduction to the flamenco music tradition is given and prior approaches to its computational study are discussed. This chapter is concluded with a summary of the contributions of this thesis, a note on research reproducibility and an outline of the remainder of the manuscript.

1.2 Flamenco Music

Flamenco is a rich oral music tradition from the Southern Spanish province of Andalucía, which attracts a growing community of aficionados around the world. Since its creation, which is speculated to date back to the late 18th century [18], flamenco has evolved from its folkloric origin into an elaborate art form. In the course of history, immigrants with a great variety of cultural backgrounds arrived at the harbours of Andalucía and settled in the surrounding areas. Consequently, it can be assumed that the flamenco tradition unifies a number of diverse musical influences, which have shaped the unique sound of the genre [225, 63].

As of today, flamenco forms an essential part of local culture and society. In Andalucía, as well as in other parts of Spain, it is common practice that folkloric and religious events are accompanied by flamenco performances [125], and local associations, which are dedicated to its preservation, organise regular concerts and conferences. Outside the professional scope, flamenco is frequently practiced in private gatherings and represents an essential element of Andalusian lifestyle [152]. Since the year 2000,

[5] http://dunya.compmusic.upf.edu
[6] http://www.cs.uu.nl/research/projects/witchcraft/
[7] http://www.liederenbank.nl/index.php?wc=true
[8] www.cofla-project.com

Fig. 1.2 Left: Monument of singer *Camarón de la Isla* in La Línea de la Concepción. Right: Typical performance setting in a flamenco association.

the genre has been inscribed in the UNESCO Representative List of the Intangible Cultural Heritage of Humanity[9], thus acknowledging its importance to cultural identity and the need for safeguard. Beyond its local popularity, flamenco is increasingly gaining appreciation abroad and has a growing impact on the Spanish economy. It is estimated, that in the year 2004, more than half a million visitors travelled to Andalucía to experience flamenco music first hand, generating an income of over 550 million Euros in the local tourist sector[10].

From a musical viewpoint, flamenco can be described as a highly expressive art form with a strong improvisational character [72]. Having evolved from a singing tradition, the voice remains the central element of flamenco, usually accompanied by the guitar, dance, and rhythmic hand-clapping. The unique sound of flamenco singing originates from the presence of extensive melismatic ornamentation, vibrato and grace notes, sudden dynamic changes and timbral and rhythmic discontinuities. While both, voice and guitar, generally follow the diatonic scale, micro-tonal embellishments frequently occur in the singing voice, causing an often ambiguous tonal perception.

In [72], flamenco is furthermore described as an *eminently individual yet highly structured form of music*, referring to the fact that the genre is based on a set of prototypical structures which are subject to individual interpretation during improvisational performances. Flamenco is organised in a hierarchy of style families, styles and

[9] http://www.unesco.org/culture/ich/en/lists

[10] http://www.juntadeandalucia.es/turismocomercioydeporte/publicaciones/19631.pdf

sub-styles (Figure 1.3), where each category is defined by a combination of melodic, harmonic and rhythmic properties [102]. These include, for example, chord sequences and rhythmic accentuation patterns. However, these structures are by no means static rules, but rather represent a basic musical skeleton, which leaves much room for individual expression and improvisation. The granularity of the pre-defined structures furthermore varies among styles. While some are merely defined by a rhythmic pattern and any chord progression and tonality can be used, others impose a basic harmonic progression and a fixed tonal organisation. An interesting aspect of flamenco styles is the fact that some categories at the lowest hierarchical level are defined by a common melodic template, which is subject to ornamentation and variation during performance. Styles are essential to the organisation of flamenco music collections, since a particular performance is often referred to by its style affinity, whereas song titles are less commonly used. Editorial meta-data does therefore often include style annotations. However, it should be mentioned that, while a number of classification schemes have been proposed, there does not exist a universally accepted taxonomy and editorial meta information varies with respect to the level of provided detail.

An important aspect of flamenco, and its analysis, is the fact that it is an oral music tradition, where knowledge of styles and musical concepts has been passed on implicitly throughout generations. This impacts the systematic study of the genre in two ways. First, since songs are taught and performed from memory, flamenco does generally not rely on written scores. In fact, the few existing manual annotations were mostly produced in the context of musicological studies and refer to a particular performance rather than the underlying musical template. Secondly, despite recent efforts to formalise flamenco music theory [57, 123], which in number and depths are modest compared to existing work on jazz, pop and classical music, many of its aspects remain implicit and consequently undocumented.

Furthermore, it can be said that flamenco undergoes a constant evolution and a number of recent trends have evolved during the past decades [189]. These include the incorporation of new instruments, for example flute and piano, as well as fusion with other genres. However, this thesis focuses on classical flamenco, a well-defined concept among flamenco experts, which refers to a set of established and renowned artists of the 20th century. Recent variants of flamenco are excluded, given their constant evolution and volatile appearance and disappearance.

For a complete guide to flamenco, its history and its musical characteristics, the interested reader is referred to the books by *Castro Buendía* [22], *Granados* [75], *Worms* [229] and *Nuñez and Gamboa* [64].

Fig. 1.3 Two illustrations of the hierarchical style organisation. Left: Artwork displayed in the flamenco association *El Melón de Oro* near Murcia. Right: Illustration by Rafael Jofré [193].

1.3 Related Work

1.3.1 Mathematics and music

Mathematics and music have crossed their paths since Pythagorean times [217] and multidisciplinary studies involving the two fields frequently attract the attention of both research communities [220]. Such approaches aim at deriving mathematical models of music-theoretical concepts (i.e. [201, 74]), human perception of music [3], musical acoustics [168] and the process of composing music [231]. Formal music theory as well as computational studies have furthermore inspired theoretical problems in various mathematical disciplines [9]. For a comprehensive introduction to the topic, we refer the interested reader to [230].

Flamenco, despite its expressive and unpredictable appearance, poses no exception to the ongoing partnership between mathematics and music and the genre has recently been the focus of mathematical research. In particular, musicological and computational

work on rhythmic similarity and the commonly used representation of flamenco rhythms as polygons have given rise to a number of problems in the field of computational geometry (see i.e. [2, 5, 42]). A comprehensive elaboration on the synergies between mathematics and flamenco, can be found in [38] and [39].

1.3.2 The COFLA project

The relatively young field of computational analysis of flamenco music has been explored in the scope of the *COFLA*[11] research project. Over the recent decade, a number of computational problems related to flamenco have been defined and several methods addressing a variety of musical aspect have been proposed.

A fundamental necessity for computational studies is the **creation of data corpora**, in our case collections of music recordings, which are not only the basis for experimental evaluation, but are furthermore a valuable resource for data-driven exploratory studies. In the context of flamenco music, there are two well-established datasets, which, in addition to newly created collections for particular applications, are used in this thesis. The *TONAS*[12] dataset contains 72 recordings of songs belonging to the *tonás* style family, together with their respective automatic and manual note-level transcriptions. The included meta-data furthermore contains information about the style-affinity of the recordings, which belong either to the *debla* or the *martinete* style. The second important dataset in the context of flamenco music is the *corpusCOFLA* [102], a collection of over 1500 commercial flamenco recordings taken from renown anthologies. This corpus has been developed specifically for computational studies and can be considered a representative sample of classical flamenco. In addition to the audio recordings, the corpus contains editorial and manually curated meta-data, as well as several manually annotated subsets. In addition to audio collections, the knowledge base *FlaBase* [149], is a valuable resource for semantic data, including biographical and music-theoretical knowledge gathered from online resources.

An important line of research in the field has focused on the **automatic transcription of flamenco singing**. The aim of this task is to extract a symbolic representation of the singing voice melody from flamenco recordings. While such score-like representations are of little use for practicing musicians, they are fundamental to a number of related MIR tasks and furthermore represent a valuable tool for educational purposes and musicological studies. Given the pitch-continuous nature of the singing voice and the presence of micro-tonal phenomena, including pitch glides and vibrato, note-level

[11] http://www.cofla-project.com
[12] https://www.upf.edu/web/mtg/tonas

transcriptions can be seen as a simplification of the underlying melodic contour, in which a certain amount of detail is omitted. For many computational approaches, this type of representation is of advantage, since it captures the melodic essence and, at the same time, allows efficient processing, in particular in large-scale applications. The transcription of flamenco singing is particularly challenging, given the presence of guitar accompaniment and the extensive use of micro-tonal ornamentation and fast melismatic note progressions. In addition, flamenco singers do generally not aim for perfect intonation, and even longer notes may be out of tune or unstable in pitch. A first approach towards computer-assisted transcription of a cappella flamenco recordings was proposed in [68] and later extended to recordings containing guitar accompaniment [70, 69]. Recently, the *CANTE* algorithm, a novel method for the transcription of flamenco singing from monophonic and polyphonic recordings has been proposed [104], which has shown to yield an improvement compared to [68]. The system is publicly available as a Python module[13] and used as a pre-processing stage for several algorithms described in this thesis.

A frequently addressed aspect of flamenco music is the existence of **melodic templates** which set the basis for improvisational performances. In particular, a number of styles and sub-styles are characterised by a specific melodic movement which undergoes heavy ornamentation and variation during individual interpretations. From a computational perspective, this characteristic is of fundamental importance to the automatic detection of styles. By recognising the template in a given performance, the style or sub-style affinity can be automatically annotated. In the context of content-based flamenco indexing, this task is referred to as **melody classification**. This is a unique computational task, due to the fact that in flamenco, the underlying template remains implicit and does not exist in form of a musical score. Therefore, the task has so far been approached as a similarity-based supervised classification scenario, based on the assumption that performances with a common template will exhibit high melodic similarity, while being dissimilar to performances based on other templates. Prior work on this topic [136, 45, 43, 135] has mainly focused on the a cappella styles belonging to the *tonás* family and a detailed discussion of existing techniques is provided in Chapter 2. In addition to the classification problem, melodic similarity in a cappella flamenco singing has been approached from a perceptual perspective. For the case of the *martinete* style, the mechanisms involved in human perception of melodic similarity have been investigated in [105] by computing the correlation between computational models and human similarity judgements.

[13] https://github.com/NadineKroher/PyCante

Another important problem in the context of content-based indexing is **melody retrieval**, the task of detecting a given melody or melodic sequence in a an audio collection. Such systems allow users to locate specific items based on their melodic content, an abstraction level which is not reflected in standard editorial meta-data. In the context of flamenco music, the detection of a manually defined melodic sequence is a non-trivial task due to the presence of ornamentation and embellishments. In a first approach towards detecting melodic patterns in flamenco recordings [157], manually defined note sequences are located within a corpus of accompanied recordings belonging to the *fandango* style. The system operates on a continuous representation of the predominant melody [175] and uses a modification of the context-dependent dynamic time-warping algorithm [159] to compensate for ornamentation and pitch estimation errors. The phenomenon of **ornamentation** in flamenco has furthermore itself been the subject of computational studies. In [73], pre-defined ornaments are detected in a corpus of note-level transcriptions using the Smith-Waterman [186] algorithm with a custom local similarity function. Focusing on flamenco melodies which have evolved from traditional popular chants, [126] extract ornaments by aligning ornamented flamenco performances to the basic melodic movement of the folkloric version. Another task related to melody retrieval is the **detection of repeated melodic patterns** within the same recording. While an extensive amount of work has focused on detecting repetition in sheet music [91], these approaches are not applicable to flamenco music due to the lack of written scores. Furthermore, the presence of solo guitar sections and the high amount of variation among repetitions of the same sung melodies pose additional challenges. A first audio-based approach to detecting melodic repetition in flamenco recordings was proposed in [106]. Sung phrases are detected using a vocal detection algorithm. Then, by computing pair-wise alignments of chroma-based representations, groups of similar phrases are formed using a frame-centric clustering scheme.

In addition to the work on the melodic domain described above, several attempts have been made to computationally model rhythm in flamenco music. In [41] and [40], mathematical measures were used to characterise **rhythmic similarity** between commonly occurring rhythmic patterns in flamenco music and to construct phylogenetic trees to visualise their relationship. Based on this tree representation, hypothetical ancestral rhythms were inferred [21]. This work was extended in [76], by comparing computational rhythmic similarity measures to human judgements.

With the aim of completing missing or ambiguous meta-information, another computational study has focused on **singer identification**. Analogously to face

recognition [236] in images and speaker identification [113] in spoken word recordings, singer identification is a well-studied problem in the MIR community (see i.e. [94, 131]). In [103], a method was developed to automatically recognise the singer in a flamenco recording. In addition to acoustic features, the machine learning approach uses statistical descriptors extracted from transcriptions to model a singer's individual performance style. This task is of importance for the organisation of flamenco collections, since in the flamenco world, singer names are not necessarily unique identifiers. Stage names may be used by various individuals of the same family and in audiovisual recordings, where the focus is on the dancer, the singer may not be annotated at all.

1.4 Contributions

1.4.1 Scientific scope

The work presented in this thesis has been conducted in a multi-disciplinary framework, characterised by a mutual interaction between musicology, applied mathematics and computer science. The research outcomes, which are novel technologies for the content-based description of flamenco recordings, are positioned in the field of computer science and use techniques from a variety of disciplines, including audio signal processing, computer vision, pattern detection and machine learning. Instead of focusing on a particular set of techniques, the methods were chosen based on the properties of the targeted problems. On the one hand, all of these fields strongly rely on advanced mathematical foundations. On the other hand, the development of computational methods for flamenco music, as well as music-theoretical concepts themselves, have given rise to new mathematical problems. These include the geometric problem described in Section 4.3, which is inspired by the presence of melodic templates in flamenco music, and the open problems mentioned at the end of Section 4.4.

On a different level, this work addresses the complex relationship between music and technology, and in particular the gap between often weakly defined, and to a certain extend unpredictable, musical characteristics and the need for strict definitions in computational models. To this end, a close collaboration with flamenco experts and musicologists is key to defining the music theoretical foundations which are essential in the development of computational approaches. In turn, computational models are valuable tools for large-scale musicological studies which aim to advance and deepen the knowledge of the genre. An example of such a data-driven approach to flamenco discovery is presented in Section 6.5.

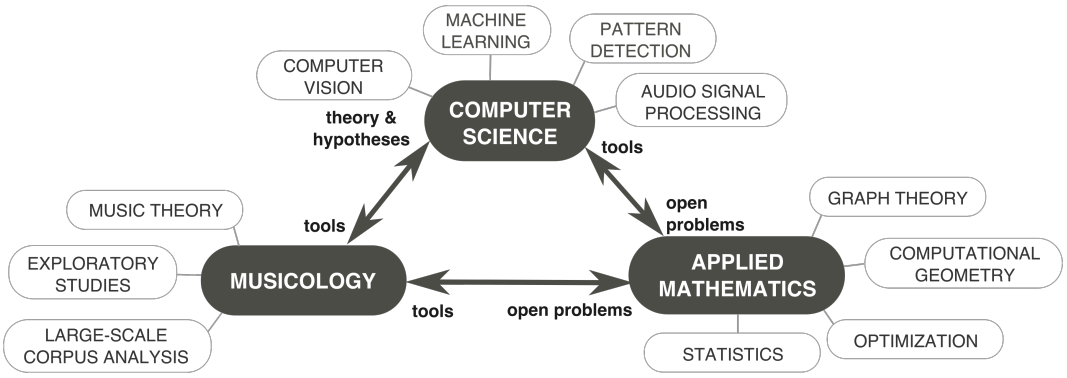


Fig. 1.4 Schematic illustration of the scientific scope.

It is furthermore worth noting, that while the main focus of this thesis is flamenco music, some of the proposed methods have shown to be applicable to genres with similar characteristics. More specifically, apart from flamenco music, Greek *rebetiko* (Chapter 5) and the Dutch folk song tradition (Chapters 2 and 5) are explored.

1.4.2 Applied mathematics in MIR

Research in applied mathematics generally aims at advancing knowledge by designing or improving methods and algorithms for specific problem statements. The problems themselves are in many cases of abstract nature, meaning that they are defined in the space of a mathematical model. Real-world problems can then be modelled mathematically and such abstract solutions can be applied to solve them. The proposed methods are commonly validated through mathematical proofs related to their correctness and computational complexity. While the work presented in this thesis follows the same objectives, namely the design, adaptation and improvement of methods for given problems, the problem formulations are strongly application-driven and directly relate to real-world tasks. In other words, the mathematical models arise from and are tied to real-world scenarios: Point-sets are an abstract representation of musical notes (i. e. Section 5), piece-wise linear functions model melodic contours (Section 4.3), graph edges model note transitions (Section 4.4), the presence of vocals given a short-term spectrogram is modelled as a non-convex optimisation problem (Chapter 6), etc.

This implies two ways in which large parts of this thesis differ from the research framework commonly followed in applied mathematics: (a) the spectrum of mathematical techniques and sub-areas considered (b) the way in which novel methods are assessed and validated. Research projects and theses in applied mathematics usually focus on a single or various closely related sets of techniques and develop methods

which can be applied to a variety of application areas. Here, the chosen techniques are application-driven and, as a consequence, a number of sub-areas of applied mathematics are encompassed:

- **Operational research** and **classification** (Chapter 2).
- **Graph theory** and **computational geometry** in the context of automatic melodic template extraction (Chapter 4).
- **Statistics** in the context of phrase segmentation (Chapter 5).
- **Dynamic programming** in the context of melody classification and retrieval (Chapters 2 and 3).
- **Probability** and **non-convex optimisation** in the context of machine learning [190] and data mining (Chapter 6).

Secondly, since the proposed methods target real-world problems, we mostly focus on experimental evidence for their evaluation instead of providing mathematical proofs. As a consequence, the format of this thesis largely differs from related publications in applied mathematics. An exception is the work presented in Section 4.3, which provides a solution to a more generic mathematical problem, which was inspired by related MIR research. In this case, the focus is shifted towards demonstrating mathematical correctness and theoretic complexity of the algorithm whereas the performance in an applied scenario plays a minor role.

1.4.3 Main contributions

This thesis encompasses approaches to several aspects of flamenco music and various novel methods for different content-based description and exploration applications are proposed. Below, the main contributions are listed on an item basis.

Chapter 2 explores different representations of flamenco melodies and evaluates their suitability in the context of **melody classification**. In particular, existing work on the topic is extended with the following main contribution:

1. A systematic study of signal representations and evaluation metrics in the context of melody classification in supervised and unsupervised scenarios.

The work presented in Chapter 3 addresses the related task of **melody retrieval**, where the following main contribution is made:

2. A novel method for the automatic retrieval of pre-defined melodic patterns, which, based on a modification of the *Needleman-Wunsch* sequence alignment algorithm, locates a given melodic sequence in a collection of automatic flamenco singing transcriptions.

Chapter 4 presents two methods for the **extraction of melodic templates**, a novel problem which has not been approached in prior work. The contributions are summarised as follows:

3. A geometric method which approximates the common underlying contour of a set of performance transcriptions as a continuous function.

4. A progressive sequence alignment algorithm, which yields a graphical model of the melodic variation occurring among performances of the same melody and allows us to extract a discrete note-level representation of the underlying template.

5. Methods to quantitative and qualitative assess local and global melodic variation among performances with a common underlying template.

The focus of Chapter 5 is on the **discovery of repetition** in flamenco music, where the following main contribution is presented:

6. A novel framework for the automatic detection of repeated sung phrases, which operates on automatic transcriptions and is robust to transcription errors and variation among repetitions of the same content.

Chapter 6 explores the application of **deep learning methods**, in particular convolutional neural networks, to the content-based description of flamenco music. In this context, the following contributions are reported:

7. An image-processing method, which, based on face recognition techniques, automatically identifies the singer in audio-visual flamenco performance recordings.

8. A structural segmentation method, which detects the presence of vocals, handclapping, strummed and picked guitar in flamenco recordings.

9. An exploratory corpus mining study of automatic structural annotations of a large collection of flamenco recordings.

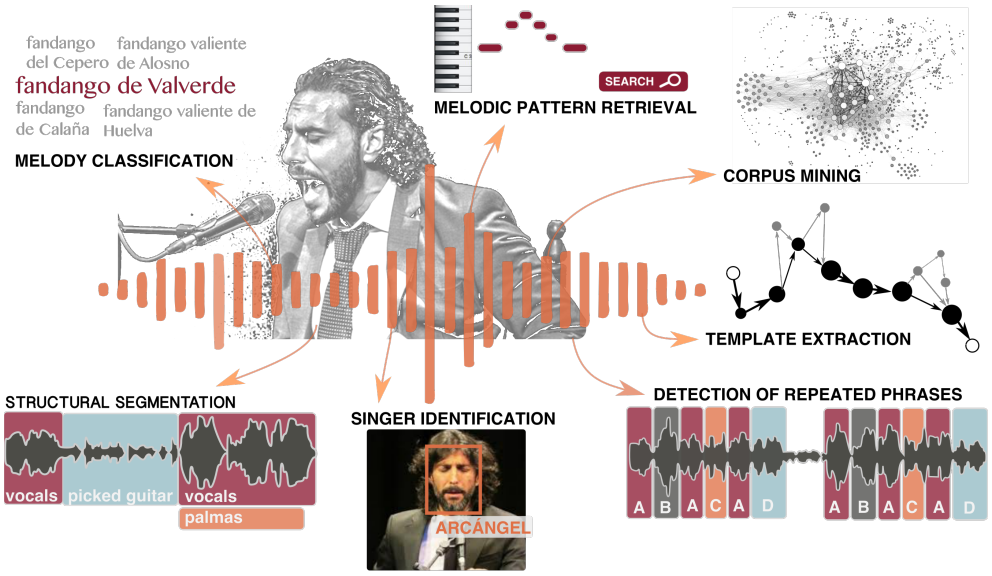


Fig. 1.5 Schematic illustration of the thesis contributions.

1.4.4 Related publications and collaborations

The main contributions of this thesis are based on the following publications, which are published or currently under review in peer-reviewed journals and conferences.

Journal articles

- N. Kroher and J. M. Díaz-Báñez (2017), Audio-based Melody Categorisation: Exploring Signal Representations and Evaluation Strategies. *Computer Music Journal*, 41(4), pp. 1–19. (Chapter 2)
- S. Bereg, J. M. Díaz-Báñez, N. Kroher and I. Ventura, Computing Melodic Templates in Oral Music Traditions. Submitted to *Applied Mathematics and Computation*. (Chapter 4)
- N. Kroher and J. M. Díaz-Báñez, Modelling melodic variation and extracting melodic templates from flamenco singing performances. Submitted to the *Journal of Mathematics and Music*. (Chapter 4)
- N. Kroher, A. Pikrakis and J. M. Díaz-Báñez (2017). Discovery of repeated melodic phrases in folk singing recordings. *IEEE Transactions on Multimedia*, 20(6), pp. 1291 - 1304. (Chapter 5)
- N. Kroher and A. Pikrakis. Exploratory Analysis of a Large Flamenco Corpus using a Convolutional Neural Network as a Structural Annotation Backend. Submitted to the *ACM Journal on Computing and Cultural Heritage*. (Chapter 6)

Peer-reviewed conference proceedings

- A. Pikrakis, N. Kroher and J. M. Díaz-Báñez (2016). Detection of Melodic Patterns in Automatic Transcriptions of Flamenco Singing. In Proceedings of the *6th International Workshop on Folk Music Analysis*, Dublin, Ireland, pp. 14–17. (Chapter 3)
- N. Kroher, A. Pikrakis and J.-M. Díaz-Báñez (2017). Image-based singer identification in flamenco videos. In Proceedings of the *7th International Workshop on Folk Music Analysis (FMA)*, Malaga, Spain, pp. 88–92. (Chapter 6)
- S. Bereg, J.-M. Díaz-Báñez, N. Kroher and I. Ventura (2017). Computing melodic templates in flamenco singing. In Proceedings of the *XVII Spanish Meeting on Computational Geometry*, Alicante, Spain, pp. 45–48. (Chapter 4)
- R. Piedra De La Cuadra, I. Ventura, J. M. Díaz-Báñez and N. Kroher (2018). Estimación de la Variación Melódica en el Cante Flamenco. In Proceedings of *Investigación y Flamenco (INFLA)*, Seville, Spain. (Chapter 4)

The following publications are related to, or inspired by, the work presented in this thesis, but are not considered in this manuscript.

- I. Morales, N. Kroher and J. M. Díaz-Báñez (2017). Melodic pattern cross-ocurrences between guitar falsetas and singing voice in flamenco music. In Proceedings of the *7th International Workshop on Folk Music Analysis (FMA)*, Malaga, Spain, pp. 119-120.
- I. Marqués, N. Kroher, J. Mora and J. M. Díaz-Báñez (2017). Extraction and classification of ornamentation in flamenco singing: An evolution-based approach In Proceedings of the *7th International Workshop on Folk Music Analysis (FMA)*, Malaga, Spain, pp. 121-122.
- R. de Valk, A. Volk, A. Holzapfel, A. Pikrakis, N. Kroher and J. Six (2017). MIRchiving: Challenges and opportunities of connecting MIR research and digital music archives. In Proceedings of the *4th International Digital Libraries for Musicology workshop (DLfM)*.
- A. Pikrakis, Y. Kopsinis, N. Kroher and J. M. Díaz-Báñez (2016). Unsupervised Singing Voice Detection Using Dictionary Learning. In Proceedings of the *European Signal Processing Conference (EUSIPCO)*, Budapest, Hungary, pp. 1212-1216.
- N. Kroher and J. M. Díaz-Báñez (2016). Towards Flamenco Style Recognition: the Challenge of Modelling the Aficionado. In Proceedings of the *6th International Workshop on Folk Music Analysis (FMA)*, Dublin, Ireland, pp. 61-63.
- L. E. Caraballo, J.-M. Díaz-Báñez and N. Kroher (2018). A Polynomial Algorithm for Balanced Clustering via Graph Partitioning. Submitted to the *European Journal of Operational Research*. arXiv preprint *arXiv:1801.03347*.

- A. Camacho Gil, J.-M. Díaz-Báñez and N. Kroher (2018). User-level tools for the study of flamenco. In Proceedings of *Investigación y Flamenco (INFLA)*, Seville, Spain.

The work presented in Chapters 3, 5 and 6 are results of a collaboration with Aggelos Pikrakis from the University of Piraeus. The method in Section 4.3 was developed in collaboration with Inmaculada Ventura from the University of Seville and Sergey Bereg from the University of Dallas at Texas.

1.5 Reproducibility

During the past decade, the MIR community has put more and more effort into fostering reproducibility of recent advances by making source code and datasets publicly available. In this way, researchers can reproduce the work of their peers, conduct a systematic evaluation of novel algorithms with respect to the state of the art and build upon the work of others rather than rewriting existing method from scratch. Making novel methods and algorithms reproducible is furthermore a step towards closing the gap between academic research and its incorporation in real-world industrial applications. It is worth mentioning at this point, that most research outcomes are made available in the form of source code and, with few exceptions, the development of user-level tools is beyond the scope most research projects.

Following these practices, a large part of the work presented in this thesis is based on existing implementations of state of the art algorithms and most of its main contributions are made publicly available for the sake of reproducibility. More specifically, the following open source datasets, tools and libraries are heavily used throughout this thesis:

- The *corpusCOFLA* and *TONAS* datasets which are available on the COFLA project website (www.cofla-project.com).
- The automatic flamenco singing transcription tool *CANTE* [104] (github.com/NadineKroher/PyCante).
- The *essentia* (essentia.upf.edu) and *librosa* (librosa.github.io) libraries for the extraction of low- and mid-level descriptors from audio recordings .
- The deep learning approaches presented in Chapter 6 rely on the open source libraries *keras* (keras.io), *opencv* (opencv.org), and *openface* (github.com/cmusatyalab/openface).

A list of links to source code and data repositories which allow reproducibility of the core contributions presented in this thesis is available at **https://nadinekroher.wordpress.com/thesis/**. This list will be updated in the future with material related to contributions which are currently pending publication.

1.6 Thesis outline

The remainder of this thesis is organised as follows. Chapter 2 presents a study on signal representations for the task of melody classification and in Chapter 3 a novel system for locating pre-defined melodic patterns in automatic transcriptions of flamenco music is proposed. The focus of Chapter 4 is on a new problem, the extraction of melodic templates from performance recordings, which was first approached in the scope of this thesis. In Chapter 5, a novel method for detecting repeated sung phrases in folk music recordings is presented. Chapter 6 explores the application of deep learning methods, in particular convolutional neural networks, to the automatic description of flamenco recordings and an exploratory mining study using automatic annotations is presented. Finally, the work is concluded in Chapter 7, where open problems and future work directions are discussed.

Chapter 2

Melody Classification

fandango valiente de Huelva
fandango del Cepero
fandango valiente de Alosno
fandango de Valverde
fandango de Calaña
fandango valiente de Huelva
fandango del Cabonerillo

2.1 Introduction

A defining feature of flamenco singing is the existence of reoccurring basic melodic progressions, which set the basis for improvisational performances. These melodic templates, which are performed from memory, have been passed implicitly from generation to generation without ever being formalised in any form of musical notation. In contrast to classical singing, where the sung note progression largely follows a written score, and expressivity is mainly introduced by means of dynamics, timing and accentuation, flamenco singers heavily embellish the underlying template through the insertion of grace notes and the use of melismatic ornamentation and rhythmic distortion. While flamenco aficionados can, despite the amount of melodic variation, easily identify performances based on the same template, this task is non-trivial for unaccustomed listeners [105]. It is worth noting, that flamenco experts commonly judge singing performances based on two criteria, the "correct" execution of the template and the virtuosity of the interpretation by means of variation, ornamentation and expressivity. Two very distinct performances of the same melody are shown in Figure 2.1.

An interesting aspect is the relation of melodic templates to the hierarchical style organisation of flamenco music. As mentioned in Chapter 1, categories situated at lower levels of the hierarchy are often defined by a common melodic skeleton. Consequently, automatically

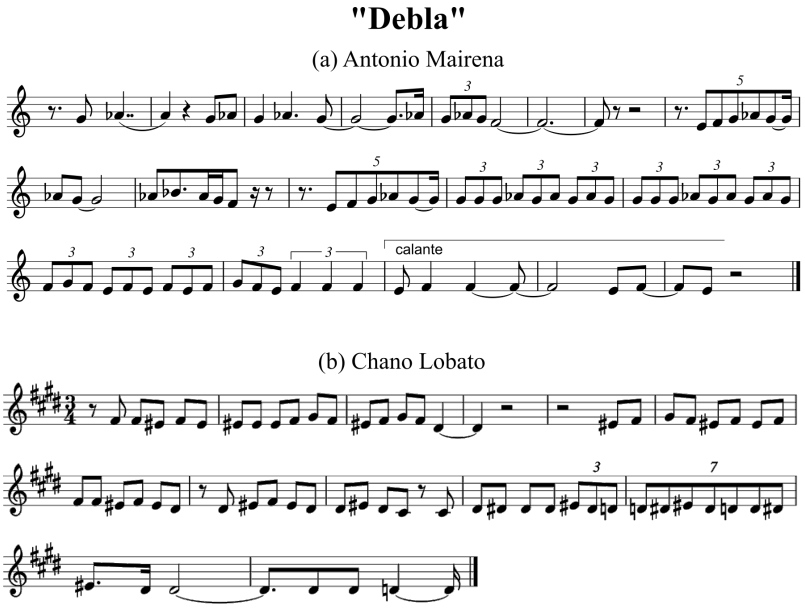


Fig. 2.1 Two performances belonging to the *debla* style. Top: Antonio Mairena. Bottom: Chano Lobato. Transcription: Joaquin Mora.

classifying a recording according to its underlying template can reveal its style affinity. As a result, melody classification is an important task for the automatic content-based organisation of flamenco music collections.

The amount of intra-style variation depends on the style itself. More specifically, styles which are close to their folkloric origin tend to show less deviation from the template compared to other, more interpretation-oriented styles. An example is shown in Figure 2.2: The three interpretations of the *fandango de Valverde* melody exhibit significantly less variation compared to the performances of the *debla* style shown in Figure 2.1. Another interesting aspect is the fact that the templates themselves exhibit a hierarchical substructure. In other words, there may exist several variants of a template, which are similar in contour, but exhibit subtle differences, which characterise variants of a style or sub-style. One such example is the *martinete* style, where two intra-style variants with a similar melodic contour can be distinguished [135].

Prior approaches to automatic melody classification in the context of flamenco rely on the concept of melodic similarity. Based on the assumption, that performances with a common melodic template are similar to each other and dissimilar to performances based on other templates, the task has been formulated as a supervised *k-nearest-neighbour* k-NN [155] classification scenario, where an unknown performance is labelled based on the labels of its k most similar items in an annotated database. In the context of this framework, two components are crucial, the representation of the melody and the melodic similarity measure.

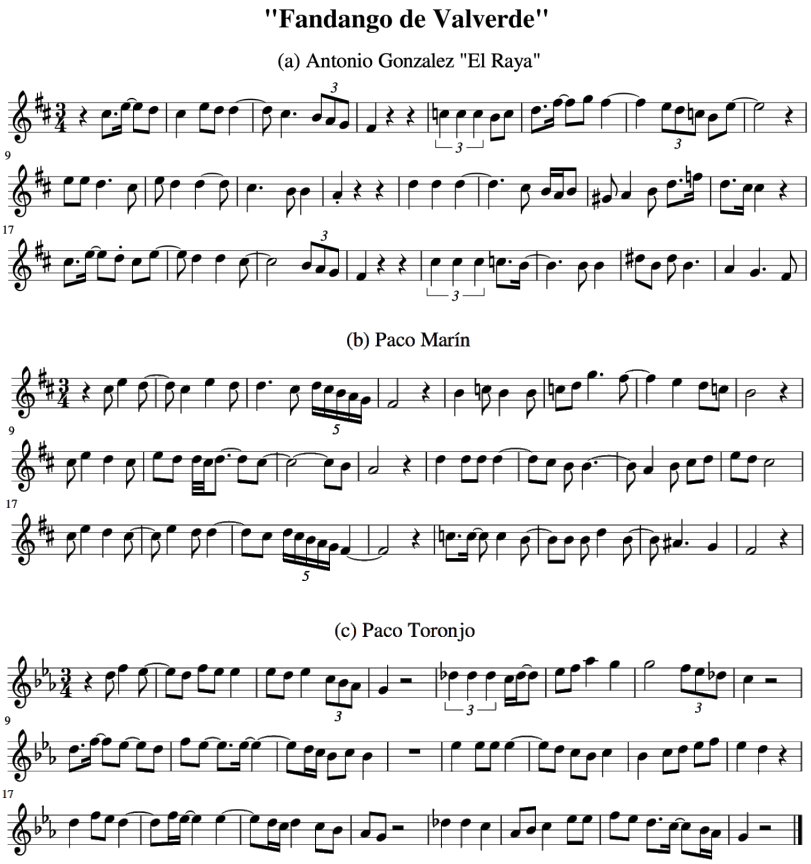


Fig. 2.2 Two performances belonging to the *fandango de Valverde* style. Top: Antonio Gonzalez "El Raya". Middle: Paco Marín. Bottom: Paco Toronjo. Transcription: Inmaculada Morales and Nadine Kroher.

In a first approach, [19] investigated the *correlation distance* and the *edit distance* between interval representations derived from automatic transcriptions. In [136], *euclidean distances* between feature vectors containing manually extracted global mid-level features were explored. A hybrid method, combining both approaches was presented in [135]. While this study provided valuable insights into the criteria used by flamenco experts to distinguish styles, the need for manual annotations poses a major limitation. Targeting fully-automatic systems, [45] and [43] explored the *warping cost* of the *dynamic time warping* (DTW) [165] alignment between pitch sequences as a melodic similarity measure. Acknowledging the relatively high computational cost of this operation ($O(M^2)$, where M is the length of the longest sequence), both studies investigated several contour simplification algorithms to reduce the sequence length. While the results for some of the proposed setups are promising with respect to classification accuracy (f-measures of up to 97% are reported), the rather small dataset containing a total of 24 recordings of two styles does not give sufficient insight into

the scalability of the method to multiple classes or its extension to unsupervised scenarios. Furthermore, mainly due to the lack of a robust transcription system form polyphonic recordings, prior approaches have solely focused on a cappella styles.

A very distinct, yet related context, in which melody classification plays an important role, is the concept of *tune families* in folk music. First introduced by [7], the term refers to popular folk tunes, which, due to oral transmission over long time spans, occur in numerous variants and consequently, when performed from memory, will show significant melodic differences. The *Meertens Tune Collection (MTC)* is a large, publicly available research corpus containing, among other sources, more than 7,000 amateur performances of Dutch folk songs. For a complete description of the collection, we refer the reader to [212]. A subset of 360 recordings was categorised by experts into 26 tune families. A study on the annotation process itself [221] revealed that melodies in the same category share common frequently occurring melodic motives which are more significant for identifying the tune family than the overall melodic contour. While in flamenco music the melody is mainly varied through ornamentation, melodies of the same tune family can exhibit structural differences, including variation in phrase repetition, and major contour modifications.

Computational studies applied to this dataset have mainly focused on symbolic approaches, i.e. [210, 216], where tune family categorisation is performed based on manual transcriptions. However, many of the proposed similarity measures rely on high-level score information, in particular with respect to meter, and are not applicable to automatic transcriptions without time quantisation. A first audio-based approach was proposed by [214]. Based on the findings of [221], the fundamental frequency envelop is first segmented into cognitively coherent units. Then, a similarity between two tracks is computed from the best match among all possible segment combinations using a sub-sequence alignment algorithm. The resulting accuracy in a similarity-based supervised kNN-classification task is reported to be 53% when classifying among 20 tune families. However, given the high temporal resolution of the fundamental frequency contour, this process is computationally expensive. Furthermore, it has not been studied how microtonal phenomena, which frequently occur in flamenco singing and which are preserved in the fundamental frequency contour, influence the pair-wise alignment costs.

2.2 A systematic study on signal representations and evaluation strategies

The aim of this study is to systematically evaluate the suitability of different signal representations in the context of melody classification. In particular, we investigate different abstraction levels from the raw signal, ranging from the fundamental frequency contour and contour simplifications to note-level transcriptions. To this end, we propose a set of evaluation metrics, which can evaluate a given representation with respect to its discriminative power among

classes, as well as its computational efficiency. Going beyond the commonly used classification accuracy, we additionally target unsupervised scenarios. In a glass-ceiling analysis, we assess the influence of pitch extraction and automatic transcription inaccuracies on the overall system performance.

While prior work on flamenco singing has been limited to the a cappella styles forming the *tonas* style family, we broaden the scope by employing an automatic transcription system capable of transcribing the singing voice melody from accompanied flamenco recordings. Here, we present a case study on intra-style categorisation, where we aim to automatically distinguish two sub-styles of the *fandango valiente* style. In order to assess the suitability of the methodology to the related task of tune family recognition, we perform a study involving a large annotated corpus of Dutch folk song recordings.

2.2.1 Classification framework

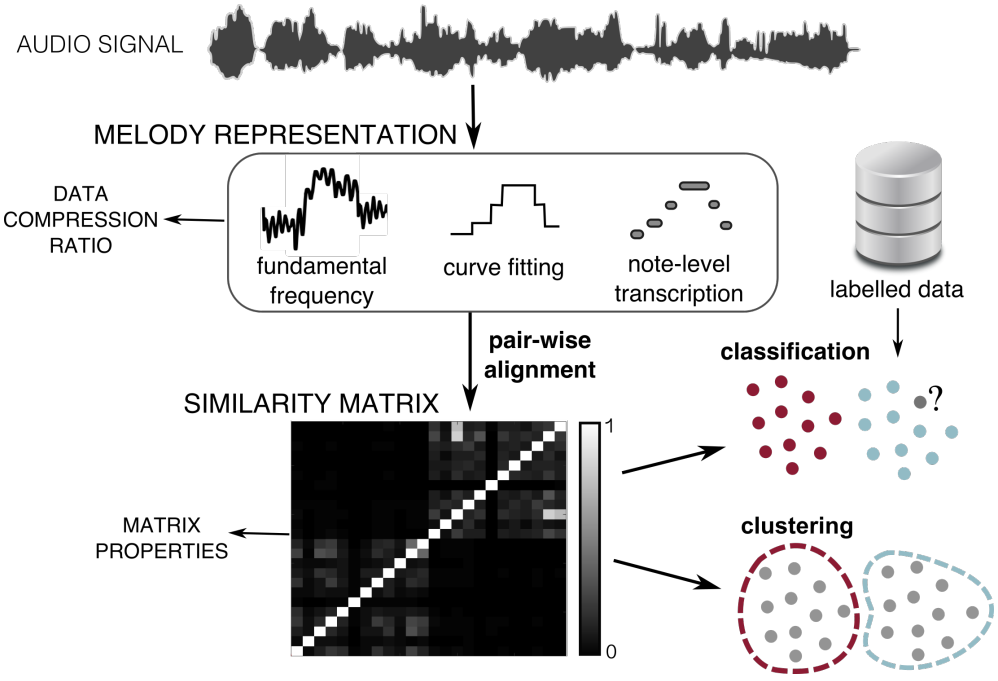


Fig. 2.3 Methodology overview.

Figure 2.3 gives an overview of the methodology applied in this study. For each of the recordings contained in a database, we first extract a signal representation describing the main melodic contour, in our case the singing voice melody. Here, we investigate three different representations. For each abstraction procedure we compute the data compression ratio with respect to the raw fundamental frequency envelope to assess its influence on the resulting computational complexity of the overall system.

We then perform pair-wise melodic alignments of the pitch sequences using the DTW algorithm. We convert the estimated global cost to a similarity value and construct a similarity matrix describing the melodic similarities among all recordings contained in the analysed database. In order to assess the properties of the resulting matrix, we formulate a graph representation and evaluate intra- and inter-cluster densities with respect to ground truth annotations. Furthermore, we use a graph visualisation tool to manually inspect the properties of the similarity matrix.

We evaluate each setup in the context of two experiments: a supervised k-NN classification task and the much more challenging unsupervised partitioning of the database into a given number of clusters. Below, all involved processing and evaluation stages are described in detail.

2.2.2 Signal representations

In this study we explore the suitability of three different melody representations: the raw fundamental frequency contour corresponding to the singing voice melody, its quantisation into piece-wise constant segments, and automatic note-level transcriptions. Figure 2.4 shows the resulting signal representations for the score depicted in Figure 2.1 (a). Each method represents a different level of abstraction with a varying degree of underlying musicological assumptions. Both, the contour simplification and the quantisation into note events, omit a certain amount of detail contained in the fundamental frequency envelope. Ideally, the resulting representation should capture the underlying melodic progression, while discarding micro-tonal phenomena which do not contribute to the perception of the overall melodic contour. Furthermore, conversion of the detailed melodic contour into constant segments or notes reduces the amount of data to be processed during the similarity calculation and consequently lowers the computational complexity. It should be mentioned that, for the case of a supervised classification task, the melody representations of the annotated database can be extracted offline. Consequently, the computational cost of this step is of minor importance in this scenario.

Fundamental frequency contour

Pitch extraction, the task of extracting the fundamental frequency f_0 corresponding to the melodic movement from an audio recording, is a well studied problem in music information retrieval and numerous approaches have been proposed. For a complete review we refer the reader to [160] and [174].

An important feature in selecting an appropriate algorithm lies in the instrumentation of the analysed recording. Here, we encounter two different scenarios: a solo singing voice and a singing voice accompanied by the guitar. The former case requires a monophonic pitch extraction, where the algorithm estimates a fundamental frequency for all voiced frames. The

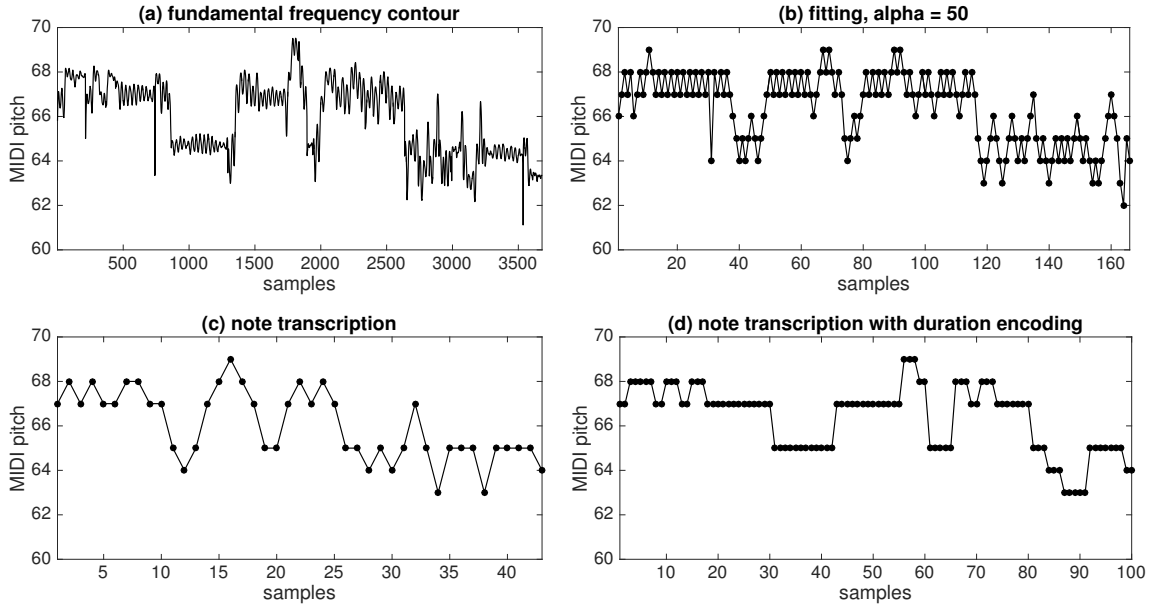


Fig. 2.4 Signal representations for a *debla* performed by *Antonio Mairena*: (a) fundamental frequency envelope, (b) piece-wise constant segments obtained with the curve fitting algorithm, (c) note transcription, (d) note transcription with duration encoding

latter refers to the task of predominant melody extraction: the fundamental frequency of the predominant source is estimated in the frames where it is present.

For the monophonic case, we use the *pYin* algorithm [128] which is available as a vamp plugin implementation[1]. We extract the fundamental frequency from an audio signal with a sampling rate of $f_s = 44.1$kHz in windows of length $w_l = 1024$ samples with a window step $w_s = 256$ samples.

To extract the fundamental frequency contour of accompanied singing, we use the *MELODIA* predominant melody extraction algorithm [175], which is also available as a vamp plugin implementation[2]. For consistency, we adopt the parameters of the *pYin* algorithm for window size and step. We set the voicing tolerance to $T_v = 0.2$, as suggested in a previous study on flamenco singing transcription [70].

For both representations, we omit silences and retain voiced frames only.

Curve fitting

While the fundamental frequency contour holds detailed information of the pitch trajectory, not all of this detail is crucial to characterising the underlying melodic progression. In an

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

attempt to remove fast pitch fluctuation caused by melodic ornamentation, vibrato and intonation instabilities, we apply the *curve-fitting algorithm (FIT)*: this contour simplification method, first proposed by [44], has been previously employed in the context of melodic similarity among flamenco styles [45]. Below, the main concept of the algorithm, which fits a step function R to f_0 in linear time, is briefly described.

Step Function Approximation: *Given a set S of M points, in our case the M_{f_0} samples of the fundamental frequency contour, and a real number $\alpha \geq 0$, determine the step function R, which approximates S with an error ϵ not larger than α, while minimising the number of links in R.*
For the error metric

$$\epsilon(R, S) = \max_{p \in R, q \in S} d_v(p, q) \tag{2.1}$$

where $d_v(p, q)$ is the vertical distance between p and q, an optimal algorithm which runs in $\Theta(M)$ time, has been proposed in [44].

The curve fitting algorithm: *Given a set of points in the plane $P = \{p_1, p_2, \ldots, p_M\}$ and an error tolerance α, place vertical segments V_i of length $D = 2\alpha$ centred around each point p_i. Let furthermore y_i^- and y_i^+ be the lower and upper endpoints of V_i, respectively. Sweeping from left to right, we maintain the intersection Δ of the y-intervals until a segment V_i is reached, whose y interval does not intersect Δ. In this case, a new step V_i is initialised, setting $\Delta = [y_i^-, y_i^+]$. Note, that the constraint specifying that each point of f_0 is within a maximum distance α from the step function, is equivalent to the condition that the step function is required to intersect all vertical segments.*

In the context of the application at hand, we first convert the fundamental frequency contour to the cent scale

$$c_0 = 1200 \cdot log_2 \left(\frac{f_0}{440\text{Hz}} \right). \tag{2.2}$$

This allows us to set α to musically meaningful values. For example, $\alpha = 50$ cents corresponds to an error tolerance of half a semitone. We then apply the curve fitting algorithm to c_0, resulting in the step function R. This process is depicted in Figure 2.5. Finally, by replacing each constant segment with a single pitch value, the melodic contour is reduced to a set of pitch values $s_{FIT} = \{s_1, s_2, \ldots s_{M_{FIT}}\}$, where $M_{FIT} \ll M_{f_0}$. An example of the resulting representation is shown in Figure 2.4 (b).

Automatic note transcription

We compare the *curve-fitting* algorithm to an automatic transcription (AT) algorithm proposed by [104]. Developed in the context of flamenco singing transcription, the algorithm extracts a note-level transcription from monophonic or accompanied singing, where each note event is characterised by its onset time, duration and a pitch value quantised to the equal-tempered scale (Figure 2.6). The system uses multiple cascaded detection functions to locate onset-

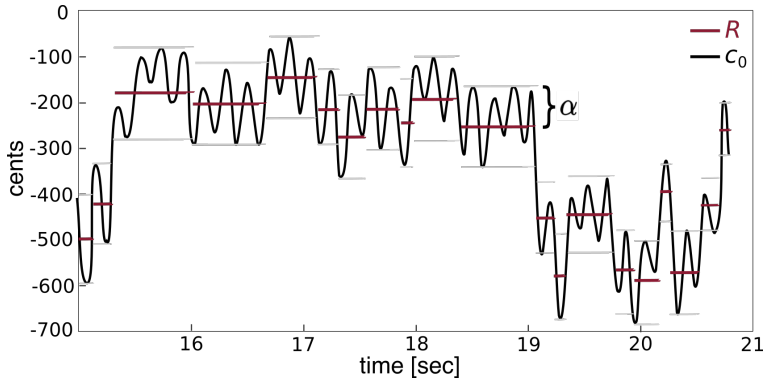


Fig. 2.5 Curve fitting example for $\alpha = 100$: Step function R and cent-scaled fundamental frequency contour c.

specific patterns in frequency contour and volume envelope. The pitch label is assigned based on local and global pitch class occurrences.

While the *curve-fitting* algorithm considers only the fundamental frequency envelope, the automatic transcription system additionally detects note events based on various onset characteristics as well as global pitch probabilities. The algorithm is available as a python module[3].

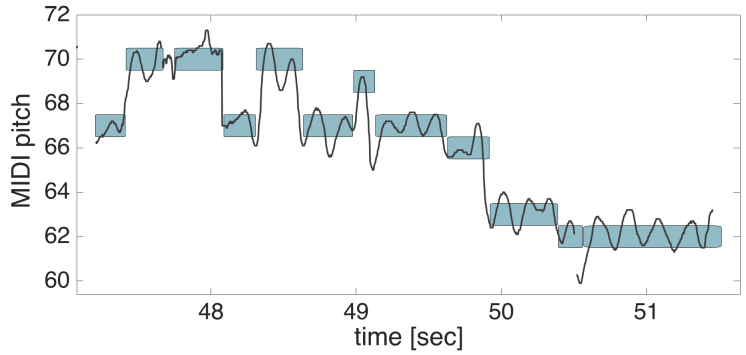


Fig. 2.6 Example of an automatic transcription: f_0 (black line) and note events (rectangles).

Omitting note durations, the fundamental frequency contour can be reduced to a set of values $s_{AT} = \{s_1, s_2, \ldots s_{M_{AT}}\}$ corresponding to the sequence of estimated pitch values, where $M_{AT} \ll M_{f_0}$. An example is depicted in Figure 2.4 (c).

[3]https://github.com/NadineKroher/PyCante

Note duration encoding

In order to investigate whether note duration information is beneficial for the computation of a similarity value between two sequences, we furthermore propose a duration-encoding scheme which assigns a stronger weight to notes which are long compared the average note duration.

More specifically, we convert a given note to a sequence of consecutive pitch values, where the number of points ν representing a note with duration dur is computed as

$$\nu(i) = \max\left(1, \left\lfloor \frac{\mathrm{dur}(i)}{\sum_j \mathrm{dur}(j) \cdot \frac{1}{M_{AT}}} \right\rceil\right) \tag{2.3}$$

and where M_{AT} is the number of notes contained in the transcription, $\mathrm{dur}(i)$ stands for the duration of the i^{th} note and $\lfloor \cdot \rceil$ denotes the nearest integer function. Figure 2.4 (d) shows an example of the resulting signal representation.

2.2.3 Alignment and similarity computation

We now aim to quantify the melodic similarity between a given pair of sequences extracted with one of the previously described methods. We have to consider the fact that versions of the same melody might be performed with significant rhythmic and melodic differences, i.e., melismatic ornamentation, prolongation or pitch errors due to poor intonation. We furthermore assume that all signal representations are to a certain extent noisy. In order to reduce the influence of these factors, we use a dynamic time-warping algorithm, originally described by [173], aiming to find the best possible alignment of the two sequences. The resulting alignment cost serves as measure to quantify their melodic distance. In this way, we compute pair-wise distances for all recordings contained in a database and subsequently construct a similarity matrix.

Dynamic time warping (DTW) and alignment cost

To determine the best possible alignment between two sequences, $s_a := \{s_{a,1}, s_{a,2}, ..., s_{a,M_a}\}$ and $s_b := \{s_{b,1}, s_{b,2}, ..., s_{b,M_b}\}$, we first define the local cost $d_{\mathrm{local}}(i,j)$ between point i of sequence s_a and point j of sequence s_b as the absolute pitch difference:

$$d_{\mathrm{local}}(i,j) = |s_{a,i} - s_{b,j}| \tag{2.4}$$

Next, we define the warping path $p = (p_1, ..., p_L)$, subject to $\max(M_a, M_b) \leq L \leq M_a + M_b - 1$. Its elements $p_k = (i,j)$ define an alignment between the two sequences, s_a and s_b, by assigning elements of s_a to elements of s_b. The warping path is subject to two restrictions: (i) the element pairing is monotonous with respect to the sample index and (ii) neighbouring elements correspond to identical or adjacent note symbols to ensure continuity.

Furthermore, the warping cost d_{warp} of an alignment path is defined as:

$$d_{\text{warp}} = \frac{\sum_{l=1}^{L} d_{\text{local}}(p_{l,1}, p_{l,2})}{L} \tag{2.5}$$

Note, that length normalisation is applied, to ensure robustness in case of variable sequence lengths.

The optimal warping path p^*, which minimises d_{warp} among all possible paths, is found via a dynamic programming approach. To this end, an M_a-by-M_b cost grid $\mathbf{D}$ is formed, where $\mathbf{D}(i,j)$ contains the cost of the optimal alignment up to $s_{a,i}$ and $s_{b,j}$. Consequently, the cost of the optimal alignment is contained in $\mathbf{D}(M_a, M_b)$. The cost grid is parsed and filled in a zig-zag mode from left to right, starting from the first row. Both, the first column and the first row represent the base case and are filled with the respective local cost values. Each of the remaining nodes stores the accumulated cost to reach it from its allowable predecessors. In particular, the matrix cell $\mathbf{D}(i,j)$ can be reached from $\mathbf{D}(i-1,j)$ (vertical transition), $\mathbf{D}(i,j-1)$ (horizontal transition) and $\mathbf{D}(i-1,j-1)$ (diagonal transition). These local path constraints were initially proposed by [173] in the context of speech recognition. Consequently, the accumulated cost results to

$$\mathbf{D}(i,j) = |s_{a,i} - s_{b,j}| + \min(\mathbf{D}(i-1,j), \mathbf{D}(i,j-1), \mathbf{D}(i-1,j-1)) \tag{2.6}$$

The optimal alignment path can be determined by tracing back the predecessors of each matrix cell, starting at $\mathbf{D}(M_a, M_b)$. However, since here only the warping cost is of interest, this procedure is not further described. The computation of the alignment cost, which runs in $O(M_a M_b)$ time, is summarised in Algorithm 2.1. An example of the cost grid and the resulting best path is shown in Figure 2.7.

Algorithm 2.1 Computing the warping cost d_{warp} between sequences s_a and s_b.

$M_a \leftarrow$ length of s_a
$M_b \leftarrow$ length of s_b
$\mathbf{D} \leftarrow$ empty $M_a \times M_b$ matrix # cost grid
for $i = 1$ to M_a **do**
    $\mathbf{D}(i,0) = |s_{a,i} - s_{b,1}|$
for $j = 1$ to M_b **do**
    $\mathbf{D}(0,j) = |s_{a,1} - s_{b,j}|$
for $j = 2$ to M_b **do**
    for $i = 2$ to M_a **do**
        $\mathbf{D}(i,j) = |s_{a,i} - s_{b,j}| + \min(\mathbf{D}(i-1,j), \mathbf{D}(i,j-1), \mathbf{D}(i-1,j-1))$
return $d_{\text{warp}} = \mathbf{D}(M_a, M_b)$

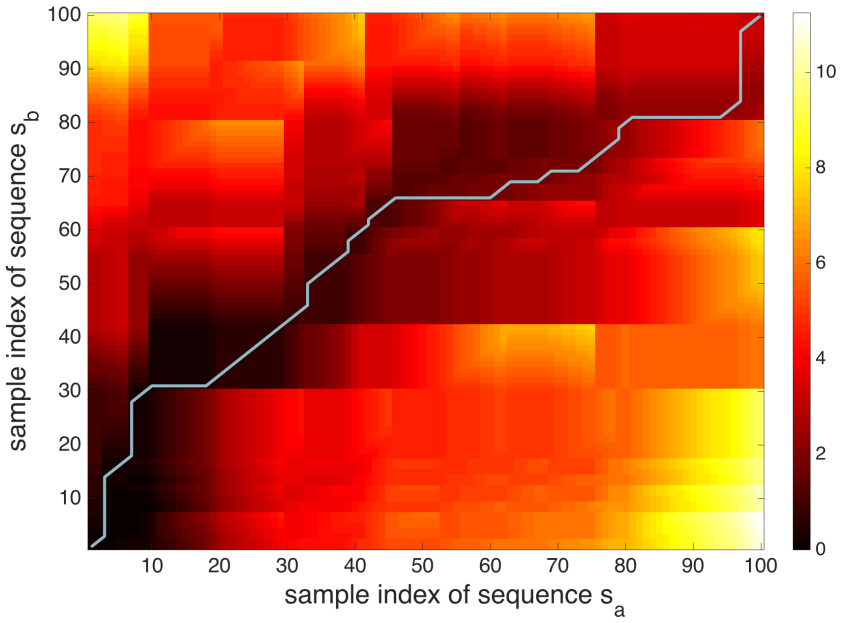


Fig. 2.7 Alignment between two note sequences: Cost grid $\mathbf{D}$ and best path p^* (light blue line).

Transposition invariance

Defining the local cost based on pitch distance assumes that the melodies under study are performed in the same key. Key transposition is a frequently occurring phenomenon in spontaneous performances which are not bound to a score: the singer adapts the key to his or her pitch range.

We therefore perform a statistical pitch occurrence analysis as described by [210] before the alignment process and adjust the sequences accordingly. When comparing melody representations s_a and s_b, we compute their pitch histograms h_a and h_b holding the number of occurrences of pitch values quantised to a semitone scale. Holding h_a fixed, we shift the bin values of h_b by Δp semitone steps, resulting in $h_{b,k}$, and then compute the correlation coefficient between both histogram vectors

$$\rho(h_a, h_{b,\Delta p}) = \frac{\mathrm{cov}(h_a, h_{b,\Delta p})}{\sigma_{h_a}\sigma_{h_{b,\Delta p}}} \tag{2.7}$$

where cov denotes the covariance and σ the standard deviation. We apply the pitch shift Δp^* to the sequence s_b which yields the highest correlation coefficient:

$$\Delta p^* = \max_{\Delta p}(\rho(h_a, h_{b,\Delta p})). \tag{2.8}$$

Similarity matrix

In order to characterise melodic similarity between N recordings contained in a dataset, we compute all pair-wise distances $d_{\text{warp}}(s_i, s_j)$ resulting in a N-by-N distance matrix $\mathcal{D}$. We convert the distances $\mathcal{D}(i,j)$ to similarity values $\mathcal{S}(i,j)$ where

$$\mathcal{S}(i,j) = \frac{1}{1+\mathcal{D}(i,j)} \tag{2.9}$$

2.2.4 Evaluation metrics

Below we describe various methods to evaluate the capability of a given melody classification system, with a specific melody representation and melodic similarity measure, to distinguish melodies according to manually annotated ground-truth categories. Most studies on melody classification (i.e. [135], [210] and [214]) evaluate proposed approaches by analysing the achieved performance in a similarity-based retrieval or classification task by means of standard information-retrieval measures. Other approaches, specifically [45] and [43], have furthermore reported computation times. However, these may depend on the implementation of the algorithm and do therefore not necessarily generalise.

In this study we employ a variety of evaluation strategies as described below in order to analyse various aspects of a given method. Given that the computational complexity of computing the DTW distance is quadratic with respect to the length of the input signal, we aim for a compact melody representation. We therefore compute the *data compression ratio* of a given representation with respect to the raw f_0 contour. Formulating the melody classification task as a graph-clustering problem, we adopt cluster-density measures to quantitatively assess the discriminative power of the similarity matrix $\mathcal{S}$ with respect to ground-truth annotations. We furthermore explore graph visualisation techniques which allow an intuitive manual inspection of the matrix structure. Apart from the results of a supervised classification task, we analyse the performance of a given setup for the more complex unsupervised clustering scenario. All evaluation measures and procedures explored in this study are described below in further detail.

Data compression ratio

In order to assess the influence of a given signal representation on the computational complexity of the DTW computation, we analyse the *data compression ratio* DCR of a signal representation of length M_{SR} with respect to the corresponding f_0 of length M_{f_0}:

$$\text{DCR} = \frac{M_{f_0}}{M_{SR}} \tag{2.10}$$

Graph representation

A similarity matrix $\mathcal{S}$ describing a set of analysed audio recordings can be represented as a complete undirected graph $G(V, E)$, where the vertices V correspond to the analysed recordings and the weighted edges E to the estimated similarity among them.

Based on this model, we apply the *Force Atlas 2* [88] algorithm to generate an approximate 2-D visualization of the graph using the *gephi* environment [6]. In this way, the dissimilarity between graph vertices is mapped to both, their distance in a 2D-plane and the thickness of their connecting line. Consequently, a pair of similar instances will be located in close proximity and connected with a thick line.

Graph visualisations allow us to manually inspect properties of the similarity matrix with respect to ground-truth melody categories and to identify data outliers. Several examples are displayed in the experimental results section.

Cluster quality

Given the graph representation of a dataset containing two or more melody categories, we aim to quantitatively assess in how far the similarity matrix reflects these categories. Ideally, melodies of the same category should be similar to each other and dissimilar to melodies belonging to other categories. Consequently, categories should correspond to clusters within the graph with strong intra-cluster edges and weak inter-cluster edges [178].

Given a ground-truth cluster C, we compute the *intra-cluster density* $\delta_{\text{int}}(C)$ as the average weight of all edges $e \in E_{\text{int},C}$ connecting nodes of the cluster

$$\delta_{\text{int}} = \sum_{e \in E_{\text{int},C}} e \frac{1}{|E_{\text{int},C}|} \tag{2.11}$$

where $|E_{int,C}|$ represents the number of edges connecting the vertices of cluster C. Similarly, we can compute the *inter-cluster density* $\delta_{\text{ext}}(C)$ as the average weight of all edges $e \in E_{\text{ext},C}$, connecting vertices of cluster C to vertices belonging to other clusters.

$$\delta_{\text{ext}} = \sum_{e \in E_{\text{ext},C}} e \frac{1}{|E_{\text{ext},C}|} \tag{2.12}$$

For a graph G consisting of ground-truth clusters $\mathcal{C}(G) = \{C_1, C_2, ...C_K\}$, the *mean intra-cluster density* $\bar{\delta}_{\text{int}}$ is computed as the average intra-cluster density over all K clusters. In a similar way, we can compute the *mean inter-cluster density* $\bar{\delta}_{\text{ext}}$ of a graph G as the average of all inter-cluster density values. Given that ideally $\bar{\delta}_{\text{int}}$ should be larger than $\bar{\delta}_{\text{ext}}$, we asses the *cluster quality* $q(\mathcal{S})$ of a given similarity matrix with respect to ground-truth clusters as

$$q(\mathcal{S}) = \frac{\bar{\delta}_{\text{int}}}{\bar{\delta}_{\text{ext}}} \tag{2.13}$$

where large values indicate a high discriminative power with respect to the ground truth classes.

Supervised classification task

As described in [45], we can evaluate a given similarity matrix $\mathcal{S}$ obtained from a dataset containing N recordings by formulating an N-fold k-nearest-neighbour classification task. Each recording is assigned a class label $\hat{C}$ based on the majority of ground-truth class labels of the k most similar tracks. The parameter k is chosen based on the empirical reference value of $k = \lfloor\sqrt{N}\rceil$ [52].

We evaluate the outcome of this classification task by means of *classification accuracy* (ACC). Let A be the contingency matrix where the first dimension represents the ground truth classes C_i and the second dimension represents the assigned labels $\hat{C}_j$. The elements A_{ij} hold the number of instances of a ground-truth class C_i which have been assigned a label $\hat{C}_j$. Consequently, the diagonal elements A_{ii} represent correctly classified instances, whereas all other elements correspond to misclassifications. The accuracy is defined as

$$\text{ACC}(A) = 100 \cdot \frac{\sum_i A_{ii}}{\sum_i \sum_j A_{ij}} \tag{2.14}$$

which corresponds to the percentage of correctly classified instances.

Unsupervised clustering task

Far more challenging than the supervised classification described above is the task of partitioning a dataset into a given number of categories K, referred to as unsupervised clustering. While this topic has been extensively studied (for a comprehensive review the reader is referred to [90]) and a number of algorithms have been proposed, the performance strongly depends on the underlying data and its discriminative power with respect to the ground-truth classes.

In this study, we use a spectral clustering algorithm proposed by [144] to partition a dataset of size N into K categories, where K corresponds to the number of annotated ground-truth classes. The related scenario, where K is unknown, is not considered here. A standard evaluation measure for unsupervised clustering algorithms is the *Adjusted Rand Index* (ARI) [86]: Given a set of ground-truth classes $\mathcal{C} = \{C_1, ...C_K\}$, we aim to evaluate a partitioning of a database $\hat{\mathcal{C}} = \{\hat{C}_1, ...\hat{C}_K\}$. Similar to the classification task described above, the entries of the contingency matrix A_{ij} hold the number of instances of ground-truth class C_i contained in cluster $\hat{C}_j$. Furthermore, we define b_j as the sum over the j^{th} column and a_i

as the sum over the i^{th} row of A. The *Rand Index* (RI) [162] evaluates the similarity of two partitions as follows:

$$\mathrm{RI} = \frac{a+b}{\binom{N}{2}} \tag{2.15}$$

The ARI is derived from the RI as

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathbf{E}(\mathrm{RI})}{\max(\mathrm{RI}) - \mathbf{E}(\mathrm{RI}}, \tag{2.16}$$

where $\mathbf{E}(\mathrm{RI})$ denotes the expected value of RI. This equation can be rewritten as

$$\mathrm{ARI} = \frac{\sum_{ij}\binom{A_{ij}}{s} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{\sum_i a_i}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{1}]/\binom{\sum_i a_i}{2}}. \tag{2.17}$$

2.3 Experimental results

We investigate three distinct applications of automatic melody categorisation. First, we focus on the task of inter-style categorisation in *a cappella* flamenco singing and evaluate the system performance for all considered melody representations. Then, based on these findings, we apply the best performing approach to the harder task of intra-style classification in *a cappella* and accompanied flamenco singing styles. Furthermore, we investigate the suitability of our approach for the task of tune-family recognition in a case study on a large corpus of annotated Dutch folk song recordings.

Below, we describe the experimental setup and the datasets used for each task in detail and discuss the results obtained.

2.3.1 Inter-style classification: Comparison of signal representations

The first experiment aims to analyse the suitability of the introduced signal representations in the context of automatic style classification in a cappella flamenco singing. We compare the average *data compression ratio* among the previously introduced melody representations, assess the *cluster quality* with respect to ground truth categories and evaluate the performance in both, a supervised k-NN classification task as well as an unsupervised clustering task.

In order to assess the influence of pitch extraction inaccuracies and automatic transcription errors, we compare, in a glass-ceiling analysis, manually corrected versions of both, the f_0 contour *(f_0-CORR)* and the automatic transcriptions *(AT-CORR)*. These corrected representations are available in the *TONAS* dataset described below.

The melody representations under study are denoted as follows:

- f_0: Raw fundamental frequency contour.
- *FIT*: Pitch sequence obtained after applying the contour-fitting algorithm with different values for α
- *AT*: Pitch sequence from automatic note transcriptions.
- *AT+DE*: Pitch sequence from automatic note transcriptions with additional note duration encoding.
- f_0-*CORR*: Manually corrected fundamental frequency contour.
- *AT-CORR+DE*: Pitch sequence from manually corrected note transcriptions with duration encoding.

Data

For this task, we compiled a subset of the publicly available TONAS dataset, which has previously been used in the context of inter- and intra-style classification [45, 134]. For a detailed description of the dataset and the *tonas* singing style, we refer to [134]. The full dataset contains 72 monophonic singing recordings belonging to two styles of the *tonas* style family, *debla* and *martinete*. Both styles are characterised by a distinct melodic skeleton, which is subject to strong melismatic ornamentation and melodic variation during performance. All recordings contain a single exhibition of this skeleton. There are two variants of *martinete* recordings contained in the dataset, annotated as *martinete I* and *martinete II*. The subset used here for the task of inter-style classification, contains all 15 *debla* and all 35 *martinete I* recordings from the *TONAS* dataset.

Results

The results of the inter-style classification task for the different signal representations are displayed in Table 2.1 and graph visualisations are shown in Figure 2.8.

We observe that the automatic transcription yields a significantly more compact melody representation (DCR = 66.78) than the contour-fitting algorithm, where the DCR ranges from 4.02 to 14.13 depending on the choice of α. This still holds when, at the cost of compactness, the note duration is encoded (DCR = 33.5). Furthermore, the note representations achieve higher values for $q(S)$ (2.38 without and 3.19 with duration encoding) and yield a better performance in both, the classification (ACC = 98%) and the clustering (ARI = 0.77) task. Comparing the two glass-ceiling setups (f_0-CORR and AT-CORR+DE), we observe a similar behaviour. This experiment furthermore shows, that the difference between extracted and manually corrected representations is larger for the fundamental frequency contour. Based

signal representation	DCR	$q(S)$	ACC	ARI
f_0	1.00	1.40	80%	0.12
FIT ($\alpha = 25$)	4.02	1.61	96%	0.25
FIT ($\alpha = 50$)	7.32	1.51	96%	0.19
FIT ($\alpha = 100$)	14.13	1.37	90%	-0.01
AT	**66.78**	2.38	**98%**	**0.77**
AT+DE	33.55	**3.19**	**98%**	**0.77**
f_0-CORR	*1.00*	*2.26*	*78%*	*0.75*
AT-CORR+DE	*23.68*	*3.34*	*98%*	*0.92*

Table 2.1 Comparison of signal representations for inter-style categorisation.

on these results, we can conclude that among the investigated options, automatic note transcriptions are the most suitable signal representation for melody categorisation.

It can be assumed that the raw fundamental frequency contour contains details, such as microtonal ornamentation, pitch instabilities or vibrato, which cause local changes in distance unrelated to the similarity of the melodic contents. Among the considered contour-simplification algorithms, best results were achieved for a value of $\alpha = 25$ cents. This is surprising, since one might suspect that fast fluctuations crossing the window boundary of a quarter-tone result in over-segmentation of the contour. It is possible that, in this way, long notes get over-segmented but result in consecutive segments of the same pitch and consequently produce an implicit duration encoding.

Comparing the obtained classification accuracies, which all lie above 90% except for the raw fundamental frequency contour, to the matrix quality and the clustering performance, exemplifies the advantage of these evaluation methods: despite the consistently high classification performance, only the note transcriptions seem to achieve a discriminatory power which allows an unsupervised partitioning into clusters which largely correspond to the ground-truth clusters.

The graph visualisations (Figure 2.8) give further insight into the structure of the similarity matrix S. For all representations we observe that the *martinete* category has a stronger cluster strength with consistently strong inner edges. This finding indicates, that there exists more variation among *deblas* than *martinetes*.

2.3.2 Intra-style categorisation

Based on the findings of the previous experiment, we now attempt to categorise variants of the same style using automatic transcriptions with note duration encoding. Intra-style classification is a harder problem than inter-style classification, since the differences in the

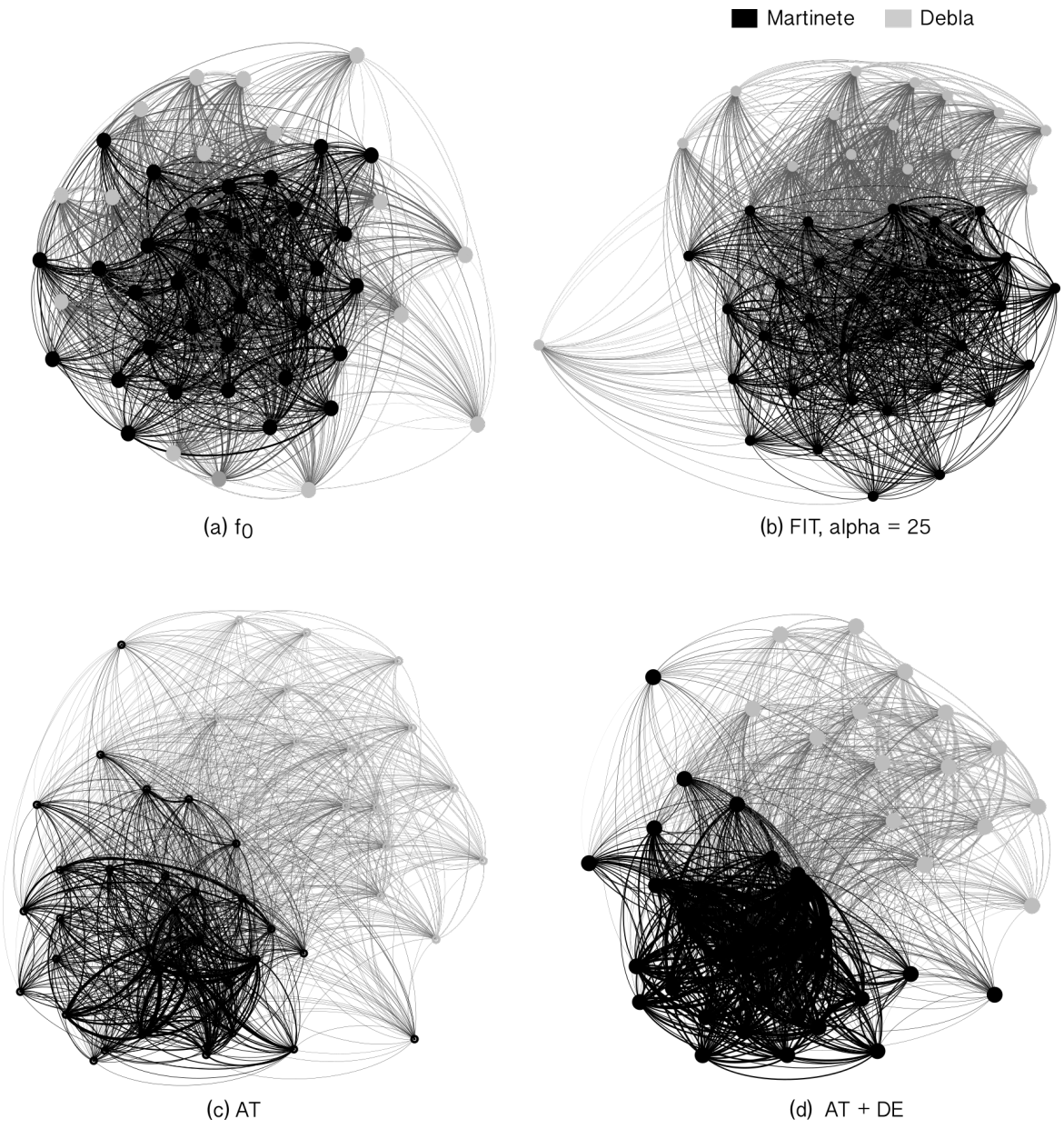


Fig. 2.8 Graph visualisations for different signal representations: (a) fundamental frequency contour, (b) curve fitting ($\alpha = 25$), (c) note transcription, (d) note transcription with duration encoding.

melodic contour are less significant. Here, we perform intra-style categorisation on two flamenco styles: The previously introduced *martinetes* and the *fandangos valientes*.

The two sub-styles of the *martinete* style are characterised by a distinct melodic progression at the beginning of the first phrase: the *martinete II* has as an additional short melodic segment which serves as an introduction and which is absent in the *martinete I* [134]. The *fandangos valientes* belong to the *fandango* style family. The two sub-styles under study are the *fandango valiente de Alosno* and the *fandango valiente de Huelva*, named after their geographic origin. An example of each category is depicted in Figure 2.9: It can be seen that both styles exhibit a nearly identical melodic progression which is furthermore subject to melodic ornamentation. Analysing both styles we notice two sections in which the categories differ: The *fandango valiente de Huelva* exhibits a characteristic melismatic movement in the first phrase and a distinct downwards movement in the third phrase which is followed by an octave jump. These characteristics are absent in the *fandango valiente de Alosno*.

Furthermore, for both, the *martinetes* and the *fandangos*, we observe melodic variation and ornamentation among the examples within the same category. Below, we describe the datasets used in this experiment and analyse the obtained results.

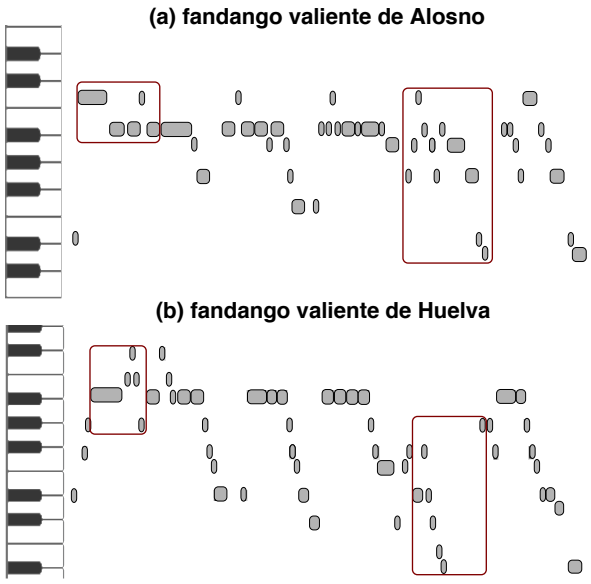


Fig. 2.9 Manual MIDI transcriptions of two *fandangos valientes*: (a) *fandango valiente de Alosno*, (b) *fandango valiente de Huelva*. Rectangles mark style-characteristic melodic progressions.

Data

In the scope of this study we gathered a dataset containing a total of 36 *fandangos valientes* (FV), 20 of which belong to the *fandangos valientes de Alosno* and 16 to the *fandangos valientes*

Dataset	$q(S)$	ACC	ARI
MAR	1.49	87%	0.62
FV	1.40	81%	0.51

Table 2.2 Intra-style categorisation based on automatic transcriptions with duration encoding.

de Huelva sub-style. All recordings are polyphonic, containing singing voice accompanied by guitar playing. Again, each recording contains a single exhibition of the main melody. Both the selection as well as the ground truth category annotation has been carried out by flamenco experts.

For the task of intra-style classification of *martinetes* (MAR), we extracted a subset of the previously introduced *TONAS* dataset, containing 35 examples of the *martinete I* and 20 examples of the *martinete II* sub-styles.

Results

The results of the intra-style categorisation task are shown in Table 2.2. As expected, the overall results for all considered evaluation measures are significantly lower compared to the simpler inter-style classification task (see the AT+DE row of Table 2.1). This observation is confirmed when comparing the graph representations (Figure 2.8 and Figure 2.10). Nevertheless, given the complexity of the task and the fact that the manual annotation process is not trivial, even for experts, classification accuracies above 80% and an *ARI* above 0.50 can still be considered a satisfying result. Both, the graph visualisations as well as the quantitative results, indicate a slightly better class distinction for the *martinete* sub-styles when comparing the two considered datasets.

2.3.3 Tune family classification

In a last experiment, we explore the suitability of the proposed approach for the task of automatic tune-family categorisation in Dutch folk music. Similar to the previously analysed case of flamenco singing styles, melodies belonging to the same tune family [31] are variants of a common prototypical melody. Owing to oral transmission of songs across long time spans and the fact that performance is usually conducted from memory, recordings of tunes from the same tune family show significant differences in rhythm, melody and form. In contrast to the case of flamenco styles, where the melodic skeleton is modified mainly through melismatic ornamentation, grace notes and prolongation, we observe that examples of the same tune family may show distinct melodic movements as well as structural differences, i.e. the absence or presence of the repetition of a single phrase. Consequently, one should expect a stronger

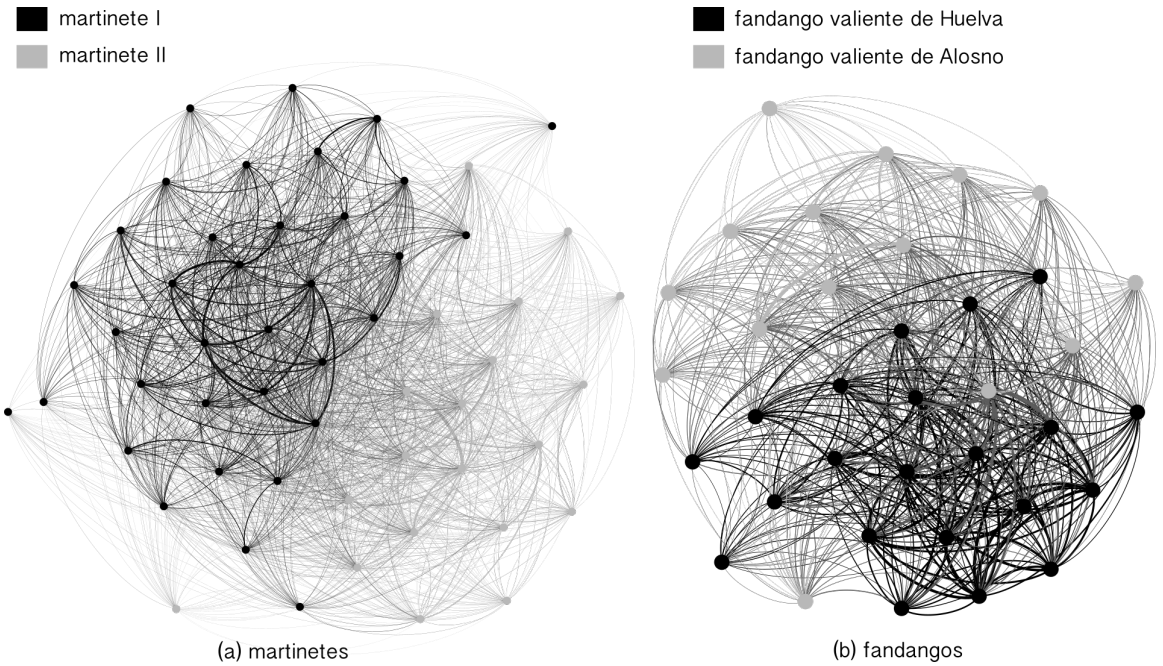


Fig. 2.10 Graph visualisations for intra-style categorisation: (a) two *martinete* variants, (b) two *fandango* variants.

variation of the melodic contour among the members of a given tune-family than among instances of the same flamenco sub-style.

Data

The *MTC-OGLAUDIO* dataset is a subset of the *Meertens Tune Collection* (MTC) [212], which contains a total of more than 7000 audio recordings of Dutch folk tunes. All tracks in *MTC-OGLAUDIO* were recorded *a cappella* by amateur singers from the 1950s to the 1990s for the radio program *Onder de groene linde*. For 360 recordings in the corpus, folk music experts from the Meertens Institute annotated the corresponding tune family. We selected a subset of 344 recordings, where performances containing multiple voices and/or accompaniment were discarded. Since the number of sung verses varied among recordings, we furthermore manually extracted the first verse only. The resulting dataset (MTC-26) and its distribution with respect to tune families is summarised in Table 2.3. As the name suggests, the dataset encompasses a total of 26 annotated categories. For the sake of comparison to the previous binary categorisation task, we furthermore define a subset (MTC-2), containing only the examples of the two tune families *Stad* and *Verre*.

denotation	tune family	# instances
Boom	Zolang de boom zal bloeien	16
Bruidje	Vaarwel bruidje schoon	9
Dochtertje	Een Soudaan had een dochtertje	11
Dreikoningenavond	Het was op een dreikoningenavond I	13
Femme	Femmes voulez vous eprouver	29
Haleweijn II	Heer Haleweijn I	3
Haleweijn IV	Heer Haleweijn IV	6
Heer	Daar ging een heer I	16
Herderinnetje	Er was een Herderinnetje	5
Jonkheer	Daar reed een jonkheer I	8
Koopman	Er was een koopman rijk en machtig	11
Lindeboom	Een lindeboom stod in het dal	10
Maagdje	Daar sou er een maagdje vroeg opstaan II	10
Meisje	Er was een meisje van zestien jaren I	12
Ruiter I	Er reed er eens een ruiter I	18
Ruiter II	Daar was laatstmaal een ruiter II	15
Schipper	Lieve schipper vaar me over I	20
Soldaat	Soldaat kwam uit de oorlog	17
Stad	Ik kwam laatst eens in de stad	21
Stavoren	Het vriouwtje van Stavoren I	16
Stil	Kom laat ons nu zo stil niet zijn I	11
Verre	Wat zag ik daar van verre I	21
Vrouwtje	Er wwonde een vrouwtje al over het bos	9
Zoetelifies	En er waren eens twee zoeteliefjes	20
Zomerdag	Het was laatst op een zomerdag	17

Table 2.3 Contents of the OGL dataset.

Results

The results of the automatic tune family categorisation are shown in Table 2.4 and a graph-visualisation for the binary task is shown in Figure 2.12. For both the supervised classification task (ACC = 95%) as well as the unsupervised clustering (ARI = 0.81), we obtain for the binary subset MTC-2 comparable results to the intra-style classification of flamenco styles. Nevertheless, the matrix quality $q(S)$ indicates a poorer separation of the classes, probably due to weaker inner edges. A reason could be the particular characteristics of the intra-class variation encountered in this dataset: due to the nature of the alignment algorithm, the presence of a repeated phrase in a track can lead to a misalignment of a large part of the sequence when compared to a melody which does not contain this repetition. In comparison, the influence of melodic ornamentation may result in an increased alignment cost, but does not usually lead to a misalignment. An example is shown in Figure 2.11. In Figure 2.11 (a), two sequences representing the melody of performances of the tune family

Verre differ in the repetition of the last two phrases, while the rest of the melodies are fairly similar. Their computed similarity value results to $s = 0.14$. Figure 2.11 (b) shows two *debla* performances which show strong differences resulting from ornamentation and note prolongation. Nevertheless, their computed similarity is significantly higher, with $s = 0.70$. This observation coincides with the findings of [221], which indicate that, for the assignment of a tune family, the occurrence of particular motives is more significant than the overall melodic contour.

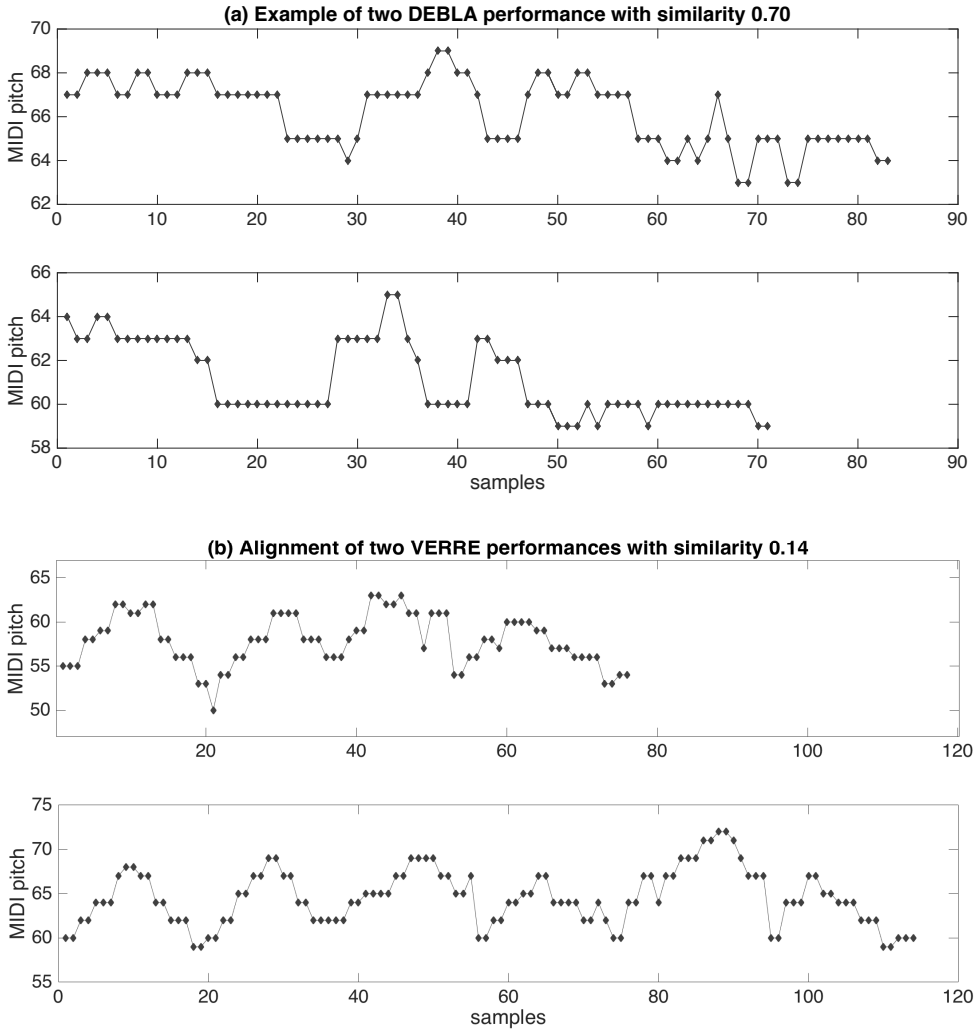


Fig. 2.11 Example of sequence pairs and their computed melodic similarity: (a) two performances of the *Verre* tune family and (b) two *debla* performances.

When increasing the complexity of the task by extending to the MTC-26 dataset, both the ACC and ARI drop significantly (ACC = 64% and ARI = 0.09). In particular, the unsupervised clustering does not seem to give meaningful results for this complex setup.

Dataset	$q(S)$	ACC	ARI
MTC-26	1.52	64%	0.09
MTC-2	1.81	95%	0.81

Table 2.4 Tune family categorisation based on automatic transcriptions with duration encoding.

Nevertheless, we obtain a better performance than that reported by [214] (ACC = 46%), where the f_0 was used to classify among 20 tune families.

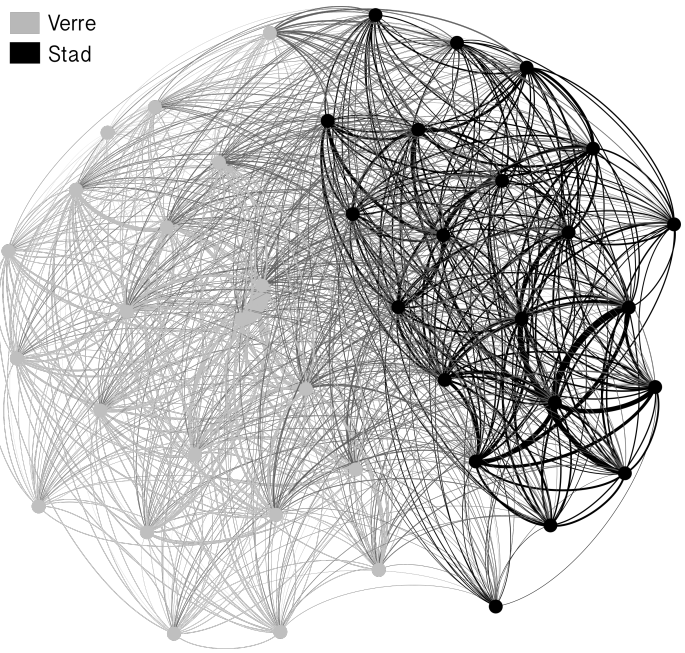


Fig. 2.12 Graph visualisation for the binary tune categorisations task: Tune families "Verre" vs. "Stad".

2.4 Conclusions

In an audio-based approach, we investigated signal representations and evaluation metrics for automatic melody classification in two application contexts, style classification in flamenco music and tune-family recognition in Dutch folk songs. We compared various melody representations, which can be extracted directly from the audio signal, and showed that pitch sequences from automatic transcriptions yield the best results with respect to system performance. This representation provides a data-compression ratio above 30 and consequently reduces the computational complexity of the alignment process with respect to the raw

fundamental frequency contour significantly. We showed that this approach gives convincing results in a supervised classification scenario for inter- and intra-style classification of flamenco music, for both, manual and automatic transcriptions.

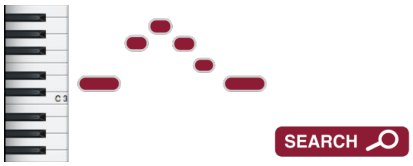
We moved beyond the commonly used evaluation based on the system performance in a classification task towards the more complex challenge of unsupervised clustering. We demonstrated that promising classification results do not necessarily imply sufficient discriminatory power for a meaningful unsupervised partitioning of the dataset. Furthermore, we showed how graph visualisations can be employed to reveal interesting data properties, such as varying strength of intra-cluster densities among classes. We applied evaluation measures from graph theory and data clustering in order gain a better understanding of system limitations.

Comparing the results obtained on flamenco music to the case study on tune-family recognition in Dutch folk music, we found that the type of melodic variation which occurs among instances of the same class influences the achievable performance. Therefore, in the context of tune family classification, future work could be directed towards a similarity measure robust to structural differences, i.e. by considering phrase repetition. Despite the performance drop when the number of ground-truth categories increases, using automatic note transcriptions, we obtained a higher classification accuracy compared with the results reported in a related audio-based study [214] using the fundamental frequency envelope.

An important aspect of future work in the context of flamenco music is the creation of a large annotated audio collection, similar to the MTC dataset. This requires a significant amount of musicological effort, since a clear style taxonomy, in particular with respect to melodic templates, has so far not been established. However, this effort represents a necessary step towards the development of a large-scale flamenco classification system. In addition, the proposed evaluation metrics could be extended to better fit the hierarchical structure of the templates, in a sense that class confusion among sub-styles should be penalised less than misclassifications on a higher hierarchical level. A further possible line of research is the unsupervised analysis, i.e. formulated as a clustering task, of large flamenco corpora with respect to melodic similarity. In this context, the development or adaptation of clustering algorithms which exploit application-specific data structures is another challenging research task.

Chapter 3

Melody Retrieval



3.1 Introduction

Melody retrieval refers to the task of automatically locating a given melodic sequence among a large number of candidates in a digital music collection [150]. Research on this topic has mainly been carried out in the context of *query-by-humming* (QBH) systems. Initially introduced in the mid 1990s [66], the goal of QBH is to retrieve a specific melody from a database, based on a sung or hummed user query. In this way, users can find metadata for a tune, of which they only remember the melody. The basic framework is depicted in Figure 3.1. Over the years, numerous methods have been proposed (see for example [97] and [172]) and the task has been included in the annual *Music Information Retrieval Exchange* (MIREX) challenge. Furthermore, several fully functional QBH engines have made their way into market-ready applications[1][2]. In most cases, QBH systems operate on databases containing monophonic musical scores in machine-readable format. Fewer approaches have focused on the more challenging task of detection of sung melodies in collections of raw audio files [187, 177].

The method presented in this chapter targets a different application. Our goal is to detect occurrences of a manually defined melodic pattern in a large corpus of flamenco recordings. Beyond the application in user-oriented data exploration systems, this task if of fundamental for data-driven musicological studies. Future research can, for example, apply this system to

[1]https://www.soundhound.com
[2]https://tunepal.org/

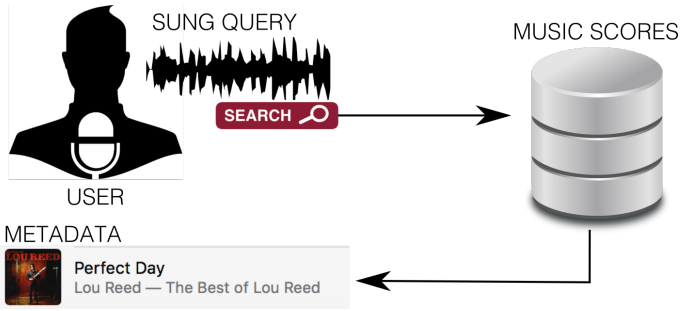


Fig. 3.1 Illustration of a query by humming system.

explore the preservation of patterns encountered in the folkloric origins of flamenco, as well as the occurrence of shared patterns across styles.

Given the absence of musical scores, a first approach in the context of flamenco [157] operates on the fundamental frequency (f_0) contour of the singing voice melody. More specifically, a modification of the *context-dependent dynamic time warping algorithm* is used to locate manually defined prototypical MIDI sequences in a collection of flamenco recordings. However, as established in Chapter 2, the f_0 contour contains, apart from the underlying melodic movement, a high amount of micro-tonal detail, which can distort similarity scores and which furthermore significantly increases the computational cost.

Here, we propose a method which locates a prototypical pattern, manually defined in MIDI format, within a large corpus of automatic transcriptions, directly extracted from flamenco music recordings. A schematic overview of the targeted framework is depicted in Figure 3.2. The task encompasses a number of computational challenges:

1. Given the improvisational nature of flamenco music and the frequent use of melodic ornamentation, we do not expect to encounter exact instances of the query pattern. Instead, we aim to locate melodic sequences which largely follow the contour of the query melody, but may contain grace notes, melismatic ornamentation and micro-tonal embellishments.

2. Due to the absence of written scores, the use of automatic transcriptions is inevitable. However, these symbolic representations can be expected to contain a certain amount of noise in form of pitch estimation, segmentation and voicing errors. As a result, not all of the query notes are necessarily contained in the matching sequence.

3. An instance of the targeted query pattern can be located anywhere in a transcription. In other words, the query sequence matches at best a subsequence of a given transcription. In addition, there is no a priori knowledge about whether the pattern is contained in a specific transcription or not.

4. It is not guaranteed that query and performance transcription are performed in the same key. Given that the query melody only refers to a short subsequence of the performance, a transposition based on pitch histograms (as described in Section 2.2) is likely to be inaccurate.

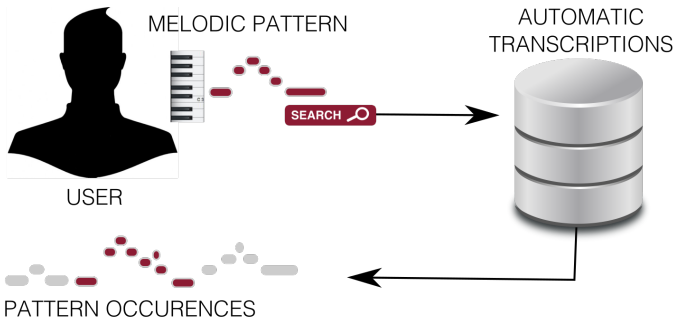


Fig. 3.2 Illustration of the targeted application.

To address these issues, we developed a gap-tolerant alignment method, which detects occurrences of the query pattern in subsequences of automatic transcriptions and assigns a score to each match. The gap-tolerance allows for non-exact matches of the query pattern to note sequences containing transcription errors and melodic variation. The method furthermore considers the intervallic structure instead of absolute note pitches, in order to account for key transposition. The proposed algorithm is described in Section 3.2 and its evaluation in a case study on melodic patterns of the *fandango* style is detailed in Section 3.3.

3.2 Retrieval of melodic patterns in automatic transcriptions

3.2.1 The Needleman-Wunsch algorithm

The core of our method is a modification of the well known Needleman-Wunsch (NW) algorithm [143] from the area of bioinformatics, which has previously been employed in the context of melodic sequence comparison [210, 216, 163, 15]. Initially, the NW algorithm was proposed as a method for global alignment of molecular sequences. The term global alignment refers to the fact that when two sequences of discrete symbols are being matched, the objective is to align them from the beginning to the end, without omitting parts around the endpoints. During the alignment procedure, gaps are allowed to be formed. In the original NW formulation gaps are not penalised. Given two sequences of discrete symbols, the original NW algorithm can be formulated as a dynamic programming method that creates a dot matrix and finds the best path of dots on it, i.e., a path of dots (nodes) of increasing

index that accumulates the largest score (number of dots). The dot matrix (also known as similarity grid) is formed by placing one pattern on the x-axis and the other one on the y-axis. An element of the dot grid is set equal to "1" if the symbols corresponding to its coordinates coincide. However, more recent variants of the NW algorithm use more complex scoring functions and usually employ a gap penalty term (see as an example the BLAST2 algorithm [195]).

Let $A := \{a_1, a_2, \ldots, a_{M_A}\}$ and $B := \{b_1, b_2, \ldots, b_{M_B}\}$ be two sequences to be aligned. First, we define the local similarity function $\gamma(i, j)$, which assigns a similarity score to the pair formed by the i-th element of sequence A and the j-th element of sequence B. This function is usually adapted to the application context. If, as in the problem at hand, the sequences contain musical notes, $\gamma(i, j)$ ideally relates in some way to the perceived pitch difference. Furthermore, a value for the gap penalty g is chosen, which penalises the skipping of elements in the alignment. Then, an $(M_A + 1)$-by-$(M_B + 1)$ similarity grid $\mathbf{D}$ is formed, where the last row and column are initialised as

$$\mathbf{D}(M_A + 1, j) = -(M_B + 1 - j) \cdot g$$

and

$$\mathbf{D}(i, M_B + 1) = -(M_A + 1 - i) \cdot g.$$

Starting at element (M_A, M_B), the gird is then parsed in a zig-zag path, moving from right to left and from bottom to top, where the elements are filled as follows:

$$\mathbf{D}(i, j) = \max \begin{cases} \mathbf{D}(i + 1, j) - g \\ \mathbf{D}(i, j + 1) - g \\ \mathbf{D}(i + 1, j + 1) + \gamma(i, j) \end{cases} \tag{3.1}$$

Furthermore, in order to be able to trace the best alignment path, for each cell, the successor is stored in matrix Ψ, where $\Psi(i, j)$ holds the pair of indices corresponding to the matrix coordinates from which $\mathbf{D}(i, j)$ is reached according to equation 3.1.

The resulting alignment score d_{align} quantifying the similarity between A and B is contained in $\mathbf{D}(1, 1)$ and the optimal alignment path p can be traced in a forward manner by following the successors stored in Ψ from $\Psi(1, 1)$ to $\Psi(M_A, M_B)$. A horizontal transition from a cell to its successor corresponds to a gap being inserted into A, a diagonal transition represents a match, and a vertical transition corresponds to a gap being inserted into B. Note, that equation 3.1 is not necessarily uniquely defined, and consequently there may be multiple optimal sub-solutions. Furthermore, the algorithm can also be formulated in the reverse direction, meaning that the matrix is filled from the top left to the bottom right corner and the best path is obtained through backtracking.

Algorithm 3.1 Computing the alignment score d_{align} and the alignment path p for two sequences, s_a and s_b with the NW algorithm.

$M_A \leftarrow$ length of s_a
$M_B \leftarrow$ length of s_b
$\mathbf{D} \leftarrow$ empty $(M_A + 1) \times (M_B + 1)$ matrix # similarity grid
$\Psi \leftarrow$ empty $(M_A + 1) \times (M_B + 1)$ matrix # successor index matrix
for $i = 1$ to $M_A + 1$ **do**
    $\mathbf{D}(i, M_B + 1) = -(M_A + 1 - i) \cdot g$
for $j = 1$ to $M_B + 1$ **do**
    $\mathbf{D}(M_A + 1, j) = -(M_B + 1 - j) \cdot g$
for $j = M_B + 1$ to 1 step -1 **do**
    for $i = M_A + 1$ to 1 step -1 **do**
        $\mathbf{D}(i, j) = \max(\mathbf{D}(i + 1, j) - g, \mathbf{D}(i, j + 1) - g, \mathbf{D}(i + 1, j + 1) + \gamma(i, j))$
        $\text{ind} = \text{argmax}(\mathbf{D}(i + 1, j) - g, \mathbf{D}(i, j + 1) - g, \mathbf{D}(i + 1, j + 1) + \gamma(i, j))$
        if ind == 1 **then**
            $\Psi(i, j) = (i + 1, j)$
        else if ind == 1 **then**
            $\Psi(i, j) = (i, j + 1)$
        else
            $\Psi(i, j) = (i + 1, j + 1)$
$d_{\text{align}} = \mathbf{D}(1, 1)$ # alignment score
$p \leftarrow$ empty array # alignment path
$i = 1$
$j = 1$
while $i \leq M_A$ and $j \leq M_B$ **do**:
    if $\Psi(i, j) == (i + 1, j + 1)$ **then**
        append (a_i, b_j) to p # diagonal transition
    else if $\Psi(i, j) == (i + 1, j)$ **then**
        append (a_i, gap) to p # vertical transition
    else
        append (gap, b_j) to p # horizontal transition
    i = $\Psi(i, j, 1)$
    j = $\Psi(i, j, 2)$
return d_{align}, p

Pseudo-code for the NW algorithm is given in Algorithm 3.1 and a toy example for two numerical sequences is shown in Figure 3.3, where $g = 1$ and

$$\gamma(i,j) = \begin{cases} 1.5 & \text{if } a_i = b_j \\ 0 & \text{otherwise} \end{cases}. \tag{3.2}$$

In this example, the alignment path p results to

56	57	58	59	57
–	57	58	–	57

with score $d_{align} = 2.0$.

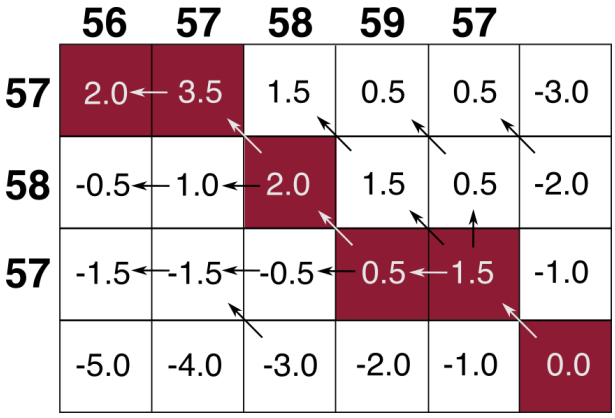


Fig. 3.3 Alignment of two sequences with the NW algorithm. Red cells form the alignment path. Arrows indicate transitions from successors when parsing the grid.

3.2.2 Proposed alignment algorithm

The problem addressed in this chapter cannot be treated as a global alignment task because our goal is to detect occurrences of a pattern in a significantly longer stream of notes. We are therefore proposing a modification of the NW algorithm, that preserves its fundamental characteristics and adds the capability to retrieve a ranked list of subsequences from an automatic transcription. Each retrieved result aligns, in some optimal sense, with the given prototype pattern. The novelty of our approach lies in the fact that it introduces a systematic way to:

(a) extract iteratively occurrences of the reference pattern, ranked with respect to similarity score

(b) embed endpoint constraints in the NW method

(c) ensure invariance to key changes because the alignment takes place on the sequences of intervals derived from the pitch sequences that are being matched

(d) formulate transition costs between nodes of the similarity grid as a function of intervalic differences

At a first stage, the proposed method operates on pitch sequences only, ignoring note durations. At a second stage, the results are refined by removing alignments that correspond to excessive local time-stretching. In the remainder of this chapter, we will use the abbreviation *mNW* for the proposed method.

In order to describe *mNW*, let $A := \{a_1, a_2, \ldots, a_{M_A}\}$ and $Q := \{q_1, q_2, \ldots, q_{M_Q}\}$ be the pitch sequences of the automatic transcription and the search pattern, respectively, where elements a_i and q_j are pitch values in some symbolic (MIDI-like) format. At this stage, note durations are ignored. Sequence Q is manually defined and reflects our musicological knowledge of the pattern to be detected. For example, pattern "A" of our experimental setup (Section 3.3) is represented by the following sequence of MIDI values:

$$\{64, 67, 65, 64, 67, 65, 65, 64, 62, 60, 58, 57\}$$

We now define that,

$$\delta_Q(j_2, j_1) = q_{j_2} - q_{j_1},$$

subject to $1 \leq j_1 < j_2 \leq M_Q$, is the music interval formed between the j_1-th and j_2-th pitch value of the prototype pattern, which are not necessarily adjacent, and, similarly

$$\delta_A(i_2, i_1) = a_{i_2} - a_{i_1},$$

subject to $1 \leq i_1 < i_2 \leq M_A$, is the music interval formed between the i_1-th and i_2-th pitch value of the automatically generated transcription. The proposed *mNW* algorithm seeks a subsequence of A with increasing, but not necessarily adjacent indices, such that the resulting sequence of intervals matches in some optimal scoring sense, a sequence of intervals formed by a subsequence of p, also of increasing, but not necessarily adjacent index.

To solve this problem from a dynamic programming perspective, we place A on the vertical and Q on the horizontal axis and form an $(M_A + 1)$-by-$(M_Q + 1)$ scoring grid $\mathbf{D}$, where the last row and column are initialised to zero.

As in the NW algorithm described above, we then proceed row-wise, decreasing the row index and examining the nodes of each row at decreasing column index, which stands for a standard zig-zag scanning procedure. The accumulated score, $\mathbf{D}(i, j)$, at node (i, j), where $i < M_A$ and $j < M_Q$ is computed as follows:

$$h = \max_{j+1 \leq k \leq j+G_h} \mathbf{D}(i+1, k) + \gamma(\delta_A(i+1, i), \delta_Q(k, j)) \tag{3.3}$$

$$v = \max_{i+1 \leq m \leq i+G_v} \mathbf{D}(m, j+1) + \gamma(\delta_A(m,i), \delta_Q(j+1,j)) \tag{3.4}$$

$$\mathbf{D}(i,j) = \max\{h, v\} \tag{3.5}$$

where parameters G_h and G_v are positive integers that define the search radius for successors on the horizontal and vertical axis, respectively, and function $\gamma(.)$ is defined as:

$$\gamma(x,y) = \begin{cases} 1, \text{if } x = y, \\ -1, \text{if } \mid x - y \mid = 1, \\ -\infty, \text{if } \mid x - y \mid > 1, \end{cases} \tag{3.6}$$

The first two equations impose that the best successor of node (i, j) resides either on the next row (the $(i+1)$-th row) or on the next column (the $(j+1)$-th column). Parameters G_h and G_v control the horizontal and vertical gap length, respectively. In other words, they control how many pitch values can be skipped horizontally or vertically when searching for the best successor of the node.

Function γ rewards equal intervals with a score equal to $+1$, penalizes with -1 any pair of intervals that differ by one semitone and forbids intervalic differences larger than a semitone to take place, hence the $-\infty$ penalty. An example is shown in Figure 3.4: The transition from $\mathbf{D}(i+2, j+1)$ to $\mathbf{D}(i,j)$ shown in (a) yields a score of $\gamma = 1$, because the musical interval between 60 and 63 in the transcription is equal to the interval between 64 and 67 in the query melody. The transition from $\mathbf{D}(i+1, j+1)$ to $\mathbf{D}(i,j)$ shown in (b) does however yield $\gamma = -\infty$, since the interval in the automatic transcription (from MIDI note 60 to MIDI note 59) deviates more than one semitone from the corresponding interval in the query pattern (from MIDI note 64 to 67). In this way, the algorithm does not only compensate for key transposition, but furthermore considers the case when an interval is broken into several subintervals. For example, the note sequence succession $\{51, 54\}$ in a query pattern might correspond to $\{52, 53, 55\}$ in a performance transcription, where a note in the middle is inserted in form of a transitional grace note. However, the intervals from first to last note are identical in both sequences.

After a node has been processed, the coordinates of its best successor are again stored in a separate matrix, Ψ. Once the whole grid has been scanned, the highest accumulated score on the first E_1 columns is selected and forward tracking on matrix Ψ reveals the best alignment path. However, this path will be rejected if it does not end in one of the last E_2 columns of the grid. Therefore, parameters E_1 and E_2 stand for the endpoint constraints of the alignment procedure, i.e., we permit that at most $E_1 - 1$ and $E_2 - 1$ notes are omitted from the left and right endpoints of the prototype pattern, respectively. If a path is rejected, we repeat from the second highest score until a valid path is detected or until all nodes of the first E_1 columns have been processed as candidate starting points of the best path. For the sake of completeness, pseudo-code for the proposed method is provided in Algorithm 3.2.

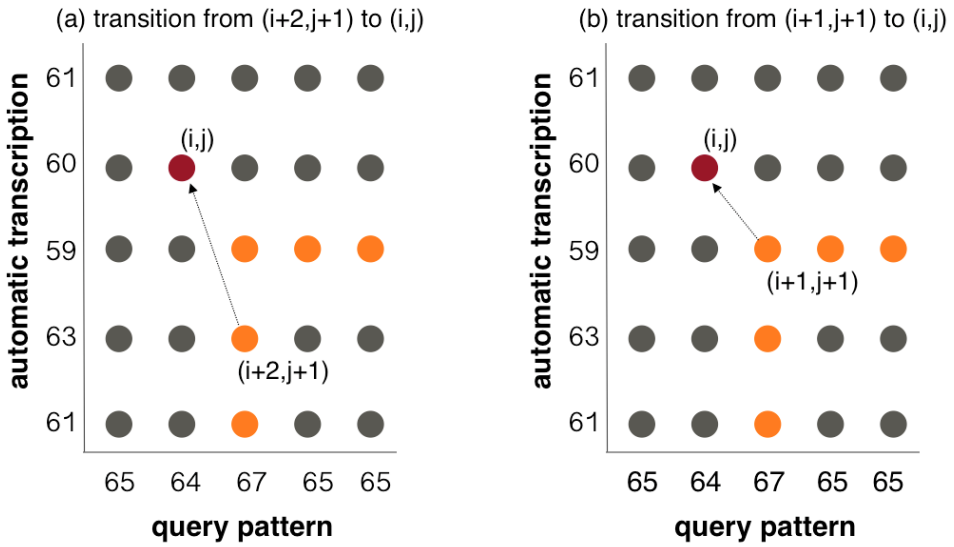


Fig. 3.4 Example of transitions to $\mathbf{D}(i, j)$. Orange cells denote the search range ($G_v = G_h = 3$).

Obviously, if we want the algorithm to return two pattern occurrences, the procedure will be repeated until a second path is revealed, and, of course, this can be readily extended to address any number of desired occurrences.

An example of the alignment process is shown in Figures 3.5 and 3.6 and Table 3.1. Figure 3.5 shows the detection of a query pattern in an automatic performance transcription. The corresponding similarity grid together with the extracted alignment path is depicted in Figure 3.6. Table 3.1 details the alignment result. It can be seen, that the transcription is significantly longer than the query pattern and the matching subsequence is located at the end of the stream. The resulting alignment shows that three notes are skipped in the transcription, and one note is skipped in the manually defined pattern.

3.2.3 Post-processing

Initial experiments with the proposed algorithm showed that some search results, while being numerically significant, have little musical meaning. In particular, ignoring note durations results in matches of the query pattern with subsequences which exhibit a very different rhythmic structure.

Therefore, after the first processing stage has been completed, the obtained results are subsequently filtered at a second stage. More specifically, in order to restrain note duration variability, we compute the sequence of inter-onset differences of the notes of a formed path on both axes and discard any path for which at least two ratios of aligned inter-onset durations

Algorithm 3.2 Computing the mNW-alignment of two sequences, A and B.

$(G_h, G_v) \leftarrow$ horizontal and vertical search radius
$(E_1, E_2) \leftarrow$ left and right endpoint constraint
$M_A \leftarrow$ length of s_a
$M_B \leftarrow$ length of s_b
$\mathbf{D} \leftarrow$ empty $(M_A + 1) \times (M_B + 1)$ matrix # similarity grid
$\Psi \leftarrow$ empty $(M_A + 1) \times (M_B + 1)$ matrix # successor index matrix
for $i = 1$ to $M_A + 1$ **do**
    $\mathbf{D}(i, M_Q + 1) = 0$
for $j = 1$ to $M_Q + 1$ **do**
    $\mathbf{D}(M_A + 1, j) = 0$
for $j = M_Q + 1$ to 1 step -1 **do**
    for $i = M_A + 1$ to 1 step -1 **do**
        for $k = 1$ to G_h **do**
            $h[k] = \mathbf{D}(i + 1, j + k) + \gamma(\delta_A(i + 1, i), \delta_Q(j + k, j))$
            $h_{\text{val}} = \max(h)$
            $h_{\text{ind}} = \text{argmax}(h)$
        for $m = 1$ to G_v **do**
            v[m] $= \mathbf{D}(i + m, j + 1) + \gamma(\delta_A(i + m, i), \delta_Q(j + 1, j))$
            $v_{\text{val}} = \max(v)$
            $v_{\text{ind}} = \text{argmax}(v)$
        if $h_{\text{val}} > v_{\text{val}}$ **then**
            $\mathbf{D}(i, j) = h_{\text{val}}$
            $\Psi(i, j) = (i + 1, j + h_{\text{ind}})$
        else
            $\mathbf{D}(i, j) = v_{\text{val}}$
            $\Psi(i, j) = (i + v_{\text{ind}}, j + 1)$
while $\max(\mathbf{D}[:, 1 : E_1]) > -\infty$) **do**
    $(\text{sCol}, \text{sRow}) = \text{argmax}(D[:, 1 : E_1])$ # start indices
    $d_{\text{align}} = \mathbf{D}(\text{sCol}, \text{sRow})$ # alignment score
    $p \leftarrow$ empty array # alignment path
    $i = 1$
    $j = 1$
    while $i \leq M_A$ and $j \leq M_Q$ **do**:
        $(l_i, l_j) = \Psi(i, j, 1)$
        for $n = i + 1$ to l_i **do**
            append (a_i, gap) to p # skip notes in transcription
        for $n = j + 1$ to l_j **do**
            append (gap, q_j) to p # skip notes in query
        append (a_{l_i}, q_{l_j}) to p # add target note to path
        i $= l_i$
        j $= l_j$
    if $p(\text{end}, 2) \geq (M_Q - E_2)$ **then return** d_{align},p # return score and path
    else
        $\mathbf{D}(\text{sCol}, \text{sRow}) = -\infty$
return 0 # no path found

transcription		**query pattern (A)**	
pitch	duration	pitch	duration
59	0.27	63	0.50
59	0.54	63	0.50
58	0.15	62	1.00
58	0.59	62	0.50
58	0.22	62	0.50
62	0.26	66	0.50
58	0.18	–	–
62	0.17	–	–
58	0.52	62	1.0
60	0.17	–	–
57	0.17	61	0.50
–	–	58	0.50
53	0.16	57	0.50

Table 3.1 Best alignment result of pattern A against an automatically generated Valverde transcription: symbol "-" marks a skipped note (gap insertion).

exceed a predefined stretching threshold (equal to 3 or 1/3 in our study). This is equivalent to imposing, at a post-processing stage, a local time-warping threshold.

3.3 A case study on *fandango* patterns

We demonstrate the performance of the proposed algorithm in a query-by-example task. We aim at detecting occurrences of manually annotated MIDI sequences in a corpus of automatic transcriptions of polyphonic flamenco recordings. In this study, we focus on *fandangos de Valverde* (FV), a singing style belonging to the family of the *fandangos* [102].

Like most *fandangos*, the *fandangos de Valverde* are bi-modal in a structural sense [58]: solo guitar sections are set in *flamenco mode*, a scale with the diatonic structure of the Phrygian scale but with the dominant and sub-dominant located on the second and third scale degree, respectively (Figure 3.7). Singing voice sections are set in major mode and modulate only in the last phrase back to *flamenco mode*.

Having evolved from Spanish folk tunes, songs belonging to this style are based on a particular melodic skeleton which, during interpretation, is subject to melodic and rhythmic modifications in terms of an expressive performance. The skeleton is composed of five distinct patterns (Figure 3.8) which occur in the form A-B-A'-C-A-D (where A' refers to a variant of A).

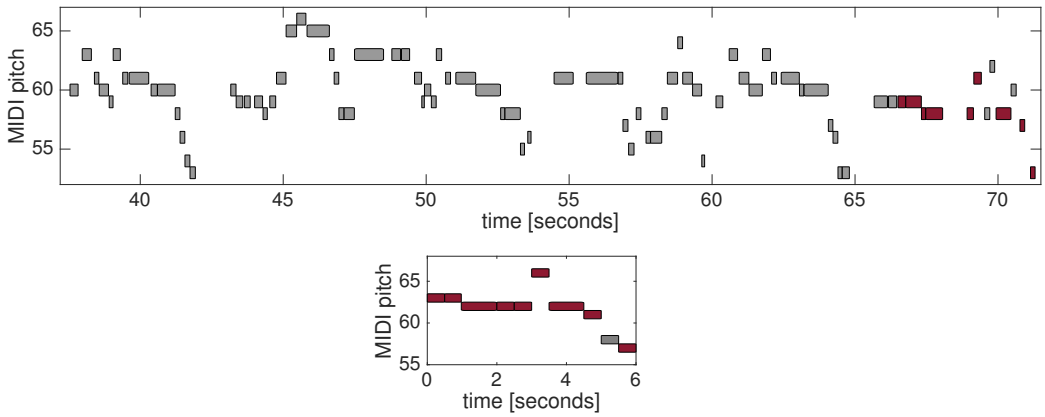


Fig. 3.5 Example of a query pattern detected in an automatic transcription. Notes marked in red form part of the alignment path.

3.3.1 Data

In this study, we use as query patterns manually defined melodies representative of the five phrases constituting the *fandango de Valverde* skeleton (Figure 3.8) and aim to retrieve their ornamented and modified occurrences in automatic transcriptions of real performances.

To this end, we gathered a collection of 20 *fandangos de Valverde* taken from commercial recordings. The *cante100* dataset [102] was added as noise to the collection: The contained 100 accompanied flamenco recordings cover a variety of singing styles and serve as a representative sample of flamenco music. None of the tracks contained in the *cante100* dataset belong to the *fandangos de Valverde* style. For each of the 120 tracks of the resulting collection we generated an automatic note-level transcription of the vocal melody using the algorithm described by [104].

3.3.2 Parameter settings

There are 4 user parameters to be set: G_v, G_h, E_1 and E_2. The search ranges for allowed note successors, G_v and G_h, are directly related to the amount of expected melodic variation and noise in the transcriptions. More specifically, G_v specifies the number of pitch values which can be skipped in the performance transcription between the alignment of two notes. Similarly, G_h determines the number of query pattern notes which can be skipped. Here, these parameters were set to $G_h = G_V = 2$, meaning that in both sequences at most one note can be located between two matched notes. For more improvisational styles, where a large amount of variation is expected, these values should be increased. The endpoint constraints were set to $E_1 = E_2 = 2$, meaning that the algorithm permits the first and last note of the query pattern to be excluded from the alignment.

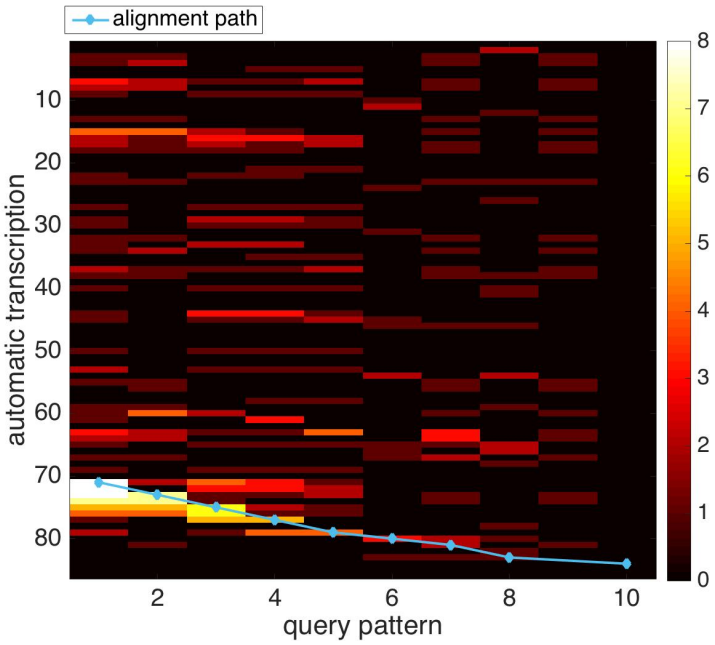


Fig. 3.6 Similarity grid **D**.

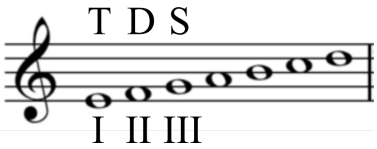


Fig. 3.7 The flamenco mode: The tonic is located on the first, the dominant on the second and the sub-dominant on the third scale degree.

3.3.3 Evaluation

Given that the occurrences of the patterns are not annotated in the audio recordings, we evaluate the method as a retrieval engine. In other words, we only evaluate, how many of the highest ranked results are relevant to the query. To this end, we compute the precision of the top 5 (P_5) and top 10 (P_{10}) ranking, where P_k represents the fraction of the k highest ranked results which are relevant to the query. A result is considered relevant if its origin is a *fandangos de Valverde* recording and the detected melodic sequence corresponds to the query phrase.

3.3.4 Results

Table 3.2 gives the quantitative evaluation of all five query patterns and the top 5 results for pattern A are shown in Figure 3.9. It can be seen that the percentage of relevant melodic sequences in the top ranked results is significantly higher for patterns A, A' and B compared

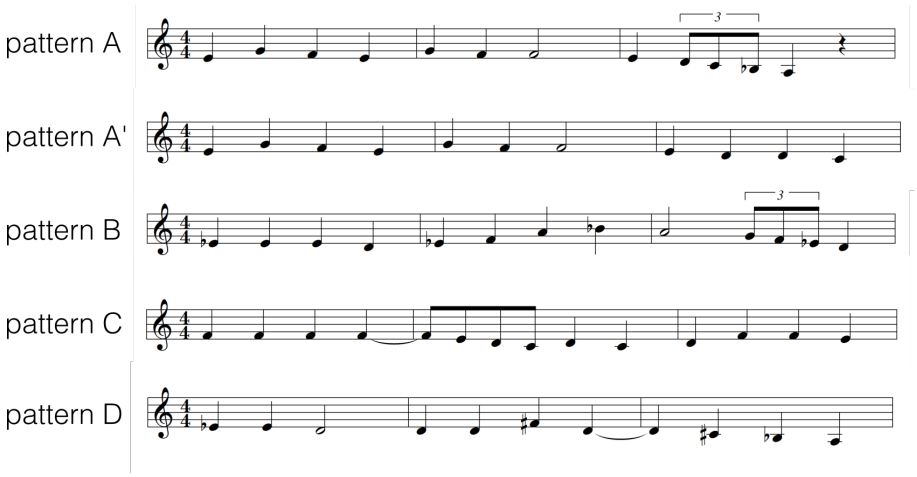


Fig. 3.8 Score representations of the query patterns.

query	P_5	P_{10}
A	80%	60%
A'	100%	70%
B	100%	70%
C	40%	40%
D	20%	10%

Table 3.2 P_5 and P_{10} measures among queries.

to patterns C and D. In particular, for patterns A' and B, all of the 5 highest ranked results are relevant with respect to the query, while for pattern D only one relevant result is retrieved.

A reasonable explanation for this behaviour is related to the amount of variation a pattern is subjected to during performance: Pattern D, referred to as *caída* in flamenco terminology, constitutes the end phrase and, at the same time, the musical "highlight" of the interpretation. During this phrase, the melody modulates from major mode to flamenco mode and resolves in the Andalusian cadence. Consequently, singers tend to apply more expressive resources, which result in a higher performance variance. Within a lesser extent, the same applies to pattern C, where a high degree of ornamentation, in particular prolongation through a sequence of grace notes, tends to appear during the last two bars. This numerical observation itself represents an interesting quantification of musicological concepts in flamenco music.

Four examples of manual MIDI transcriptions of *caídas* are shown in Figure 3.10 in order to highlight observed performance variation, free of possible transcription errors. Furthermore, automatic transcriptions are particularly prone to errors in the end of the singing voice section, since the guitar accompaniment tends to significantly increase in volume. As a result,

notes belonging to the singing voice melody might be missed and guitar notes might be transcribed instead.

Nevertheless, it can be seen from Figure 3.9 that the algorithm is capable of detecting ornamented and modified occurrences of a query pattern. It is also interesting to note that the obtained results contain a similar melodic sequence that was found in a recording of a different style (Figure 3.9 (b)), a *Bulería*. Despite this result being rated as not relevant in this task, it nevertheless demonstrates the potential of this tool for uncovering hidden structures and similarities in the context of large mining studies.

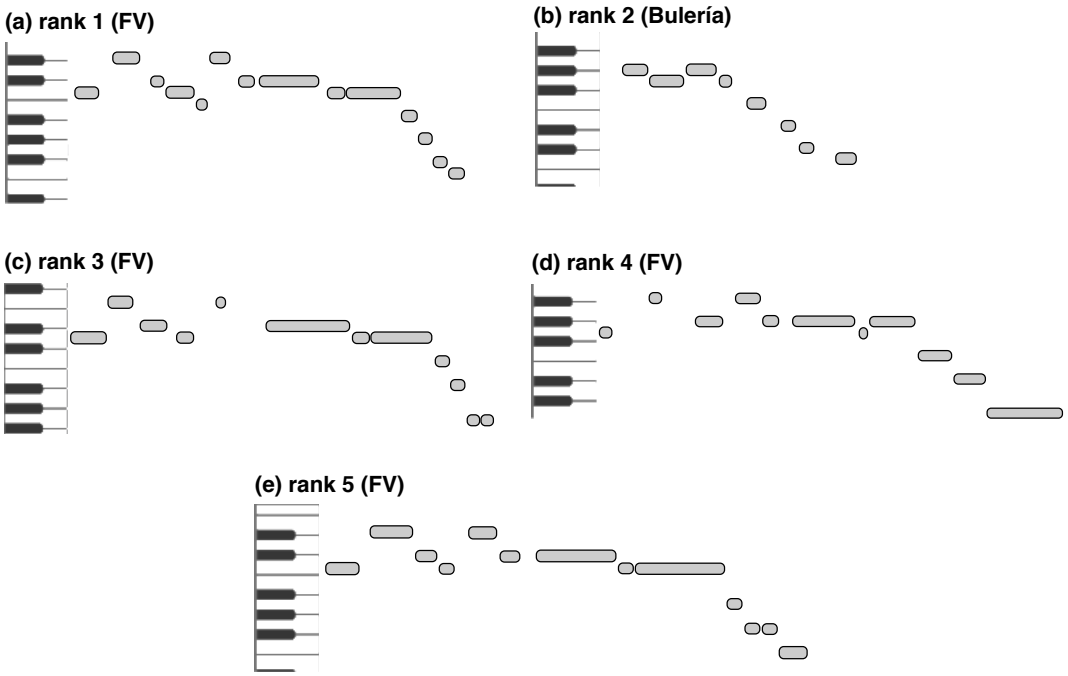


Fig. 3.9 MIDI representations of the top 5 results for query pattern A.

3.4 Conclusions

We presented an algorithm for the detection of ornamented instances of manually defined query patterns in automatic performance transcriptions and demonstrated examples of the capabilities and limitations of the system. Future extensions of this work could be directed towards the integration of the rhythmic domain into the alignment algorithm, in order to avoid the post-processing stage. Furthermore, the algorithm could be extended in such a way, that the system parameters are optimised within the dynamic programming scheme. In addition, the dataset can be augmented to include different styles and the application to other genres could be explored. Future applications are expected to include the incorporation of the algorithm in a framework for unsupervised pattern detection, the retrieval of typical

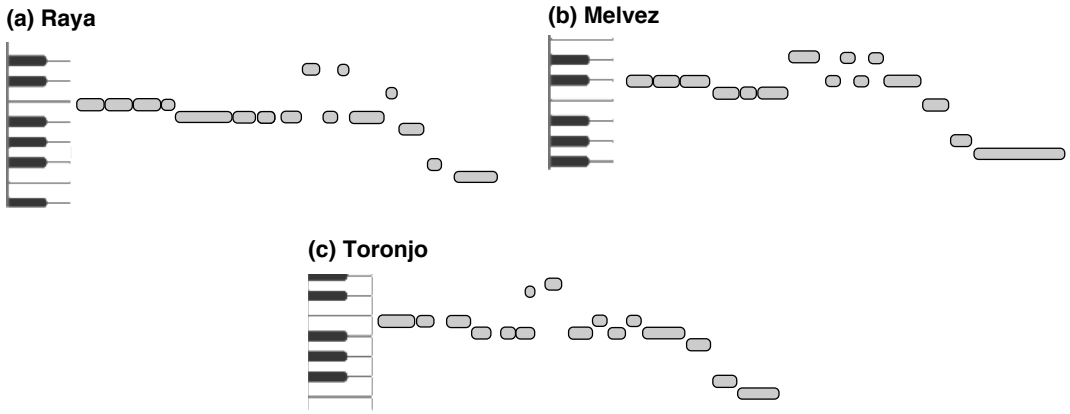
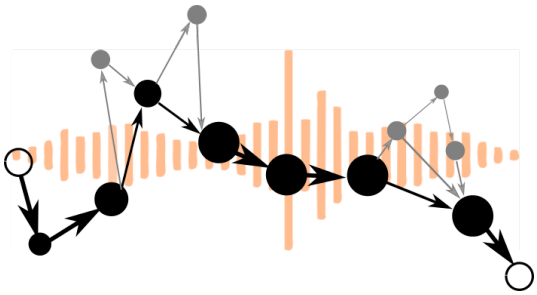


Fig. 3.10 Manual transcriptions of pattern D for three singers: (a) A. Raya, (b) M. Vélez and (c) P. Toronjo.

ornamentations from music recordings, the detection of short melodic guitar fragments (*falsetas*) in the melody of the singing voice and the investigation of preservation and evolution of specific patterns over time and across styles.

Chapter 4

Extraction of Melodic Templates



4.1 Introduction

The concept of reoccurring melodies in flamenco music has been introduced in Chapter 2 in the context of melody classification. There it was mentioned that, due to the oral transmission of flamenco music and the absence of written scores, the melodic templates on which the performed melodies are based, remain implicit and do not exist in any form of musical notation. The objective of the melody classification task is to categorise melodies according to their underlying melodic skeleton in supervised or unsupervised scenarios. In this chapter, we address a new computational problem related to the concept of reoccurring melodies in flamenco music: Based on a set of interpretations of the same melody, we aim to approximate a representation of the underlying template. In the sequel, we will refer to this task as **template extraction**. To the best of our knowledge, this is the first time this task is being approached from a computational perspective.

The automatic extraction of melodic templates can potentially find application in two scenarios: Supervised melody classification and expressive performance analysis. A major limitation to the scalability of supervised melody classification systems is the necessity for pair-wise comparisons with a large amount annotated melodies. Comparing an unlabelled melody instead with a single melodic sequence, which captures the essential melodic pro-

gression, can significantly reduce the computational complexity of this task. Expressive performance modelling [228] aims at creating computational models of the intentional shaping of musical dimensions by the musician during performance. Such models can either be used to automatically render a music score as an expressive performance [59], or to gain insight into the creative process itself (see [36] for an example). However, existing approaches usually assume the presence of a score and analyse expressivity by comparing the performance to the symbolic representation. However, given that flamenco performances are not score-based, most existing techniques are not directly applicable. Therefore, being able to automatically extract a template, which captures the commonalities of a set of performances, opens up new perspectives for expressive performance analysis in flamenco music.

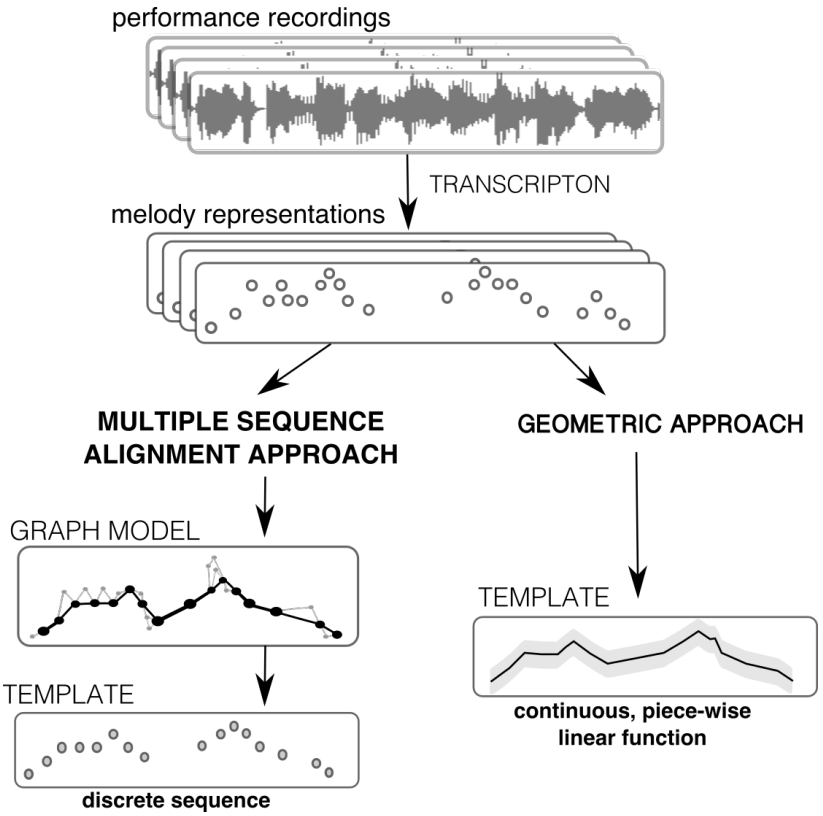


Fig. 4.1 Schematic illustration of the two template extraction approaches and their output discussed in this chapter.

In the scope of this thesis, two approaches towards melodic template extraction are proposed (Figure 4.1) which operate on note-level transcriptions of performance recordings. The two methods differ not only by the techniques used, but also by the resulting representation of the template and the amount of information contained in the computational model. The first approach (Section 4.3), which is heavily based computational geometry, aims at fitting a continuous function to a set of performances of the same melody which, in some optimal

sense, reflects the common melodic movement. The second method, described in Section 4.4, applies multiple sequence alignment techniques to create a graph model which captures both, commonalities and differences among the set of performance transcriptions. This model furthermore allows us to extract a discrete note sequence which approximates the melodic template. Both approaches are validated empirically in a case study on four *fandango* styles. The respective dataset is described in Section 4.2.

4.2 Data

The two methods proposed in this chapter are evaluated in a case study on four variants of the *fandango* style, a popular flamenco genre. Performances belonging to the same variant are characterised by a common melodic template which is subject to ornamentation and melodic variation, consciously introduced by the performer as an expressive resource. The four variants, named after their area of origin, are:

- *fandangos de Calaña* (FC)
- *fandangos de Valverde* (FV)
- *fandangos valientes de Alosno* (FVA)
- *fandangos valientes de Huelva* (FVH)

The two latter *valiente* variants usually exhibit more ornamentation and stronger variation compared to the *Calaña* and *Valverde* sub-styles, which in their performance aesthetic are closer to their folkloric origin. All four styles are performed and taught from memory without reliance on a written score. To the best of our knowledge, there do not exist any written scores of the underlying melodic templates.

We gathered a total of 40 performance recordings, 10 of each sub-style, taken from commercial studio productions, online audio sharing platforms and concert recordings. From the usually longer recordings, which encompass various repetitions of the same or different sub-styles, we manually extracted a single exhibition of the melody. All audio files are stored in a 44.1 kHz / 16 Bit stereo format and contain, apart from a solo singing voice, guitar accompaniment.

Furthermore, for each file, we automatically extracted a note-level transcription of the singing melody (AT) using the *CANTE* algorithm [104] and created manually corrected versions of the transcriptions (MT) where pitch, timing and voicing errors were corrected by a trained musician. In Section 4.3, only MT was used, whereas in Section 4.4 the use of both, AT and MT, is investigated.

As an example, a performance transcription (MT) of each of the styles is shown in Figure 4.2.

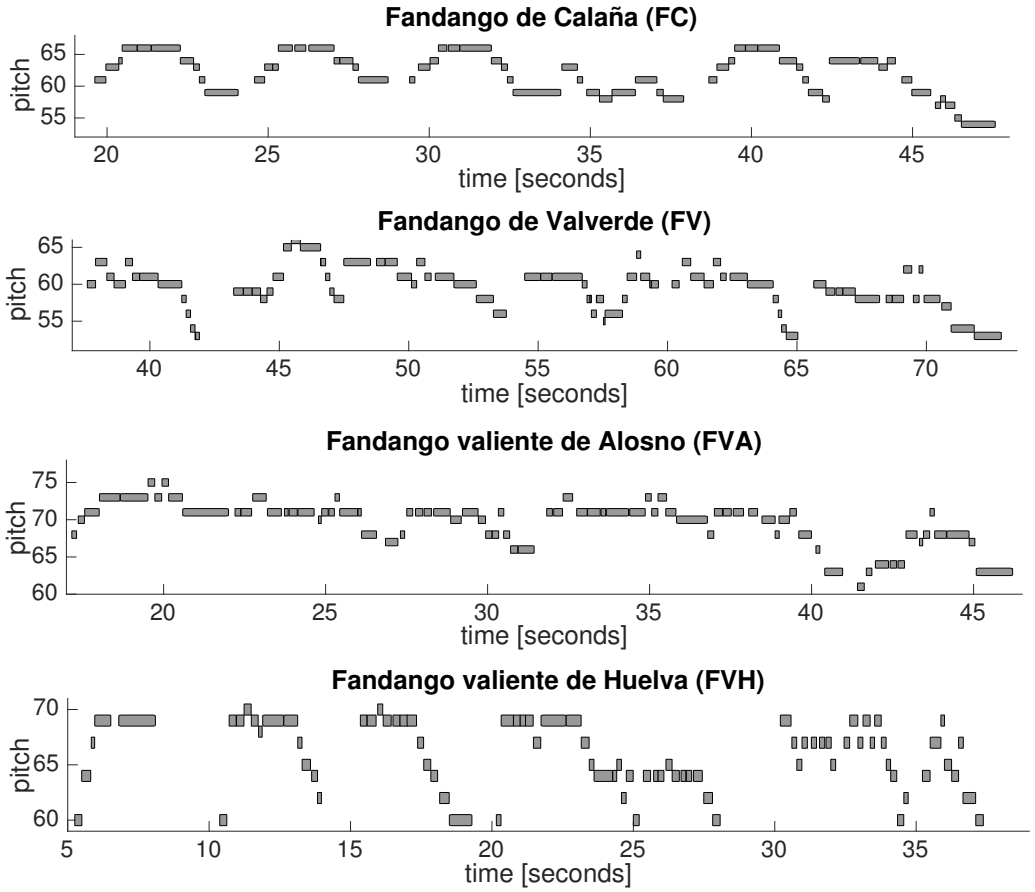


Fig. 4.2 Example performances (MT) of each of the four melodies considered in the case study. From top to bottom: FV performed by "El Cabrero"; FV performed by "El Raya"; FVA performed by "Camarón de la Isla" and FVH performed by Diego Clavel.

4.3 A geometric approach to template extraction

This section describes a geometric approach to automatic melodic template extraction. The goal is to approximate a continuous function of pitch over time which captures the common underlying melodic contour of a set of performance transcriptions. To this end, every performance transcription is modelled as a piecewise linear function, where vertices correspond to note events. Then, mathematically speaking, we are interested in approximating the set of n piecewise linear functions $f_i, i = 1, \cdots, n$ representing melodies in the time-pitch domain by a polygonal curve f^* so that, at any given point in time, f^* is in close proximity to an acceptable amount of the input melodies. A schematic view of this problem formulation depicted in Figure 4.3.

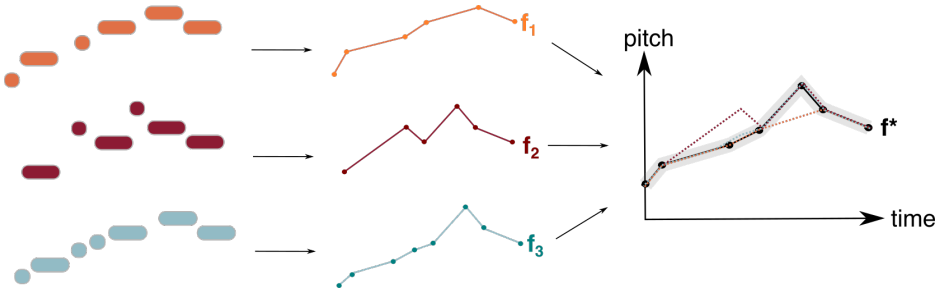


Fig. 4.3 Schematic illustration of the proposed geometric approach to template extraction.

4.3.1 Related work

Related tasks are polygonal approximation and curve fitting. The aim of polygonal approximation is to approximate a given complex polygonal curve by a simpler polygonal curve so that the total approximation error is minimised for a given error metric. Such methods have found application in computer vision, computer graphics, digital cartography, statistics, and data compression. Polygonal approximation is a well-studied problem which has been approached from various perspectives, such as geographic information systems [48], digital image analysis [87, 84] and computational geometry [24]. On the other hand, the fitting problems addresses the approximation of a discrete set of points by a polygonal curve or step function with a bounded number of line segments minimising a fitting measure [44, 60].

Unlike polygonal approximation and fitting problems, the input of our problem is a set of polygonal curves instead of a single polygonal curve or a set of points, and the constraint of the output is given by the number of neighbouring curves at any time in the time-pitch domain. To the best of our knowledge, this problem, which is motivated by the concept of constant re-interpretation of melodies in flamenco music, has so far not been studied in the literature.

4.3.2 Preprocessing

At a first stage, we convert each transcription, which is a discrete set of notes, described by their onset time, duration and MIDI pitch value, to a point set representation $A = \{a_1, a_2, .., a_m\}$ where a note $a_i = \{(x_i, y_i)\}$ is characterised by its onset time relative to the last note onset, $x_i \in [0, 1]$, and its MIDI pitch value, y_i. Given that the transcription tool does not provide any rhythmic or metric quantisation (which in flamenco, even when done manually, is a non-trivial task), the relative representation of the onset time is chosen to compensate for strong tempo variation among performances.

Furthermore, variants of the same melody may be performed in different keys. Singers usually select a key according to their individual pitch range. In order to meaningfully compare

the pitch values of a set of performances, we therefore need to perform key normalisation. Here, we again apply the procedure described in Section 2.2 to shift all transcriptions to a common reference key.

In addition to key transposition, variants of the same melody may exhibit strong rhythmic distortions. We therefore perform a rhythmic normalisation as follows. Given n transcriptions of a variant, we select one transcription A_{ref} as a temporal reference to which align all remaining transcriptions. This is realised with the *Needleman-Wunsch* algorithm [143] which finds an optimal matching with gaps among two sequences. The algorithm was described in detail in Section 3.2. The output of the NW algorithm is an alignment path, which assigns notes of one sequence either to notes of the other sequence, or to gaps. The onset of notes of a given transcription, which have been matched with notes of A_{ref}, are moved to the position of the respective onset of the matched note. We use linear interpolation to assign an onset time for notes, which have not been matched to other notes, but to gaps in A_{ref}. A more detailed description of this stage will be given in Section 4.4. Figure 4.4 shows an example of five transcriptions before and after the temporal normalisation.

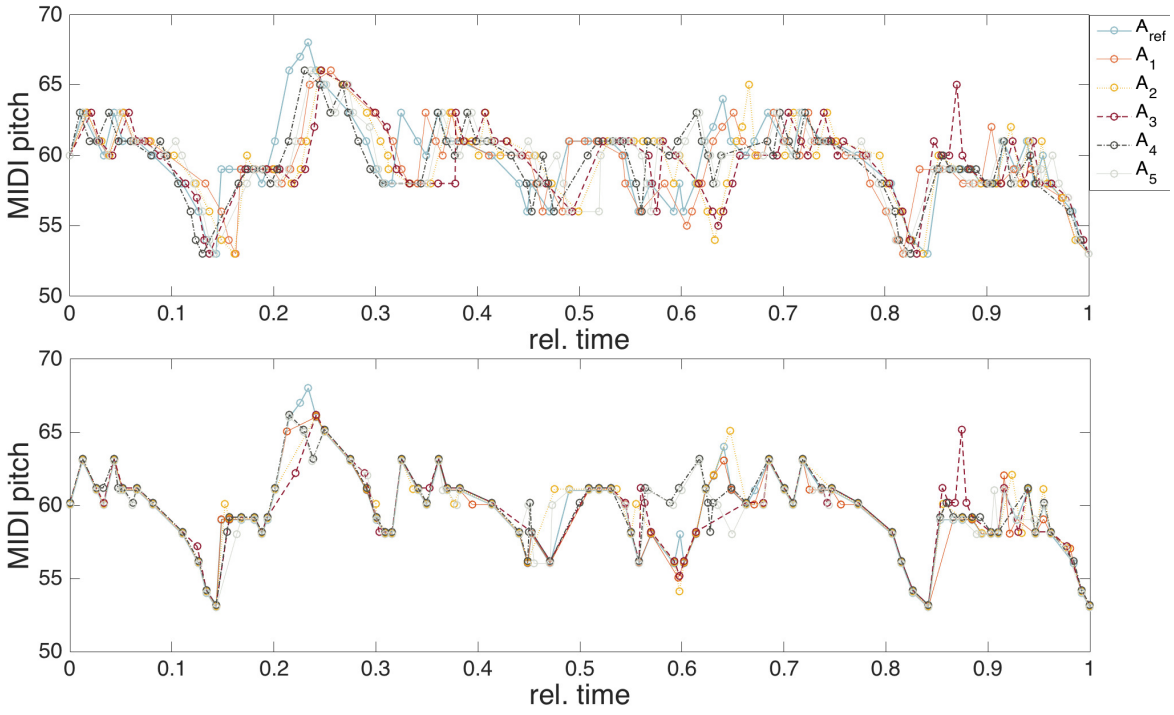


Fig. 4.4 Five performance transcriptions before (top) and after (bottom) temporal normalisation.

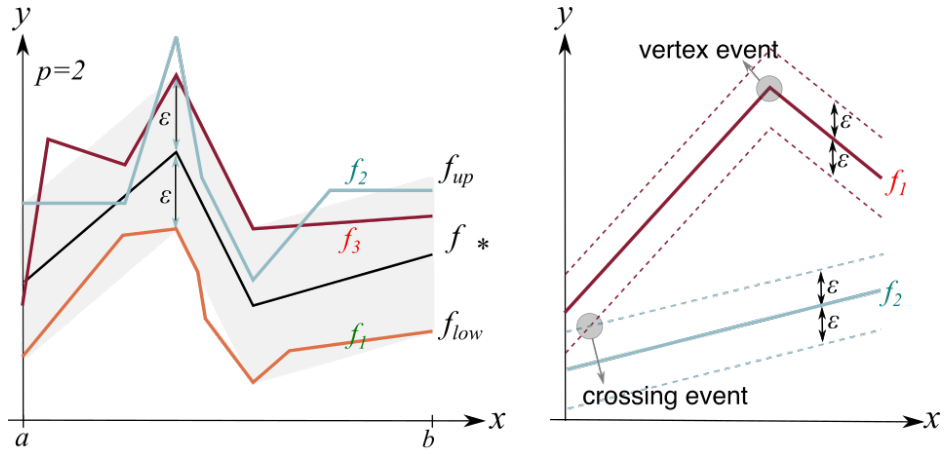


Fig. 4.5 Left: Schematic illustration of the *Spanning Tube Problem*: At any given point in time, at least $p = 2$ out of $n = 3$ functions are located inside the spanning tube; Right: Examples of vertex and crossing events.

4.3.3 The geometric problem

In order to estimate a melodic template, we aim to find a continuous function, that is monotone with respect to time, which represents a set of n temporally aligned performance transcriptions given by a set of x-monotone polygonal curves. We consider that the data under study may contain local outliers and consequently some notes could be located far from the template. Furthermore, our goal is to quantitatively assess the amount of melodic variation which occurs across performances. Geometrically, we view each of the n temporally aligned performance transcriptions as a polygonal curve in the plane with the time axis and the pitch axis. Since they are monotone with time, we call them x-monotone curves. Typically, the performance transcriptions are discrete and we make an assumption that the curves are polygonal. These objectives and considerations give rise to a geometrical problem, which we call the *Spanning Tube Problem*.

Definition 1. *Given $a, b \in \mathbb{R}$ and a continuous function $f(x)$ with domain $[a, b]$, we define the ε-tube of f, $T(f, \varepsilon)$, as the locus of points (x, y) s.t. $(x, y) \in [a, b] \times [f(x) - \varepsilon, f(x) + \varepsilon]$.*

The Spanning Tube Problem (STP): *Let $a, b \in \mathbb{R}$, with $a < b$; let $n, m, p \in \mathbb{Z}^+$; and for $i = 1, \cdots, n$, let $f_i : [a, b] \to \mathbb{R}$ be a piecewise linear function with at most m links. Given $p \leq n$, find minimum $\varepsilon^* > 0$ such that there exists a continuous function $f^*(x)$ fulfilling that, for each $x \in [a, b]$ the vertical segment of length $2\varepsilon^*$ centered at $(x, f^*(x))$ intersects at least p functions.*

In other words, we are seeking a piece-wise linear function f^*, for which, at any given point in time, at least p out of n functions are within a range of ε^* from f^*.

Remark 2. *Note that the p intersected functions are not necessarily the same in every point in time and the template f^* has to be continuous and t-monotone.*

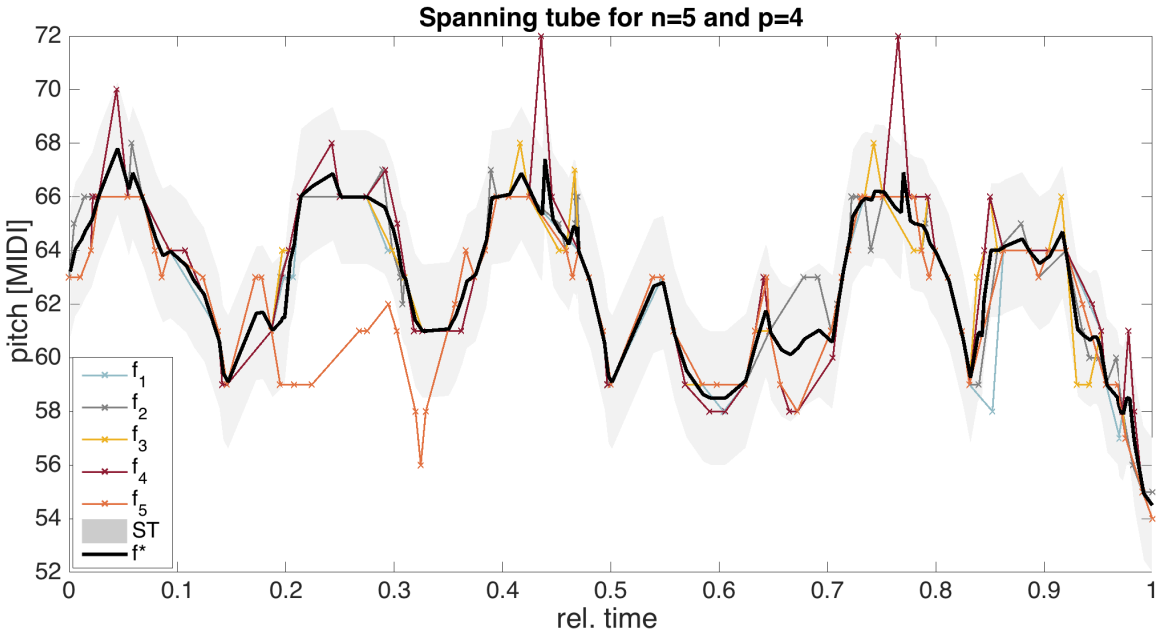


Fig. 4.6 The $\varepsilon-$tube for *Fandango de Calaña (n=5, p=4, $\varepsilon = 2.5$).*

Definition 3. *Given a collection of n continuous functions f_i, let f_{up} be the upper and f_{low} be the lower envelope enclosing the n functions. Let f_{median} be the point-wise median of the f_i's.*

Remark 4. *For $n = p$,*

$$\varepsilon^* = \max \frac{f_{up}(x) - f_{low}(x)}{2}$$

and

$$f^*(x) = \frac{f_{up}(x) + f_{low}(x)}{2}$$

for $x \in [a, b]$.

Figure 4.5 (left) provides a schematic illustration of the *Spanning Tube Problem.* An example of the spanning tube for five performance transcriptions is shown in Figure 4.6.

Solving the general problem

The following result can be obtained by using a left-to-right linear sweeping:

Theorem 5. *The decision problem for the STP can be solved in $O(n^2 m \log n)$ time.*

Proof. Let $f_1(x), f_2(x), \ldots, f_n(x)$, $x \in [a, b]$ be n given functions and let ε be a parameter. Our task is to decide whether $\varepsilon < \varepsilon^*$ or $\varepsilon \geq \varepsilon^*$. This can be accomplished by sweeping the

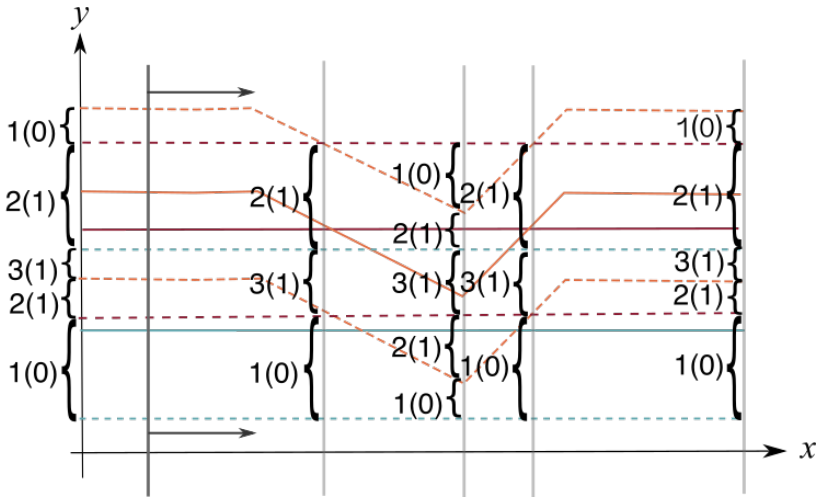


Fig. 4.7 Example of maintaining $Count(int)$ and $Ind(int)$ for the intervals between two consecutive functions of type $f_i(x) \pm \varepsilon$ with $p = 2$. Curly braces indicate the intervals int. The annotated integer refers to $Count(int)$ and the number in parenthesis denotes the boolean variable $Ind(int)$.

slab between the lines with equation $x = a$ and $x = b$ by a vertical line. During this process, we maintain two sorted lists. Let L_f be the list of functions $f_i(x) \pm \varepsilon$ intersecting the sweep line and let L_e be the list of events where the sweep line stops. There are two types of events in L_e, see Figure 4.5 (right):

Vertex event. The sweeping line at a vertex event has an equation $x = c$ where one of the functions $f_i(x)$ changes from one linear function to another. Each given function contributes one vertex event to the list of events.

Crossing point event. At this event one of the functions $f_i(x) - \varepsilon, f_i(x) + \varepsilon$ intersects one of the functions $f_j(x) - \varepsilon, f_j(x) + \varepsilon$, for some $i < j$.

In addition, using L_f, we store a number $Count(int)$ for each interval int between two consecutive functions crossing the sweep line which indicates the number of intervals of form $[f_i(x) - \varepsilon, f_i(x) + \varepsilon]$ intersecting int. If $Count(int) \geq p$ we also store a boolean $Ind(int)$ indicating whether there is a continuous function $f(x)$ such that $\forall x \in [a, x_0]$ there are at least p functions in $T(f, \varepsilon)$ where x_0 is determined by the current position of the sweep line. An example is shown in Figure 4.7.

We also maintain the ranges of form $[f_i(x) + s_i\varepsilon, f_j(x) + s_j\varepsilon]$, where $s_i, s_j \in \{-1, 1\}$, containing values of y such that, for current $x = c$ and any y in the range, there is a continuous function $f(x)$ with domain $[a, c]$ and $f(c) = y$. Note that the continuity of the required function is guaranteed by computing the values of $Count(int)$ and $Ind(int)$.

Observe that L_e can be updated in $O(\log n)$ time and the order of functions $f_i(x) \pm \varepsilon$ is fixed between two events (Figure 4.8). For any fixed i and j, there are at most $O(m)$ crossing

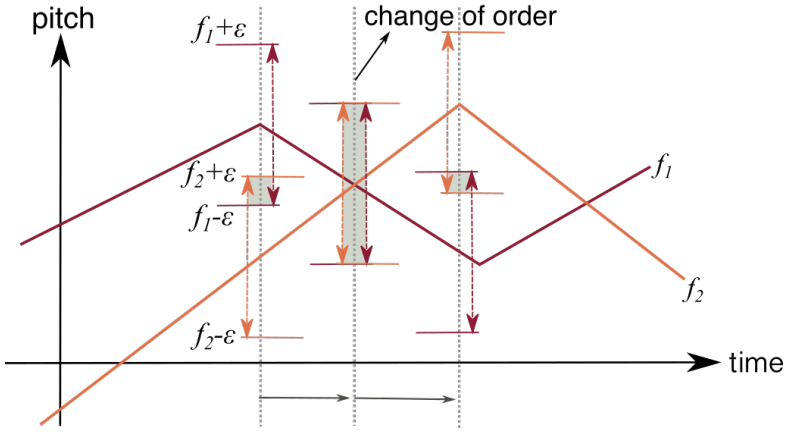


Fig. 4.8 The order of functions $f_i(x) \pm \varepsilon$ changes at a crossing event.

points events. The total number of events is $O(n^2m)$ and the list of events has size $O(n)$. Since each event can be processed in $O(\log n)$ time, the running time is $O(n^2m \log n)$.

□

Based on the result described above, we can use bisection to compute an approximate solution. However, if we dispose of a discrete set of candidate values for ε^*, an exact solution can be found. The following lemma gives us the discrete set of candidate values for the optimisation problem (STP).

Definition 6. *We define a vertex of a piecewise linear function as a vertex-point and an intersection between two functions as an intersection-point.*

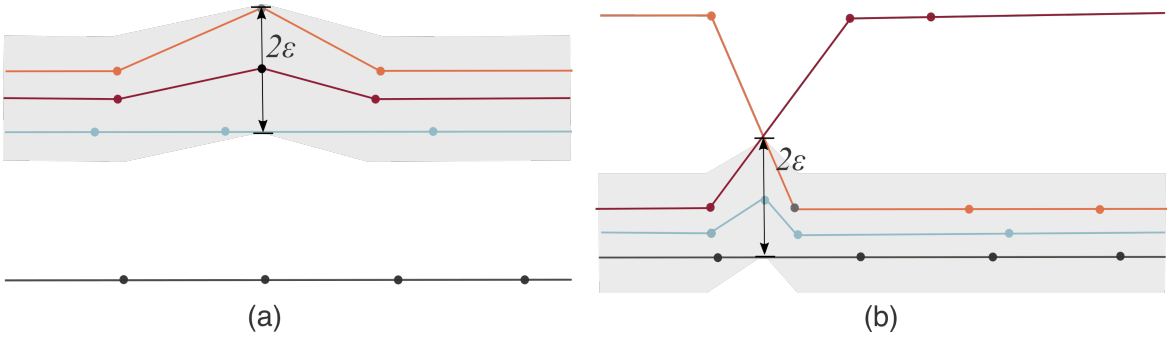


Fig. 4.9 Two scenarios for determining the candidate values: (a) $2\varepsilon^*$ corresponds to the distance between a vertex of a function f_i and the value of another function f_j at the same time instance; (b) $2\varepsilon^*$ corresponds to the distance between the intersection of two functions f_j and f_k and the value of another function f_i at the same time instance.

Lemma 7. *For $n > 2$ and $1 < p < n$, let ε^* be the optimal value for the optimization problem. Then, $2\varepsilon^*$ is equal to one of the values given by the vertical distance between a vertex- (Figure 4.9 (a)) or an intersection-point (Figure 4.9 (b)) and its $(p-1)$ or p-nearest function.*

Proof. Let ε^* be the minimum value such that there exists a continuous function $f^*(x)$, where $\forall x \in [a, b]$ there are at least p functions in $T(f^*, \varepsilon^*)$. Then, there are two functions f_i and f_j in $T(f^*, \varepsilon^*)$ and $x_0 \in [a, b]$ such that $f_i(x_0) = f^*(x_0) + \varepsilon^*$ and $f_j(x_0) = f^*(x_0) - \varepsilon^*$. Otherwise, we can decrease the width of the tube contradicting optimality. It is easy to prove that the intersection points between the functions f_i and f_j and the boundary of the tube must have the same abscissa. There are three cases: the point $u = (x_0, f_i(x_0))$ is a vertex-point, a crossing point, or none of the above. Analogously, the same holds for $v = (x_0, f_j(x_0))$.

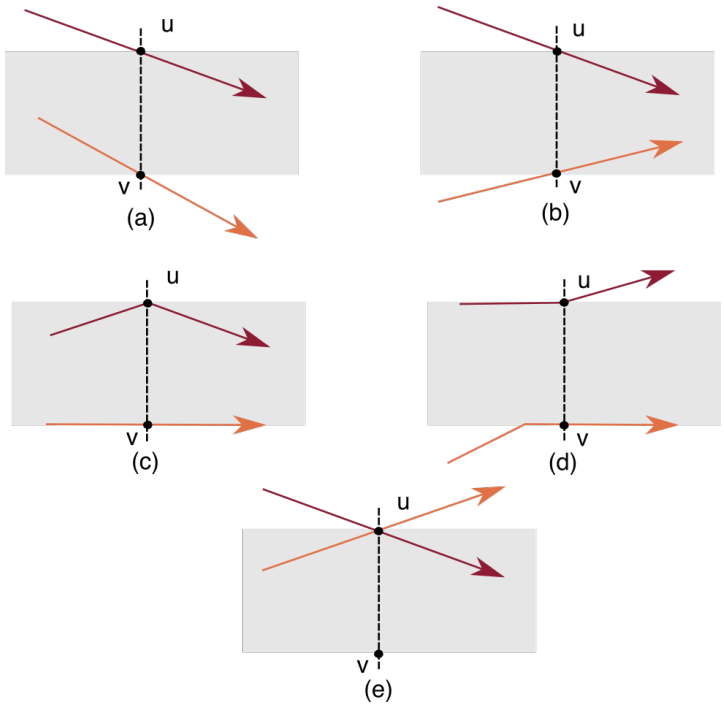


Fig. 4.10 (a) and (b): u and v are neither vertex nor crossing points. (c) and (d): u is a vertex point. (e): u is a crossing point.

Suppose that u and v are neither a vertex nor crossing points. Then, up to symmetry, we have two cases as shown in Figure 4.10 (a) and (b). In both cases, the width of the tube can be slightly decreased, while maintaining continuity. This fact contradicts the optimality condition. In the case shown in Figure 4.10(a), we can move the endpoints u and v on f_i and f_j respectively, while maintaining vertical the segment uv in the direction in which the length of the vertical segment decreases. Then, there exists an instant in which a vertex or crossing point is found, and this case reduces to the vertex or crossing point event. Note, that even

when f_i and f_j are parallel, theses events are found if we maintain at least p functions inside the tube. A similar reasoning can be applied in the second case (in Figure 4.10(b)): since the part of the slab to the left of the vertical segment uv must have at least p functions, then in the right part there are at least $p+2$ functions. Consequently, there exists a narrower tube containing p or more functions.

Consider now the case in which u is a vertex-point of $f_i(x)$ (w.l.o.g. u lies on the top boundary of the tube). Depending on the slopes of the neighbouring segments, we can distinguish two cases, which are displayed in Figure 4.10 (c) and (d).

1. The function $f_i(x)$ is located inside the tube for x in some neighbourhood of x_0, as shown in in Figure 4.10(c). In this case, $f_j(x)$ is the $p-1$-nearest function of u.

2. Figure 4.10(d) shows the case where, for $\delta > 0$ and $x \in (x_0, x_0+\delta)$ or $x \in (x_0-\delta, x_0)$, $f(x)$ is not located inside the tube. The function $f_i(x)$ on the top boundary leaves the tube at time x_0 and consequently, function $f_j(x)$ is the p-nearest function.

Finally, the case in which u is an intersection-point can be easily reduced to the vertex-point type (see Figure 4.10(e)). □

Theorem 8. *For $n > 2$ and $p > 1$, the optimization problem can be solved in $O(n^2 m \log n \log nm)$ time.*

Proof. The main steps of the algorithm can be summarised as follows:

(I) Compute the candidate values.

(II) Sort the candidates.

(III) Perform binary search using the decision algorithm.

Step (I) can be accomplished by sweeping the arrangement of the n functions with a vertical line. In this status line we maintain the order of the intersection of the line with the functions (these intersections can be vertex-points, intersection-points and function crossings) in order to compute in $O(1)$ the corresponding $(p-1)$- or p-nearest functions (one function above and the other below) for every candidate. The overall complexity of the sweep is $O(n^2 m \log n)$.

Step (I) can be performed in $O(n^2 m \log n)$ time, Step (II) requires $O(n^2 m \log nm)$ time and Step (III) in $O(n^2 m \log n \log nm)$ time. □

An efficient solution for a particular case

For a particular case, when $n = 3$ and $p = 3$, we can report the following solution.:

Lemma 9. *Set $n = 3$ and $p = 2$ and let ε^* be the optimal value for the optimisation problem. Then the value of $2\varepsilon^*$ results to of the following values:*

- *a local maximum value of* $f_{up} - f_{median}$ *or*
- *a local maximum value of* $f_{median} - f_{low}$ *or*
- *a local minimum value of* $f_{up} - f_{low}$.

Proof. For $\varepsilon > 0$, let $R_1(\varepsilon) = T(f_{up}, \varepsilon) \cap T(f_{mean}, \varepsilon)$ and let $R_2(\varepsilon) = T(f_{low}, \varepsilon) \cap T(f_{mean}, \varepsilon)$. Observe that a solution exits for ε if and only if there exists a monotone function inside the region $R_1(\varepsilon) \cup R_2(\varepsilon)$. Now, a solution exists for $R_1(\varepsilon^*) \cup R_2(\varepsilon^*)$ but none within $R_1(\varepsilon) \cup R_2(\varepsilon)$ for any $\varepsilon < \varepsilon^*$. Note that $R_i(\varepsilon) \subset R_i(\varepsilon^*)$ for $i = 1, 2$. We can distinguish 3 cases:

- Two points in $R_1(\varepsilon^*)$ are connected by a function but are not connected in $R_1(\varepsilon)$. Then, for some $x^* \in (a, b)$, we have $f_{up}(x^*) - \varepsilon = f_{mean}(x^*) + \varepsilon$ and $f_{up}(x) - \varepsilon < f_{mean}(x) + \varepsilon$ for all $x \neq x^*$ in some neighborhood of x^*. Therefore $f_{up} - f_{mean}$ has a local maximum value at x^* with value $2\varepsilon^*$.
- Two points in $R_2(\varepsilon^*)$ are connected by a function but are not connected in $R_2(\varepsilon)$. Then, for some $x^* \in (a, b)$, we have $f_{mean}(x^*) - \varepsilon = f_{lower}(x^*) + \varepsilon$ and $f_{mean}(x) - \varepsilon < f_{low}(x) + \varepsilon$ for all $x \neq x^*$ in some neighborhood of x^*. Therefore $f_{mean} - f_{low}$ has a local maximum value at x^* with value $2\varepsilon^*$.
- Two points, one in $R_1(\varepsilon^*)$ and one in $R_2(\varepsilon^*)$, are connected in $R_1(\varepsilon^*) \cup R_2(\varepsilon^*)$ but are not within $R_1(\varepsilon) \cup R_2(\varepsilon)$. Then, for some $x^* \in (a, b)$, we have $f_{up}(x^*) - \varepsilon = f_{low}(x^*) + \varepsilon$ and $f_{up}(x) - \varepsilon > f_{low}(x) + \varepsilon$ for all $x \neq x^*$ in some neighborhood of x^*. Therefore $f_{up} - f_{low}$ has a local minimum value at x^* with value $2\varepsilon^*$.

□

By Lemma 9, we have $O(m)$ candidates and consequently, by performing binary search, the optimal solution can be computed in $O(m \log m)$ time by using the decision algorithm. We will now prove some properties, which will subsequently yield and improvement of this runtime.

Definition 10. *Let* $x_0 = a, x_1, \ldots, x_t = b$ *be the events defined by vertices and intersection points of the functions. Consider a slab between two events at* x_i *and at* x_{i+1}. *Suppose that a spanning tube* T *in* $[x_0, x_{i+1}]$ *covers* $f_{up}(x_i), f_{median}(x_i)$ *and* $f_{low}(x_{i+1}), f_{median}(x_{i+1})$. *We refer to this scenario as the tube making a* ***transition HL*** *in the* i*-th slab.*

Lemma 11 (Two trapezoids)**.** *Suppose a spanning tube* T *makes a transition HL in the* i*-th slab. Then, there exists* $x \in [x_i, x_{i+1}]$ *such that* $f_{up}(x), f_{median}(x)$ *and* $f_{low}(x)$ *are covered by* T. *Furthermore, the width of* T *is at least*

$$w = \max(\varepsilon_1^i, f_{up}(x) - f_{low}(x), w_2^{i+1}) \tag{4.1}$$

and there exists a tube of width w *using two trapezoids in the slab (see Figure 4.11).*

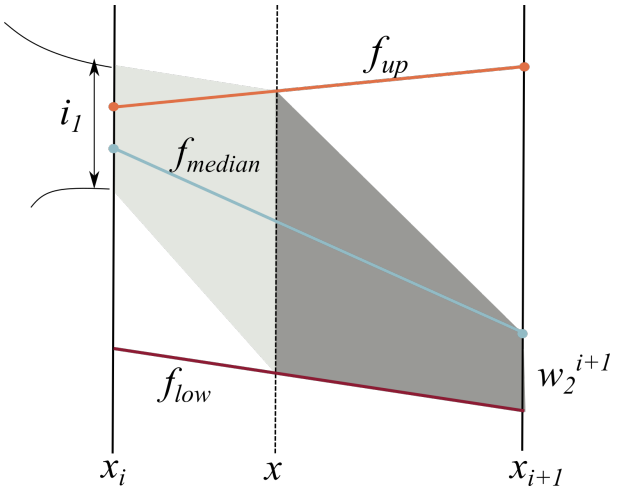


Fig. 4.11 Two trapezoids: The tube makes a transition in the i-th slab.

Proof. The existence of x can be demonstrated through the continuity of the tube. Since all 3 functions are linear in the slab, the first trapezoid covers f_{up} and f_{median}. Similarly the second trapezoid covers f_{median} and f_{low}. The lemma follows since the width of the tube formed by two trapezoids is w. □

Lemma 12 (Transition)**.** *The smallest width of a spanning tube T making a transition HL in the i^{th} slab is equal to*

$$\max(\varepsilon_2^i, \min(w^i, w^{i+1}), w_1^{i+1}).$$

*The vertical line for the transition can be selected by comparing w^i and w^{i+1}. Thus, there exists an optimal tube having **all** transitions at the events only.*

Proof. This can be demonstrated by varying x between x_i and x_{i+1}. The smallest value of the second term in Equation 4.1 is $\min(w^i, w^{i+1})$. Thus, if $f_{up}(x)$ and $f_{low}(x)$ are not parallel, we can select the line for the transition ($x = x_i$ or $x = x_{i+1}$ using the smallest value of w^i and w^{i+1}). If they are parallel, one can choose line $x = x_i$ (or $x = x_{i+1}$) for the transition. □

Theorem 13 demonstrates that we can solve the problem in linear time.

Theorem 13. *The optimisation problem for $n = 3$ and $p = 2$ can be solved in $O(m)$ time.*

Proof. Let $x_0 = a, x_1, \ldots, x_t = b$ be the events defined by vertices and intersection points of the functions. For each event at $x = x_i$, we compute ε_1^i and ε_2^i for the spanning tubes in $[x_0, x_i]$ covering $[f_{up}(x_i), f_{median}(x_i)]$ and $[f_{median}(x_i), f_{low}(x_i)]$, respectively. Let $w_1^i = f_{up}(x_i) - f_{median}(x_i)$ and $w_2^i = f_{median}(x_i) - f_{low}(x_i)$. At the beginning, $\varepsilon_j^0 = w_j^0$ for $j = 1, 2$. Then ε_1^{i+1} can be computed using two cases where the tube covers (i) $f_{up}(x_i)$ or (ii) $f_{low}(x_i)$. Then ε_1^{i+1} is the minimum of the two. The second value is computed using Lemma 12 and

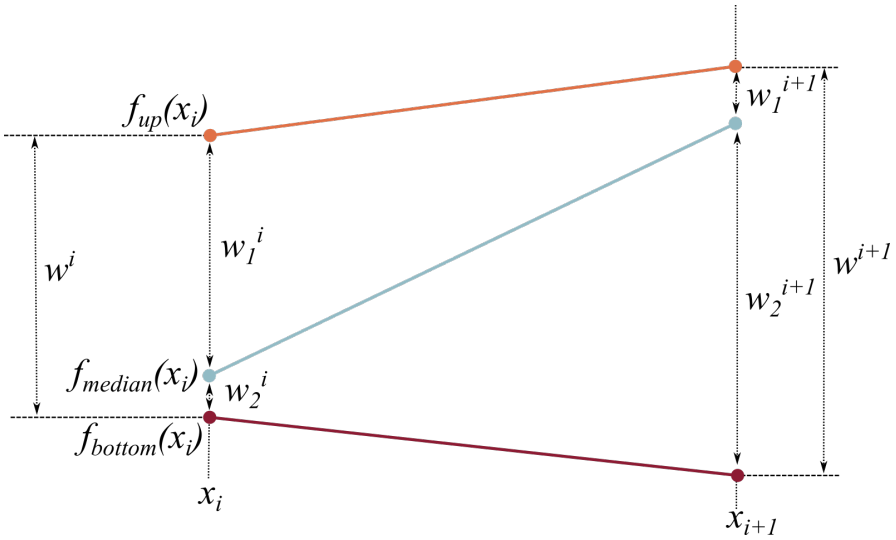


Fig. 4.12 Updating the optimal tube in the slab between the events x_i and x_{i+1} for $n = 3$, $p = 2$.

$w^i = f_{up}(x_i) - f_{low}(x_i)$, see Figure 4.12. Thus,

$$\varepsilon_1^{i+1} = \min(\max(\varepsilon_1^i, w_1^{i+1}), \max(\varepsilon_2^i, \min(w^i, w^{i+1}), w_1^{i+1})). \tag{4.2}$$

Similarly,

$$\varepsilon_2^{i+1} = \min(\max(\varepsilon_2^i, w_2^{i+1}), \max(\varepsilon_1^i, \min(w^i, w^{i+1}), w_2^{i+1})). \tag{4.3}$$

Finally, $\varepsilon^* = \min(\varepsilon_1^t, \varepsilon_2^t)$.

Running time. For each event $i + 1$, the values of $w^{i+1}, w_1^{i+1}, w_1^{i+2}$ (see Figure 4.12) and the values of $\varepsilon_1^{i+1}, \varepsilon_2^{i+1}$ (by Equations (4.2) and (4.3)) can be computed in constant time. Since we have $O(m)$ events, the overall running time is linear. □

Remark 14. *Note that Theorem 13 improves the complexity obtained in Theorem 8 for* $n = 3$.

4.3.4 Case study: Quantifying melodic variation

In this Section, we exemplify the use of the proposed method in a comparative performance analysis of the four *fandango* styles investigated in this study. In particular, we aim to quantify the amount of melodic variation the skeleton is subjected to during performance, which we compare across styles and across phrases of the same style.

From a music theoretic standpoint we know, that among the four styles under study, the *Fandangos de Calaña* and the *Fandangos de Valverde* are close to their folkloric origin and performers tend to largely preserve the melodic skeleton during performance. In the *Fandangos Valientes de Huelva* and the *Fandangos Valientes de Alosno* on the other hand,

Style	$\varepsilon = 1.0$	$\varepsilon = 2.0$	$\varepsilon = 3.0$
Fandango de Calaña	0.2	0.5	0.8
Fandango de Valverde	0.1	0.3	0.5
Fandango Valiente de Alosno	0.1	0.3	0.3
Fandango Valiente de Huelva	0.0	0.1	0.1

Table 4.1 Fraction of performances enclosed by the tube for different styles.

performers tend to use heavy melodic ornamentation as an artistic asset, resulting in a more distorted version of the skeleton.

The proposed method allows us to quantify the amount of melodic variation occurring in a set of performance transcriptions by fixing the parameter ε and determining, in a decision problem, the maximum fraction of performances $\frac{p}{N}$ which can be enclosed by the tube. Here, we compute this value for each of the four styles under study, where ε is varied between 1.0 and 3.0.

The largest differences among styles can be observed for $\varepsilon = 3.0$. Note, that in this case, the tube covers 6 semitones, which corresponds to half an octave. Consequently, melodic segments outside the tube correspond to relatively large deviations from the basic melodic contour. For this case, the results (Table 4.1) confirm the musicological considerations described above: For the *Fandangos de Calaña*, 80% of the analyzed performance transcriptions can be enclosed by the tube with $\varepsilon = 3.0$, indicating that performers largely follow the underlying skeleton. The value for the *Fandangos de Valverde* is slightly lower with $\frac{p}{N} = 0.5$. The two *valiente* styles show a significantly higher amount of melodic variation. For the *Fandangos Valientes de Huelva* only 10%, and for the *Fandangos Valientes de Alosno* only 30% of the transcriptions are enclosed by the tube. Figure 4.13 shows the aligned transcriptions together with the computed templates using the values for $\frac{p}{N}$ from table 4.1. A similar trend can be observed for smaller values of ε.

However, analysing the transcriptions in relation to the maximum spanning tube (Figure 4.13), reveals that there exist local differences in the amount of occurring variation. For example, for the *Fandangos Valientes de Alosno*, all transcriptions are located inside the 3ε-tube from the beginning of the melody until approximately 0.3 on the relative time axis. In order to obtain a finer granularity of the amount of occurring variation, we therefore repeat the previous experiment on a phrase level. More precisely, each of the recordings contains 6 musical phrases, which we manually annotated. Fixing $\varepsilon = 3.0$, we solve the optimization problem and compute $\frac{p}{N}$ for each phrase separately.

The results in Table 4.2 show, that with exception of the *Fandango de Valverde*, the last phrase tends to exhibit a high amount of variation compared to other phrases. As mentioned in Chapter 3, this phrase, which is referred to as *caída* in flamenco jargon, represents the

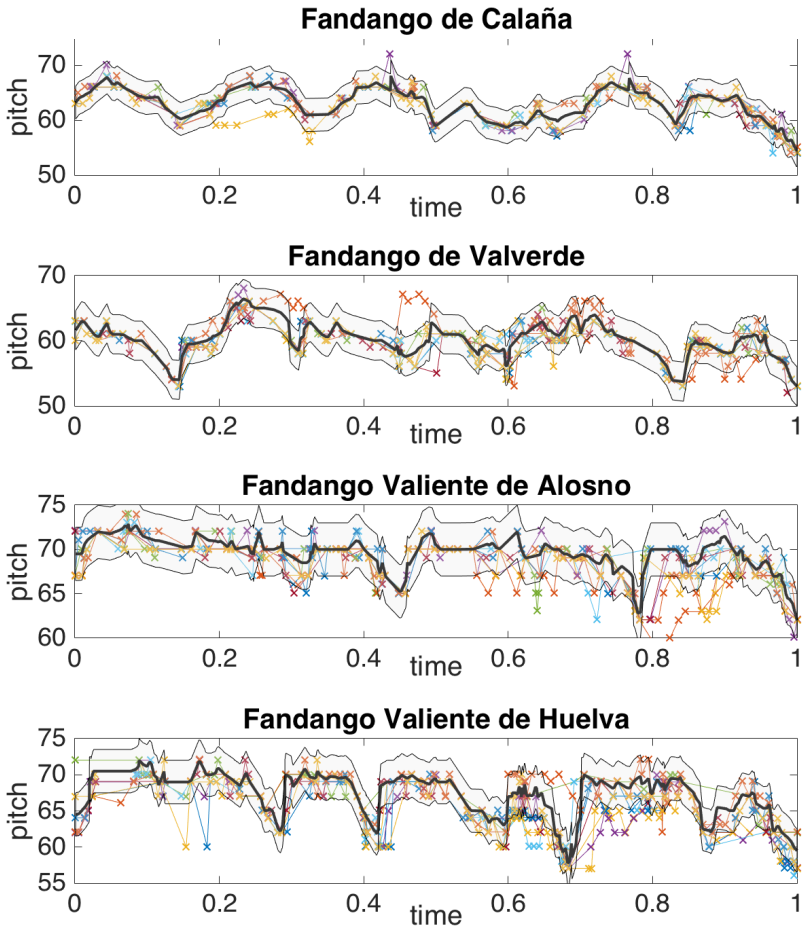


Fig. 4.13 Aligned transcriptions and computed skeleton for the four *fandango* styles under study for $\varepsilon = 3.0$.

highlight of the flamenco performance, and consequently, it is likely that performers use a larger amount of ornamentation and variation as an expressive asset. For all styles, we furthermore observe a relatively high amount of variation for the fourth phrase.

These observations are of interesting from a musicological viewpoint and give rise to several lines of study related to melodic variation and expressiveness in flamenco music. It should however be noted that the proposed metric mainly assess the amount of variation based on the pitch distance between grace notes and the template. A fast semitone melisma for example would not cause particularly low values for the melodic stability measure defined here. This poses a limitation for reliability of the application of the proposed algorithm in the context of musicological studies. These shortcomings are addressed by the algorithm in Section 4.4.

Style	phrase 1	phrase 2	phrase 3	phrase 4	phrase 5	phrase 6
Fand. de Calaña	0.5	0.7	0.7	0.5	0.7	0.4
Fand. de Valverde	0.5	0.5	0.2	0.2	0.3	0.5
Fand. Val. de Alosno	0.3	0.4	0.2	0.1	0.5	0.1
Fand. Val. de Huelva	0.3	0.7	0.5	0.2	0.3	0.1

Table 4.2 Fraction of performances enclosed by the tube per phrase for different styles and $\varepsilon = 3.0$.

4.3.5 Open problems

There are several immediate suggestions for further research.

- The first issue is related to the complexity. Can the asymptotic complexity of the problem be improved? For large data sets, the time complexity of our approach is roughly cubic and we ask if a more detailed study allows us to improve the algorithm, as we did for the case $n = 3$ and $p = n - 1 = 2$. Since p could be close to n, it is even interesting to find an $O^*(n(n-p)m)$-time algorithm.

- In the STP, the width of the tube is the same at any point in time. However, we observed in the transcriptions that the melodic variability is not constant. Consequently, future research could target a STP with variable width. Observe, that the simple idea of intersecting the optimal tube with the polygon between the upper and lower hulls of the functions gives us a more accurate visualisation of the variability.

- Another possible variant is to restrict the number of vertices of the template. In fact, for highly ornamented melodies, the template appears to be very complex and a simpler prototype could better model the underlying melodic movement.

- A further interesting task is to efficiently solve the reverse problem: Given an $\varepsilon > 0$, compute the maximum number p^* of melodies that can be captured by an ε-tube. Note, that our algorithm solves this problem in $O(n^2 m \log n \log p)$ using binary search. Is it possible to improve the running time using a different approach?

- Finally, the extension of our problem to 3D leads to an interesting task: the recognition of user-defined temporal gestures from tactile interfaces.

Given a set of n curves (gestures) f_i in the XY plane, we want to compute a new continuous curve, which approximates the set of curves f_i. Note that the signatures f_i can be represented by the orthogonal projection of three-dimensional curves in the XYT space that are monotone on time. Consequently, the signature problem can be seen as a generalisation of the Spanning Tube Problem (see Figure 4.14). The definition in 2D can be extended to 3D as follows:

The 3D STP: *Let $a, b, c, d, t \in \mathbb{R}$, with $a < b$, $c < d$ and $t > 0$; let $n, m, p \in \mathbb{Z}^+$; and for $i = 1, \cdots, n$, let $f_i : [0, t] \to \mathbb{R}^2$ be a T-monotone piecewise linear function with at most m links. Given $p \leq n$, find the minimum $\varepsilon^* > 0$ such that there exists a T-monotone continuous function $f^*(t) = (f_1(t), f_2(t))$ fulfilling that, for each $t \in [0, T]$, the ball (for a L_p distance $(L_2, L_1, L_\infty, \cdots)$) of radius ε^* centered at $(t, f^*(t))$ intersects at least p functions.*

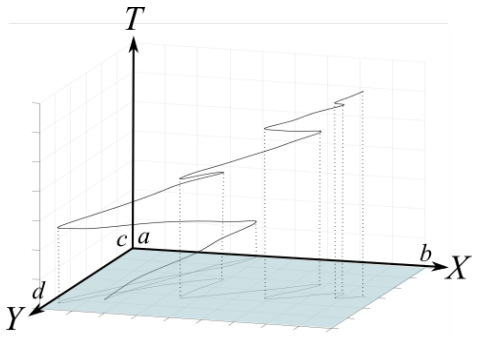


Fig. 4.14 The orthogonal projection of f onto the XY plane is the gesture. f is a T-monotone curve.

4.4 A progressive alignment approach to template extraction.

While the geometric approach introduced in the previous section gave rise to interesting mathematical properties and furthermore led to challenging open problems in the field of computational geometry, it has several limitations with respect to its practical application in MIR systems. In particular, the representation of the template as a continuous polygonal curve has the main drawback that it does not hold information on essential notes or note progressions which appear in most of the analysed performances. In addition, the high temporal resolution of the function can cause alignment methods, as i.e. DTW-based distance measures used in melody classification systems, to become computationally expensive. Furthermore, the model cannot be updated with additional melodies without being recomputing from scratch.

Motivated by these observations, we propose in this section a progressive alignment approach to melodic template extraction, which aims to overcome the limitations of the geometric approach. The method yields a graph structure holding commonalities and differences among the analysed performances, which allows us to extract the melodic template in form of a discrete note sequence.

4.4.1 Problem definition

The algorithm proposed in this section addresses the following problem: *Given a set of (manual or automatic) transcriptions of performances of the same melody, extract a discrete note sequence which approximates the underlying melodic template.* To this end, we present a method to create a dynamic discrete *model* of a set of performance transcriptions, which

1. captures the commonalities and differences among the melodic movements contained in the dataset, in particular with respect to melodic ornamentation and variation,
2. allows us to identify which sections of the melody are stable and where and how ornamentation occurs,
3. allows us to extract a note sequence which approximates the underlying melodic *template*,
4. provides a quantification of *stability* of the template, or, in other words, the agreement among the performances,
5. can be efficiently updated with additional labelled instances,
6. provides the means to jointly evaluate the similarity between a given transcription and a set of performances.

We propose a weighted directed graph structure, holding information on the frequency of occurrence of characteristic notes and note transitions, which is constructed in a *progressive alignment (PA)* manner. *PA* was initially proposed by [56] in the context of molecular biology to reveal the evolutionary connections of protein sequences. First, a *guide tree* defining the alignment order is established based on pair-wise alignment scores computed with the *Needleman-Wunsch* [143] algorithm. After initialising the joint alignment with the first sequence in the order, the model is progressively updated by aligning the remaining sequences. Gaps in the alignment, in form of insertions or deletions of symbols, are reflected in the model by inserting neutral symbols. [56] used the resulting model to construct a phylogenetic tree, displaying similarities among sequences. The approach has since been further developed and is widely used in bioinformatics (see e.g. [199, 122]), but has only recently found application in music analysis. [120] directly applied the approach described in [56] to the comparative analysis of recordings of solo violin performances. Through progressive alignment of beat-synchronous chroma feature vectors, a phylogenetic tree is constructed, which provides insight into global similarities among performances of the same piece. In the context of *music alignment* of *Mazurkas*, [224] compared *PA* and *hidden Markov model profiles* to align a given recording to a group of performances. Both approaches yielded a more accurate alignment compared to pair-wise methods. However, the challenges of approximating the underlying

melodic template and providing a comprehensive representation of local and global melodic stability, have so far not been addressed.

The *PA* algorithm proposed in this section operates on monophonic transcriptions (automatic or manual) of the singing voice melody and yields a graphical model of the notes and note transitions occurring in the set of performances. It allows us to jointly analyse the transcriptions with respect to location, type and amount of occurring melodic variation. Adapting principles from graph theory, we furthermore propose a method to extract a note sequence which approximates the underlying melodic template and derive a metric to quantify its melodic stability.

In a case study on the four *fandangos* variants described in Section 4.2, we explore the potential of the proposed method for the task of melody classification. We show that a comparison with the model, or alternatively the approximated template (both variants are investigated), yields a similar accuracy at a lower run-time cost, compared to pair-wise similarity computations. We furthermore demonstrate the potential of the graphical model for comparative performance analysis by identifying stable and unstable melody sections and by comparing the intra-group stability between the four melodies under study. This type of analysis is a powerful tool for educational purposes as well as for large-scale musicological studies

4.4.2 Methodology

An overview of the proposed framework is depicted in Figure 6.4. Given the note-level representations of N singing performances with a common melodic template, we first normalise all transcriptions to a relative time scale. Subsequently, we establish an alignment order based on pair-wise alignment scores. After initialising a weighted directed graph structure with the information contained in the first transcription, we progressively construct the model by aligning the remaining transcriptions according to the alignment order. The graph is updated after each step. Using a modification of *Dijkstra's algorithm* [47], we estimate the melodic template as the *average heaviest path* through the graph. Comparing the edge weights of this path to the sum of all edges in the model, we derive a measure for *melodic stability* among the performances. By computing the alignment score of an unlabelled performance transcription with the resulting model, or alternatively the extracted template, we can estimate its similarity to the set of transcriptions based on which the model was constructed. Below, all stages are described in detail.

Melody representation

The proposed method operates on the note-level transcriptions described in Section 4.2. In both versions of the transcriptions, AT and MT, each note is described by its onset time, pitch (MIDI pitch quantised to the equal tempered scale) and duration. In order to facilitate

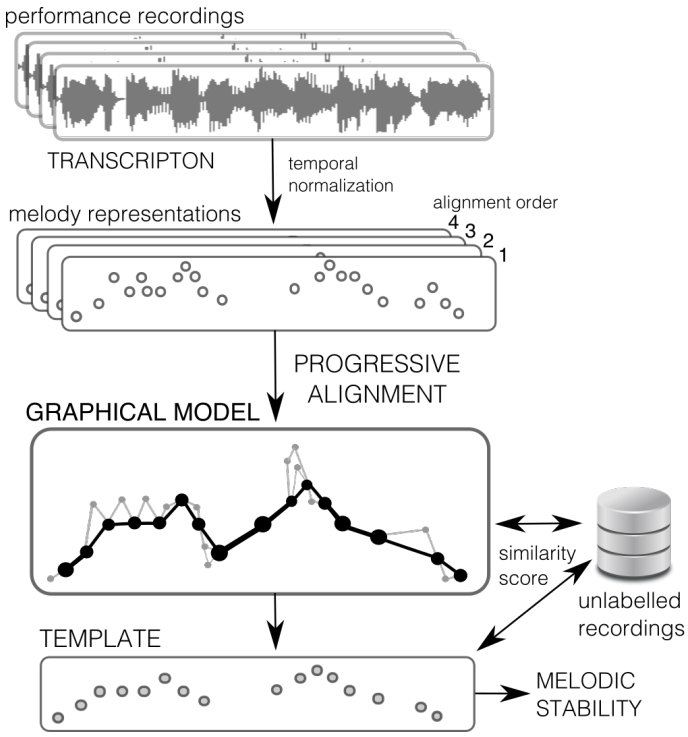


Fig. 4.15 System overview.

pair-wise and later on progressive alignment of note sequences, we first convert each note transcription into a point-set representation $\mathbf{x}^k = (x_1^k, ..., x_{M_k}^k)$, where M_k denotes the number of notes of the k^{th} sequence and each point $x_i^k = (t_i^k, p_i^k)$ is defined by its relative onset time t_i^k and pitch value p_i^k. An example of the resulting melody representation is shown in Figure 4.16. The relative onset time $t_i^k \in [0, 1]$ is computed as the relative position between the onset of the first and the offset of the last sung note, discarding silences between notes.

Key normalisation and pair-wise alignment

In order to align two melodies, we employ again the *Needleman-Wunsch* (NW) [143] algorithm. The method and its application to melodic sequence alignment has been described in detail in Section 3.2. Here, we use a gap penalty of $g = 1$ together with the following scoring function:

$$\gamma(x_i^k, x_j^l) = \begin{cases} 1 - |t_i^k - t_j^l| & \text{if } p_i^k = p_j^l \\ -1 & \text{if } p_i^k \neq p_j^l \end{cases} \tag{4.4}$$

In other words, if both notes are equal in pitch, they are considered a match and a positive score $\gamma(x_i^k, x_j^l) \in [0, 1]$ is assigned based on their onset proximity. Otherwise, the pair is considered a mismatch and a negative score of -1 is assigned, which is equal to the gap penalty.

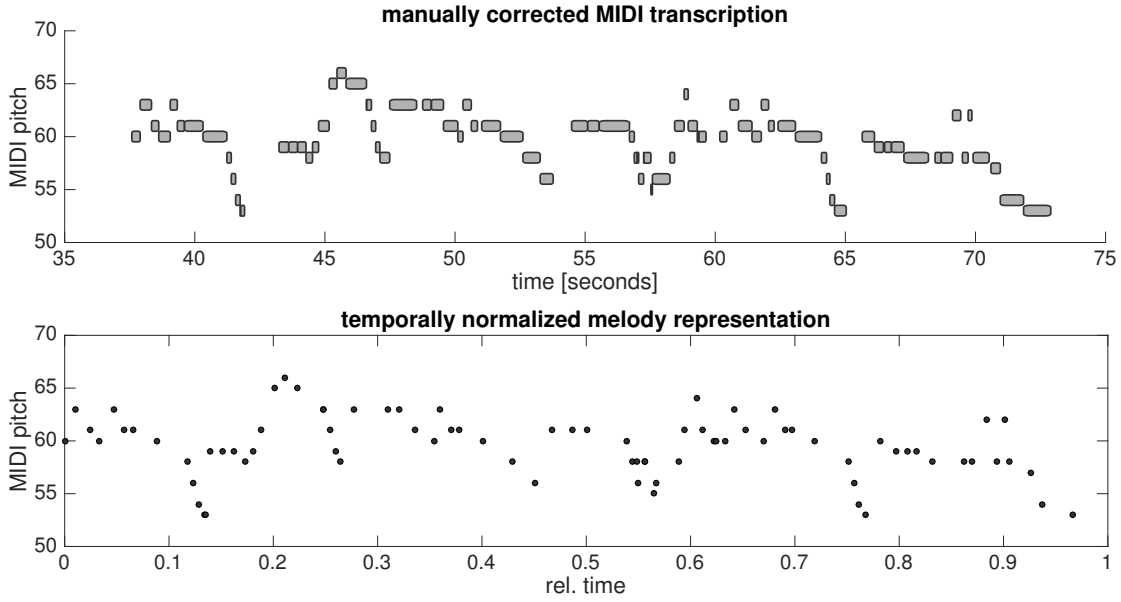


Fig. 4.16 Top: Manually corrected note transcription of a *Fandango de Valverde* sung by *El Raya*; bottom: corresponding temporally normalised point set representation.

The alignment of two sequences, $\mathbf{x}^k$ and $\mathbf{x}^l$, yields an alignment score $d_{align}(\mathbf{x}^k, \mathbf{x}^l)$ and an alignment path $\mathcal{P}_{align}(\mathbf{x}^k, \mathbf{x}^l)$. Since versions of the same melody may be performed in different keys, we furthermore employ the key normalisation procedure described in Section 2.2 prior to each pair-wise alignment.

Establishment of the alignment order

In the progressive alignment procedure to be described, the alignment order is of major importance. While the first sequences to be aligned strongly shape the model, later aligned sequences have a weaker influence. We therefore aim to align *typical* or *common* performances before aligning outliers. To this end, we establish an alignment order $\mathcal{A}$ based on pair-wise alignment scores.

We construct the score matrix $\mathcal{S}$ holding pair-wise alignment scores between all melody representations, where

$$\mathcal{S}(k,l) = d_{align}(\mathbf{x}^k, \mathbf{x}^l). \tag{4.5}$$

Since the alignment scores are length-dependent, we normalise each pair-wise score by dividing it by the average length of the two involved sequences.

$$\mathcal{S}(k,l) := \frac{\mathcal{S}(k,l)}{0.5 * (M_k + M_l)} \tag{4.6}$$

We now establish the alignment order $\mathcal{A}$, by sorting the sequences by their average similarity $\bar{\mathcal{S}}_k$

$$\bar{\mathcal{S}}_k = \frac{1}{N} \sum_{\substack{n=1 \\ n \neq i}}^{N} \mathcal{S}(k, n) \tag{4.7}$$

to all other sequences in descending order

$$\mathcal{A} = \{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^N \mid \bar{\mathcal{S}}_k \geq \bar{\mathcal{S}}_{k+1}; k = 1, ..., N-1\}. \tag{4.8}$$

In this way, melodies with a high average similarity to all other melodies appear early in the alignment order, and melodies which are dissimilar to other melodies are aligned last.

The complexity of the establishment of an alignment order of N melodies containing at most M notes each results to $O(N^2M^2) + O(N \log(N))$. The first term originates from the pair-wise alignments whereas the second corresponds to cost of the sorting procedure.

Model initialization

As a next step, we initialise a weighted directed graph G with vertices V and edges E based on $\mathbf{x}^1$, the first transcription according to the alignment order. For each note x_i in this transcription, we create a vertex $v_i \in V$ and, for each note transition from i to $i+1$, we add an edge $e_{i,i+1} \in E$ connecting v_i to v_{i+1} and set its weight $w_{i,i+1} \in W$ to 1. Furthermore, we assign a score $q_i \in Q$ to each vertex v_i, which we again initialise with $q_i^G = 1$ and maintain the pitch values $p_i^G \in P$ and onset times $t_i^G \in T$ associated with the note from which the vertex was created. We add two additional vertices, $v_{\mathbb{S}}$ and $v_{\mathbb{E}}$, which act as fictitious start and end points of the melodic sequence. The edge $e_{\mathbb{S},1}$ connects $v_{\mathbb{S}}$ to the vertex representing the first note and $e_{M_1,\mathbb{E}}$ connects the vertex representing the last note to $v_{\mathbb{E}}$. The weights of both edges are initialised with $w_{M_1,\mathbb{E}} = w_{\mathbb{S},1} = 1$. We denote the number of vertices contained in G with $|V|$. An example of the resulting structure $G(V, E, W, Q, P, T)$ is shown in Figure 4.17. Note, that W is an edge property and each $w_{i,i+1} \in W$ is associated with the edge $e_{i,i+1} \in E$. Similarly, Q, P and T are vertex properties, where elements $q_i \in Q$, $p_i \in P$ and $t_i \in T$ are associate with the vertex $v_i \in V$. An algorithmic description of the model initialisation step is provided in Algorithm 4.1.

Progressive alignment and model update

Once the model is initialised, we successively align the remaining sequences according to the order $\mathcal{A}$ and update the model after each step. First, let $\mathbf{x}^G$ be the sequence of notes corresponding to the nodes in graph G, where the order of notes is determined by their associated onset times. We align a new sequence $\mathbf{x}^k$ to $\mathbf{x}^G$ using the method in described in Section 4.4.2 with two minor modifications. First, for $\mathbf{x}^G$, we replace the duration weights of the pitch histogram required during the key normalisation process with the node scores. This

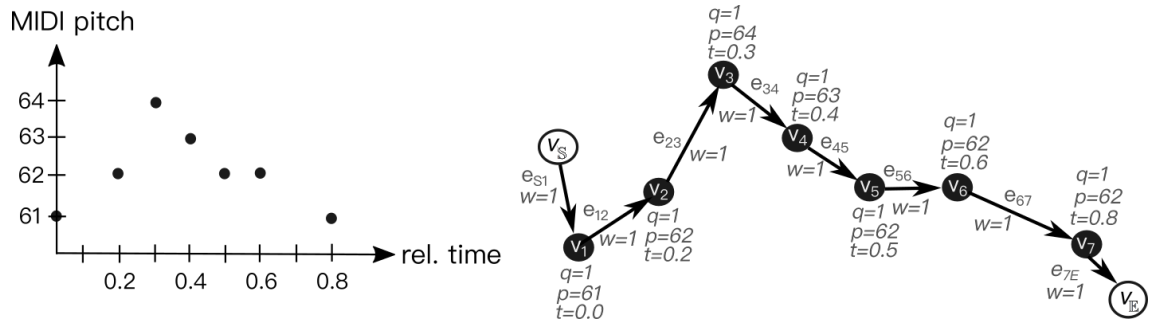


Fig. 4.17 Initialisation of the graph model: Melody representation (left) and derived initial graph structure (right).

Algorithm 4.1 Initialisation of the graph model $G(V, E, W, Q, P, T)$ based on $\mathbf{x^1}$.

$G(V, E, W, Q, P, T) \leftarrow$ empty graph structure
$M_1 \leftarrow$ length of $\mathbf{x^1}$
for $i = 1$ to M_1 **do** # add vertices and their properties
  add v_i to V
  add $q_i = 1$ to Q
  add $p_i^G = p_i^1$ to P
  add $t_i^G = t_i^1$ to T
for $i = 1$ to $M_1 - 1$ **do** # add edges and their weights
  add $e_{i,i+1}$ to E
  add $w_{i,i+1} = 1$ to W
return $G(V, E, W, Q, P, T)$

is necessary, because after several updates, $\mathbf{x}^G$ contains notes which originate from different recordings. Therefore, the absolute durations are not directly comparable. Using node scores is a suitable choice, since more importance will be given to characteristic notes which have previously formed part of alignment paths.

Secondly, in a similar manner, we modify the scoring function γ of the NW algorithm to

$$\gamma(x_i^k, x_j^G) = \begin{cases} q_j^G(1 - |t_i^k - t_j^G|) & \text{if} \quad p_i^k = p_j^G \\ -1 & \text{if} \quad p_i^k \neq p_j^G \end{cases} \tag{4.9}$$

where q_j^G is the score and t_j^G the onset associated with the j^{th} vertex. In this way, an alignment with vertices which have already accumulated a high score will be favoured. After tracing the optimal alignment path, we can extract the sequences of matches. We denote $\mathcal{P}_{\text{match}} = \{\mathcal{P}_k, \mathcal{P}_G\}$ as the sequence holding the note indices of $\mathbf{x}^k$ in $\mathcal{P}_k$ which have been matched along the optimal path with corresponding elements of $\mathbf{x}^G$, held in $\mathcal{P}_G$. Note, that a pair in the optimal alignment path is only considered a match, if the pitch values are equal.

Gaps or local similarities with a negative score are not considered. Consequently, $\mathcal{P}_{match}$ is a subsequence of $\mathcal{P}_{align}$, where element pairs containing gaps are omitted.

In order to update the model, we first increase the score of all vertices contained in $\mathcal{P}_G$ by 1. In this way, the score of a vertex reflects how many times the corresponding note has formed part of a matching path. Then, for each note $x_u^k \notin \mathcal{P}_k$ in $\mathbf{x}^k$ which is not contained in the optimal path, we add a new vertex v_{new} to the model with pitch p_u^k and score 1. We compute the onset time of the new vertex t_{new}^G by linear interpolation (Figure 4.18): Let t_{prev}^k with $prev \in \mathcal{P}_k$ be the onset of the last note preceding x_u^k which has been matched to a vertex in the model, and let similarly t_{post}^k with $post \in \mathcal{P}_k$ be the first note following x_u^k which has been matched to the model. Furthermore, let t_{post}^G and t_{prev}^G be the onset times associated with vertices matched with t_{prev}^k and t_{post}^k, respectively. The onset time of the newly inserted vertex is then interpolated as

$$t_{new}^G = t_{prev}^G + \frac{t_u^k - t_{prev}^k}{t_{post}^k - t_{prev}^k} \cdot (t_{post}^G - t_{prev}^G). \tag{4.10}$$

If x_u lies before the first matched note in $\mathcal{P}_k$, then

$$t_{new}^G = t_{post}^G - |t_u^k - t_{post}^k| \tag{4.11}$$

and, analogously, if x_u is located after the last matched note in $\mathcal{P}_k$, t_{new} results to

$$t_{new}^G = t_{prev}^G + |t_u^k - t_{prev}^k| \tag{4.12}$$

As a last step, we update the edge weights: For each note transition $(i, i+1), i \in [1, M_k - 1]$ in the sequence $\mathbf{x}^k$, we increase the weight of the edges connecting the corresponding vertices (which have either been matched to a note or which have been newly created) by 1. This applies also to the edge connecting $v_{\mathbb{S}}$ to the first note as well as the edge connecting the last note in $\mathbf{x}^k$ to $v_{\mathbb{E}}$.

Given that the computational complexity of each pair-wise alignment of two sequences of length M is $O(M^2)$ and at its last stage, the model has at most $O(NM)$ vertices, the maximum cost for updating the model results to $O(NM^2)$.

The process of updating the graph model is depicted in Figure 4.19 and pseudo-code is provided in Algorithm 4.2. This step is repeated until all sequences in $\mathcal{A}$ have been processed. The model allows dynamic updating and can be further extended with additional labelled sequences.

In the sequel, we described how we use the resulting model to (a) extract a note sequence which approximates the template, (b) derive a measure for melodic stability, and (c) compute the joint melodic similarity between an unlabelled sequence and the union of melodies contained in the model.

Algorithm 4.2 Updating the graph model $G(V, E, W, Q, P, T)$ with $\mathbf{x}^k$.

$\mathbf{x}^G \leftarrow$ CONVERT_TO_NOTE_SEQUENCE(G)
$\mathcal{P}_{align} \leftarrow$ COMPUTE_ALIGNMENT_PATH($\mathbf{x}^k, \mathbf{x}^G$)
$\mathcal{P}_{match}(\mathcal{P}_k, \mathcal{P}_G) \leftarrow$ empty sequence
for $i = 1$ **do** to length($\mathcal{P}_{align}$)
  if gap $\notin \mathcal{P}_{align}(i)$ **then**
    append $\mathcal{P}_{align}(i)$ to $\mathcal{P}_{match}$
for $i \in \mathcal{P}_G$ **do**# increase score of matched notes
  $q_i = q_i + 1$
for $i = 1$ to M_k **do**# add new vertices for unmatched notes
  if $i \notin \mathcal{P}_k$ **then**
    add v_{new} to V
    add $q_{new} = 1$ to Q
    add $p^G_{new} = p^k_i$ to P
    add $t_{new} =$ INTERPOLATE_ONSET($t_{new}, \mathcal{P}_{align}$)
    $\mathcal{P}_{align}(i, 2) =$ index(v_{new}) # replace gaps with index of new vertex
for $i = 1$ to $M_k - 1$ **do** # add edges and update weights
  $sNode = \mathcal{P}_{align}(i, 2)$
  $eNode = \mathcal{P}_{align}(i + 1, 2)$
  if $e_{sNode,eNode} \in E$ **then** # edge already exists
    $w_{sNode,eNode} = w_{sNode,eNode} + 1$
  else
    add $e_{sNode,eNode}$ to E # create new edge
    add $w_{sNode,eNode} = 1$ to W
if $e_{\mathbb{S},\mathcal{P}_{align}(1,2)} \in E$ **then** # create or update edge from start node
  $w_{\mathbb{S},\mathcal{P}_{align}(1,2)} = w_{\mathbb{S},\mathcal{P}_{align}(1,2)} + 1$
else
  add $e_{\mathbb{S},\mathcal{P}_{align}(1,2)}$ to E # create new edge
  add $w_{\mathbb{S},\mathcal{P}_{align}(1,2)} = 1$ to W
if $e_{\mathcal{P}_{align}(end,2),\mathbb{E}} \in E$ **then** # create or update edge to end node
  $w_{\mathcal{P}_{align}(end,2),\mathbb{E}} = w_{\mathcal{P}_{align}(end,2),\mathbb{E}} + 1$
else
  add $e_{\mathcal{P}_{align}(end,2),\mathbb{E}}$ to E # create new edge
  add $w_{\mathcal{P}_{align}(end,2),\mathbb{E}} = 1$ to W
return $G(V, E, W, Q, P, T)$

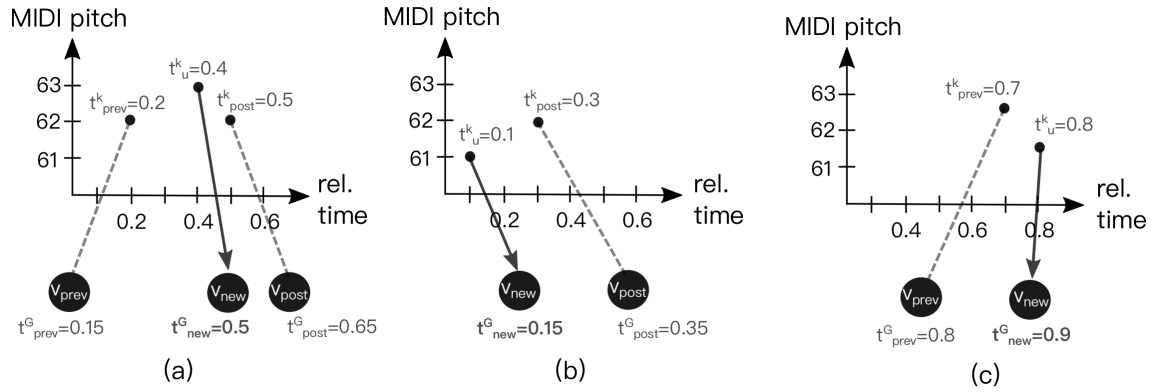


Fig. 4.18 Interpolation of onset time t^G_{new}: (a) t^k_u is located between two matched note onsets; (b) t^k_u is located before the first matched onset; (c) t^k_u is located after the last matched onset.

Template extraction

From the graph model G, we now aim to extract a note sequence $\mathbf{x}^T$ which approximates the underlying melodic template on which all N performances are based. Ideally, the template will reflect the melodic essence which the performances have in common and should therefore contain note successions which can be found in most transcriptions. Ornaments and segments which vary among performances, on the other hand, should be discarded. To this end, we compute the *average heaviest path* $\mathbf{B}$ through G, connecting the start node $v_{\mathbb{S}}$ to the end node $v_{\mathbb{E}}$, that is, the path from $v_{\mathbb{S}}$ to $v_{\mathbb{E}}$ with the highest average edge weight among all possible paths. In this way, we capture frequent notes and note transitions, while omitting ornamentation. We formulate the task of finding the *average heaviest path* as a variant of the well studied *shortest path problem* [1] and solve it with a modification of *Dijkstra's algorithm* [47], as shown in Algorithm 4.3. Note that this approach is valid in the given scenario, since G is a directed acyclic graph (DAG) and the solution can therefore be obtained by progressively extending the solutions of subproblems.

By construction, we have a linear ordering of the graph and we one by one process all vertices in order from left to right. Starting at $v_{\mathbb{S}}$, we successively parse all vertices until reaching $v_{\mathbb{E}}$ and maintain the average heaviest path leading to each vertex. When visiting vertex v_i, we examine each of its unvisited neighbours. We compute the average weight of the concatenation of the best path leading to v_i and the edge connecting v_i to its neighbour $v_{n,i}$. If this value exceeds the average weight of the best path stored for reaching $v_{n,i}$, we replace the best path accordingly. After parsing all neighbours, v_i is marked as visited and we proceed to parsing its strongest connected neighbour. This process is repeated until $v_{\mathbb{E}}$ has been marked visited and the best path stored for $v_{\mathbb{E}}$ is returned as the average heaviest path $\mathbf{B}$ from $v_{\mathbb{S}}$ to $v_{\mathbb{E}}$. An example is shown in Figure 4.20. Finally, we define the template

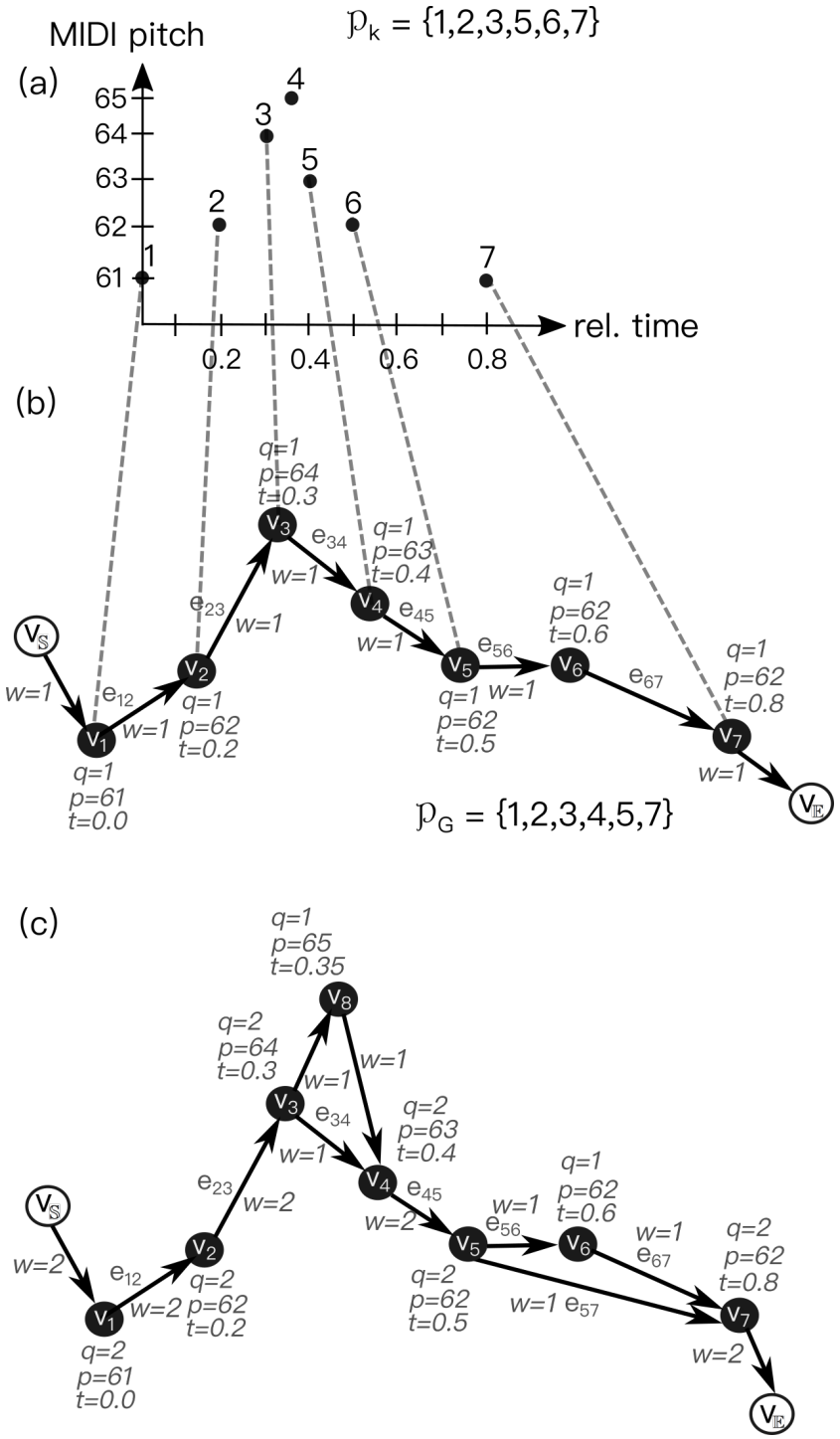


Fig. 4.19 Model update: (a) Melody representation, alignment to the model and match sequence $\mathcal{P}_k$; (b) model before update and match sequence $\mathcal{P}_m$; (c) model after update.

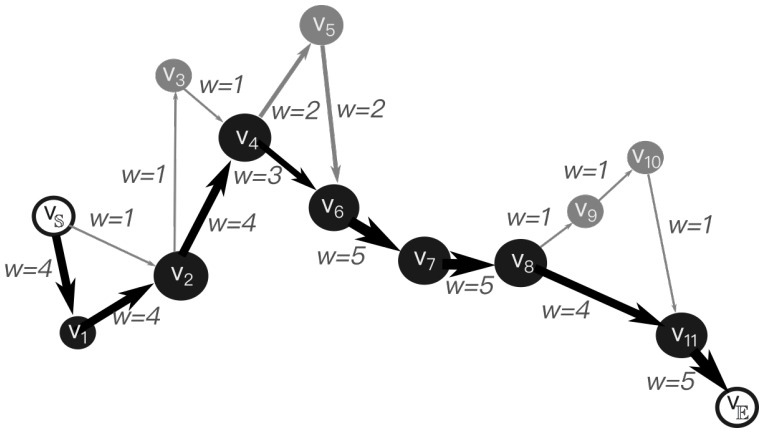


Fig. 4.20 Average heaviest path (black) through a graph G. Circle sizes and line widths are proportional to vertex scores and edge weights, respectively. Vertices and edges constituting the template are marked black, others grey.

$\mathbf{x}^T$ as the sequence of notes associated with the vertices through which $\mathbf{B}$ passes, excluding $v_{\mathbb{S}}$ and $v_{\mathbb{E}}$.

Algorithm 4.3 Algorithm for finding the *average heaviest path* from v_S to v_E in graph $G(V, E)$.

for all $v \in V$ **do**
  $bestPath[v] \leftarrow$ empty array
$\mathcal{U} \leftarrow$ V
$curr \leftarrow v_S$
while $v_E \in \mathcal{U}$ **do**
  for all $v_n \in neighbours[curr] \wedge \mathcal{U}$ **do**
    $avWeight \leftarrow (\mathbf{sum}(w(e) \in bestPath[v_n]) + w(e_{curr,v_n}))$
    $avWeight \leftarrow avWeight/(\mathbf{length}(bestPath[v_n]) + 1)$
    if $avWeight > \mathbf{sum}(w(e) \in bestPath[curr])/\mathbf{length}(bestPath[curr])$ **then**
      $bestPath[v_n] \leftarrow$ append $e(curr, v_n)$ to $bestPath[curr]$
  remove curr from $\mathcal{U}$
  curr $\leftarrow neighbours[curr]$ with largest weight connecting to curr
return $bestPath[v_E]$

The asymptotic complexity of extracting the average heaviest path on acyclic graphs is $O(|V| + |E|)$. In our case, this would yield a complexity of $O(NM)$, since each melody adds at most M vertices and $2M$ edges to the model. However, in practice we expect a large amount of notes to be matched with existing model nodes, resulting in significantly fewer nodes and edges to be added at each step. Assuming the number of newly added nodes and vertices to be constant results in an expected linear runtime in practice.

Quantifying melodic stability

Based on the graph model G and the average heaviest path $\mathbf{B}$, we now define a measure for melodic stability among the analysed performances. To this end, we analyse the weights of the edges forming $\mathbf{B}$ with respect to the sum of all edge weights in G. If the performances are very similar to each other, $\mathbf{B}$ will concentrate a large percentage of the total amount of edge weights contained in G. The more the performed melodies vary, the weaker the edges of $\mathbf{B}$ will be. Consequently, we compute the melodic stability r as the fraction of the sum of all edge weights contained in $\mathbf{B}$ as

$$r = \frac{\sum_{e \in \mathbf{B}} w(e)}{\sum_{e \in E} w(e)}. \tag{4.13}$$

Evaluating a transcription against a model

In this work, we explore two strategies to estimate the similarity between a query melody $\mathbf{x}^q$ and a set of K performance transcriptions. The first option is to align $\mathbf{x}^q$ to the sequence representing the graph model $\mathbf{x}^G$, using the NW algorithm with the modified scoring function from Equation 4.9, and estimating the similarity as the resulting score $\gamma(\mathbf{x}^G, \mathbf{x}^q)$ of the optimal alignment path. In order to perform an alignment to the model, the graph is represented as a sequence, where the order of nodes is determined by their associated onset times t_i^G. Given that different models may be constructed from datasets with a varying number of recordings and intra-group similarities, they may exhibit significant differences with respect to the sum of accumulated score values. We therefore normalise the alignment score by dividing by the sum of scores of all nodes contained in the model.

Alternatively, we compute $\gamma(\mathbf{x}^T, \mathbf{x}^q)$, the score of aligning the unlabelled melody to the template, using the scoring function from Equation 4.4. Given that the computational cost of the NW algorithm is $O(M_1 M_2)$, M_1 and M_2 being the lengths of the sequences to be aligned, it should be mentioned that aligning with the template is, in most cases, the more efficient strategy. The amount of notes contained in the template is at most the amount of notes contained in the graph, but it is expected to be lower in practice. The number of notes in the template will only equal the number of notes in the model, if all nodes form part of the average heaviest path. This scenario corresponds to the (impossible) case when all transcriptions are identical.

4.4.3 Application to melody classification

The concept of supervised melody classification has been introduced in Chapter 2. In current state of the art systems, an unknown melody is classified according to its similarity to a large number of annotated melodies. Here, we explore an alternative strategy based on the proposed method. The key idea is to compare the unlabelled melody to a single sequence per class, which is representative of all annotated melodies belonging to the respective category.

To this end, we explore both approaches described in the previous section to estimate the similarity between a given melody and a set of performance transcriptions and compare to several baseline scenarios.

Experimental setup

In order to investigate the suitability of the proposed framework for melody classification tasks, we compare five different strategies to assign one of c candidate labels (in our case $c = 4$) to an unlabelled melody based on its similarity to annotated instances, as shown in Figures 4.21 and 4.22: The three setups in Figure 4.21 are considered baseline methods, whereas the two scenarios shown in Figure 4.22 rely on the framework presented in this study.

The first baseline method described in Figure 4.21 (a), subsequently denoted as *k-NN*, corresponds to the commonly used method for melody classification, which has shown to give reliable results in the context of *flamenco* sub-style classification [43, 135] as well as tune family recognition [211, 210]. The unlabelled melody $\mathbf{x}^q$ is classified in a *k-nearest-neighbour* [155] scheme based on pair-wise similarities with all labelled instances contained in the database. While different similarity functions have been introduced in literature, see for example [215] and [216] (note that most of these methods are not applicable to automatic transcriptions without time quantisation), we use here the score of the *NW* alignment in order to provide a direct comparison to the other strategies. The major drawback of this method, and a key motivation for the proposed framework, is the high amount of computationally expensive alignments at run-time. The second baseline method displayed in Figure 4.21 (b) circumvents this issue by aligning $\mathbf{x}^q$ to a single, randomly selected labelled sequence of each class. Since randomly selected instances may represent either "typical" performances or outliers, we report average classification accuracies for 100 repetitions of the experiment. This setup is subsequently referred to as *random*. Finally, we compare to the exhaustive case, where a label is assigned based on the average alignment score with all instances of each class (Figure 4.21 (c)). This method is referred to as *average*. It is worth mentioning, that while the *k-NN* and *average* approaches are computationally expensive, the similarity calculations could potentially be parallelised.

The fourth and fifth setup, depicted in Figure 4.22, are based on the proposed template extraction framework: In (a) (denoted as *model*), $\mathbf{x}^q$ is classified based on the alignment scores $\gamma(\mathbf{x}^{G_i}, \mathbf{x}^q), 1 < i < c$ with the models G_i extracted for the different classes. While the model extraction requires exhaustive pair-wise comparison, this procedure can be computed offline. At run-time, only c alignments are required. Similarly, in the procedure shown in Figure 4.22 (b), the class is assigned based on the alignment scores $\gamma(\mathbf{x}^{T_i}, \mathbf{x}^q), 1 < i < c$ of the unlabelled instance with the templates $\mathbf{x}^{T_i}$ extracted for the different classes.

For each scenario, unlabelled melodies are first transformed to the time normalised representation and shifted to the key of the reference sequence (see Section 4.4.2). Setups

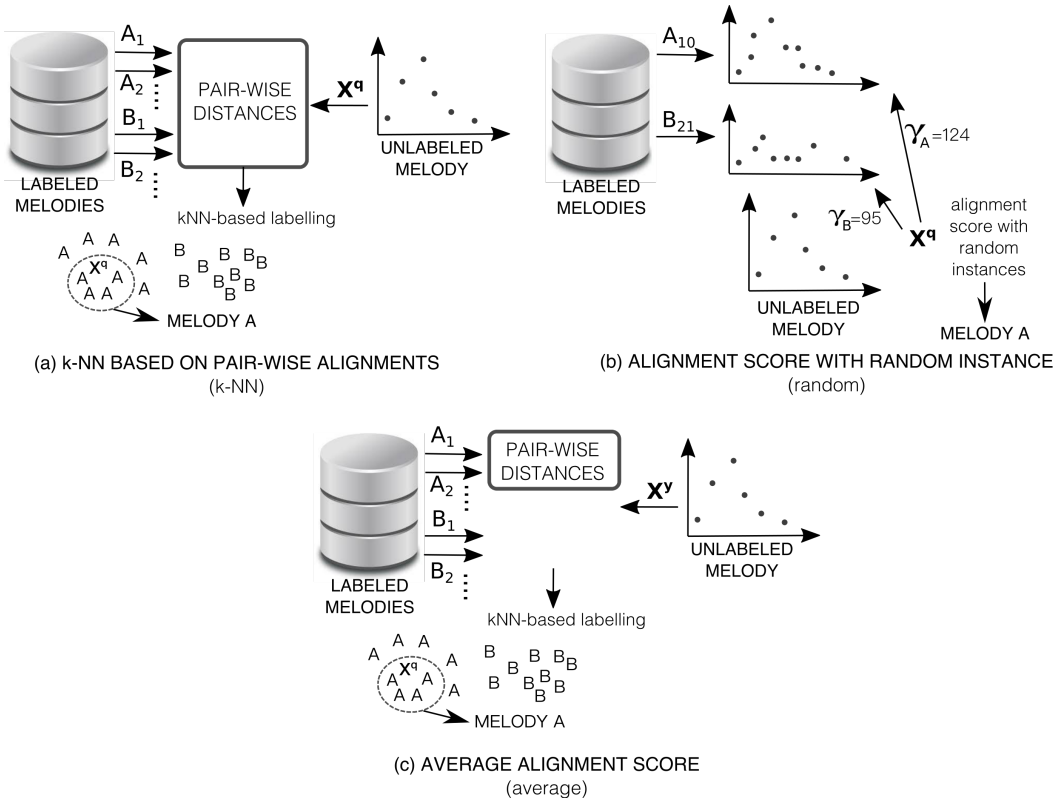


Fig. 4.21 Experimental setup of the baseline methods: (a) k-NN classification based on pair-wise alignment with all labelled instances (*k-NN*); (b) classification based on a single alignment score of a randomly selected instance of each class (*random*); (c) classification based on the average alignment scores with all instances of each class (*average*).

(c) and (d) are evaluated in a *leave-one-out* validation, meaning that the melody to be classified did not participate in the construction of the model. Furthermore, all experiments are conducted for both, *AT* and *MT*. For each setup, we report the percentage of correctly classified instances *CCI*. For the *random* setup, we furthermore provide the variance scores among repetitions. Given that the 40 instances are equally distributed among the 4 groups, the naive baseline accuracy for this scenario results to 25%.

Results

The results of the melody classification experiments are shown in Table 4.3. For the case of MT, it can be seen that both, the model-based as well as the template-based classification, yield the same classification accuracy (95%) as the computationally more expensive *k-NN* and *average* methods. When comparing to a single randomly selected instance, the accuracy drops to 87.8% on average. For the case of AT, we observe a similar behaviour. Both, *template* and

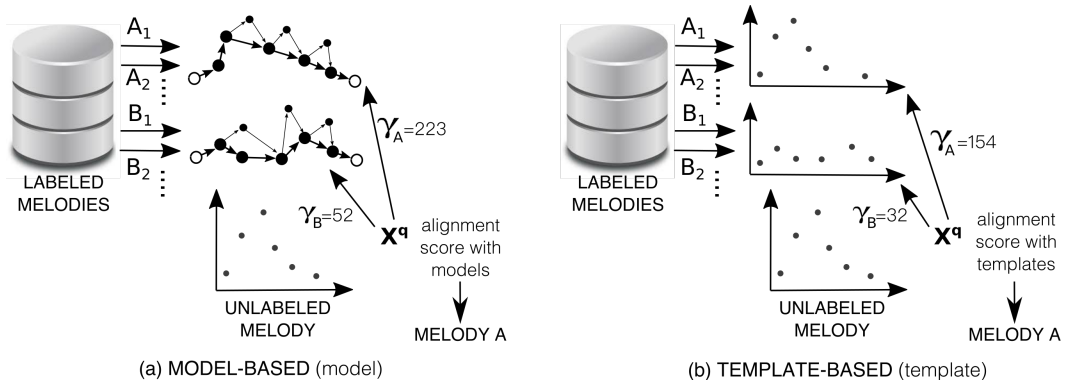


Fig. 4.22 Experimental setup of the proposed method: (a) classification based on alignment scores with class models (*model*); (b) classification based on alignment with extracted class templates (*template*).

	accuracy CCI [%]	
setup	**MT**	**AT**
k-NN	95.0	90.0
random	87.8 (var = 0.20)	76.8 (var = 0.33)
average	95.0	90.0
model	95.0	90.0
template	95.0	90.0

Table 4.3 Classification accuracy for the five different experimental setups

model obtain the same accuracy of 90% as *k-NN* and *average*. The random selection results in a lower accuracy of only 76.8%. In general, these results indicate, that both, the template- as well as the model-based approach, appear to yield a competitive performance compared to the computationally more expensive *k-NN* and *average* setups.

Error analysis

After having assessed the numerical results, we now proceed with a manual inspection of the incorrectly classified items. A first important observation is the fact that the instances misclassified by the *k-NN*, *average*, *model* and *template* setups are identical. For MT, two performances belonging to the *fandangos valientes de Huelva* style were mistakenly classified as *fandangos valientes de Alosno*. Analysing these two examples revealed that they are indeed rather atypical performances which deviate significantly from the underlying template. Figure 4.23 shows the estimated template for the *fandangos valientes de Huelva* style together with one example of a correctly classified instance and one example which was misclassified. It can be seen, that the misclassified example differs strongly from the template as well as

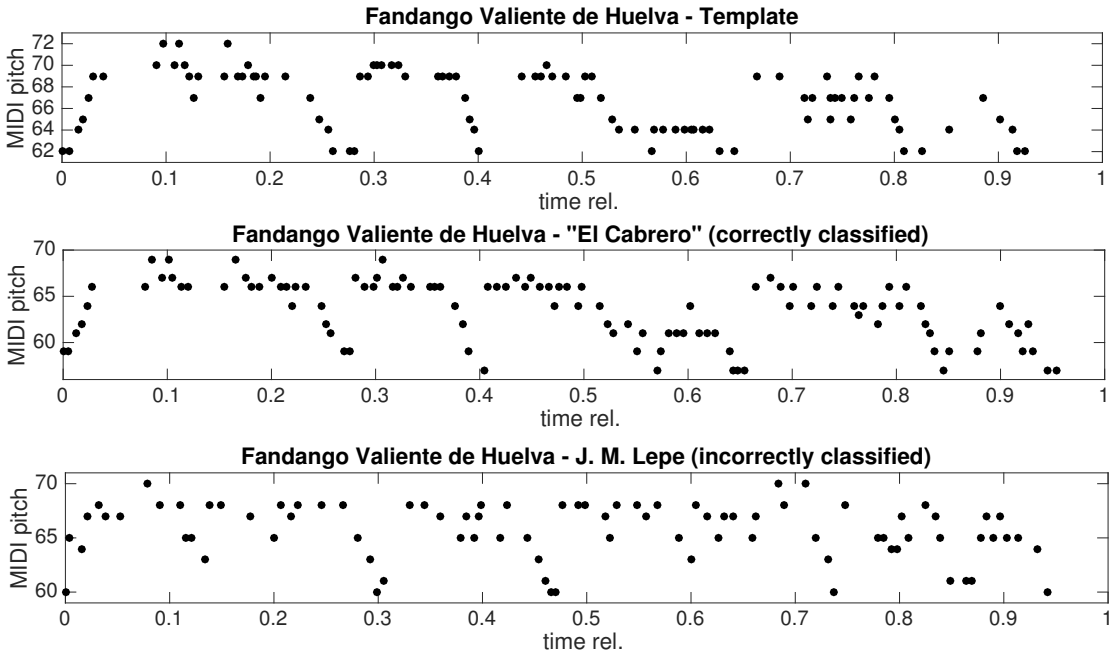


Fig. 4.23 *Fandangos Valientes de Huelva*: Template and examples of a correctly and an incorrectly classified melody.

from the correctly classified instance. For AT, two additional instances were misclassified in all four setups. For one of them, the manual and automatic transcription is shown in Figure 4.24. It can be seen that the overall contour of the AT is distorted by octave errors, in particular around $t = 0.35$, $t = 0.65$ and $t = 0.95$. It is very likely that these transcription errors caused a lower alignment score with the corresponding template, which resulted in misclassification of this melody.

Melody classification in a noisy corpus

In a last experiment, we explore the potential of the extracted templates in a melody retrieval task. In particular, we aim to identify further instances of the four melodies in a noisy corpus by computing alignment scores to the templates. To this end, we gathered five additional examples of each of the four fandango styles, which were not used in any of the prior experiments. These recordings are taken from video sharing platforms as well as private collections and the quality ranges from amateur videos of informal flamenco gatherings to professional studio productions. Recordings may contain various exhibitions of the melody as well as additional melodies belonging to other styles. To this set, we add 1169 recordings from the *corpusCOFLA* [102], a research corpus containing commercial *flamenco* recordings. These recordings belong to other style families and are not related to the four *fandango*

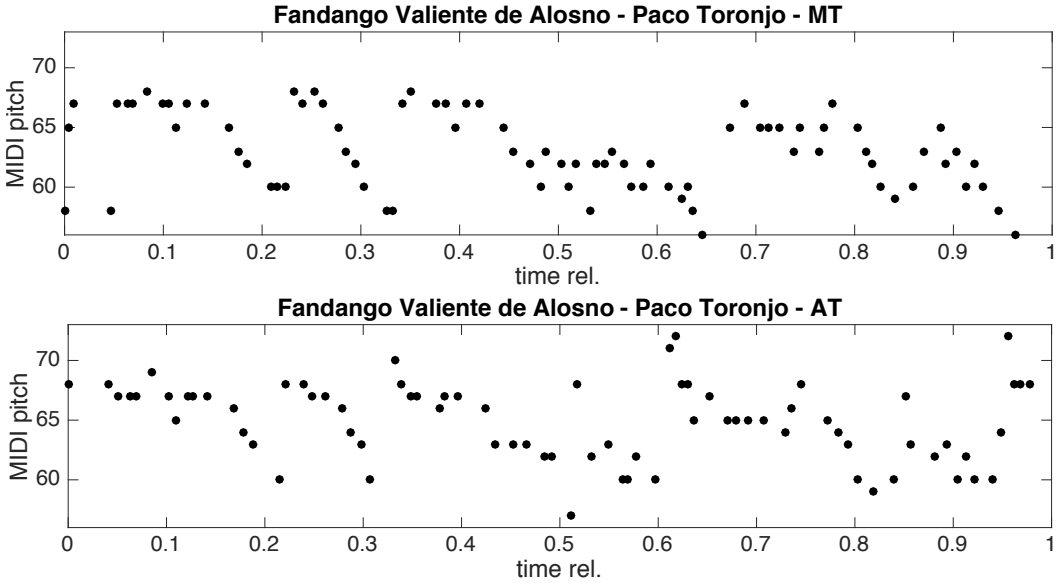


Fig. 4.24 *Fandangos Valientes de Alosno*: MT and AT of one of the incorrectly classified melodies.

melodies under study. For all recordings, we extract automatic transcriptions of the singing voice melody using the *CANTE* [104] algorithm.

Assuming that various melody exhibitions will alternate with guitar interludes, we split each recording at non-vocal sections lasting at least 3 seconds. Then, we align the resulting segments to the four templates extracted from the database used in the previous experiment. For each template, we store the highest obtained alignment score among the segments. Then, for each of the four cases, we analyse how many of highest ranked recordings are relevant to the respective template melody, meaning, they contain at least one exhibition of the template melody. To this end, we compute the retrieval recall rec_R within the R ranked recordings.

$$\text{rec}_R = \frac{\#\text{ relevant items within the top } R \text{ ranks}}{\#\text{ relevant items in the search space}} \tag{4.14}$$

Figure 4.25 shows the retrieval recall for the four melodies when considering the 5 (rec_5), 10 (rec_{10}) and 20 (rec_{20}) highest ranked recordings. We furthermore distinguish whether MT or AT were used to estimate the templates. It can be seen, that for *Fandangos de Valverde* and *Fandangos Valientes de Huelva*, the five highest ranked recordings correspond to the five recordings containing the relevant melodies. For the worst case, the *Fandangos de Calaña*, three relevant results are located within the first five ranks and the remaining two relevant melodies are not ranked among the top 20 results. For the *Fandangos Valientes de Alosno*, three relevant results are located within the first five ranks, and the remaining two are located between ranks 5 and 10. These results are promising, given that the search space

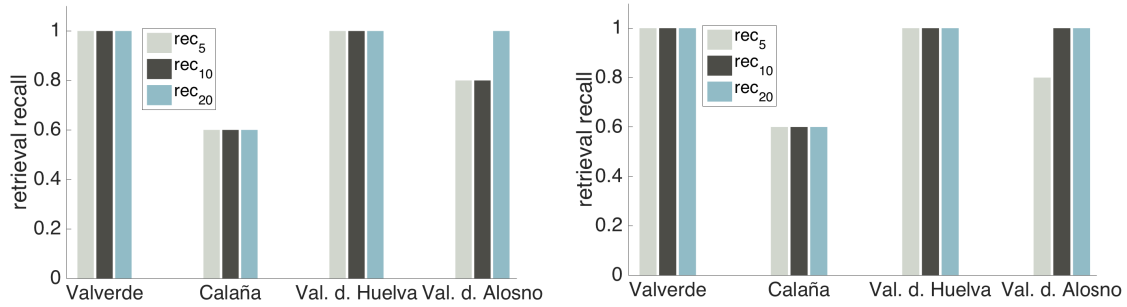


Fig. 4.25 Retrieval recall for manual and automatic transcriptions.

encompasses a total of 1189 recordings. We manually inspected the two examples which were not located within the top 20 results and found that severe vocal detection errors are most likely the cause for the low ranking. For example, in one recording, various seconds of the relevant singing voice melody are missing. In the other recording, guitar segments between the melody exhibitions were mistakenly transcribed as singing voice. In this case, the melody exhibitions are not segmented correctly and the alignment with the template yields a low score. We do not observe any differences between using the templates extracted from AT and MT.

4.4.4 Application to comparative performance analysis

In this section, we showcase the use of the proposed method in the context of a comparative performance analysis. First, we show how visualisations of the graphical models and an analysis of the melodic stability values provide a comprehensive overview of the amount and location of occurring melodic variation. Comparing the extracted templates, we furthermore discover inter-style similarities in the melodic contour. We then show how single performances of the same melody can be compared by visualising their alignment with the approximated template.

Figures 4.26 and 4.27 show the graph models extracted for the four *fandango* styles under study. Comparing among styles, we can experimentally confirm the observations described earlier: The two folkloric styles, the *fandangos de Valverde* and the *fandangos de Calaña*, exhibit a stable melodic skeleton, which results in high vertex scores and heavy edges along the notes belonging to the template. This indicates that singers largely follow the template during performance and tend to introduce only minor melodic variation. The templates of the two *valiente* styles appear to be less defined, which indicates that singers tend to introduce more variation. These observations are furthermore reflected in the melodic stability values shown in Table 4.4, where the highest values are observed for he *fandangos de Valverde* and the *fandangos de Calaña* ($r = 0.62$ and $r = 0.64$, respectively).

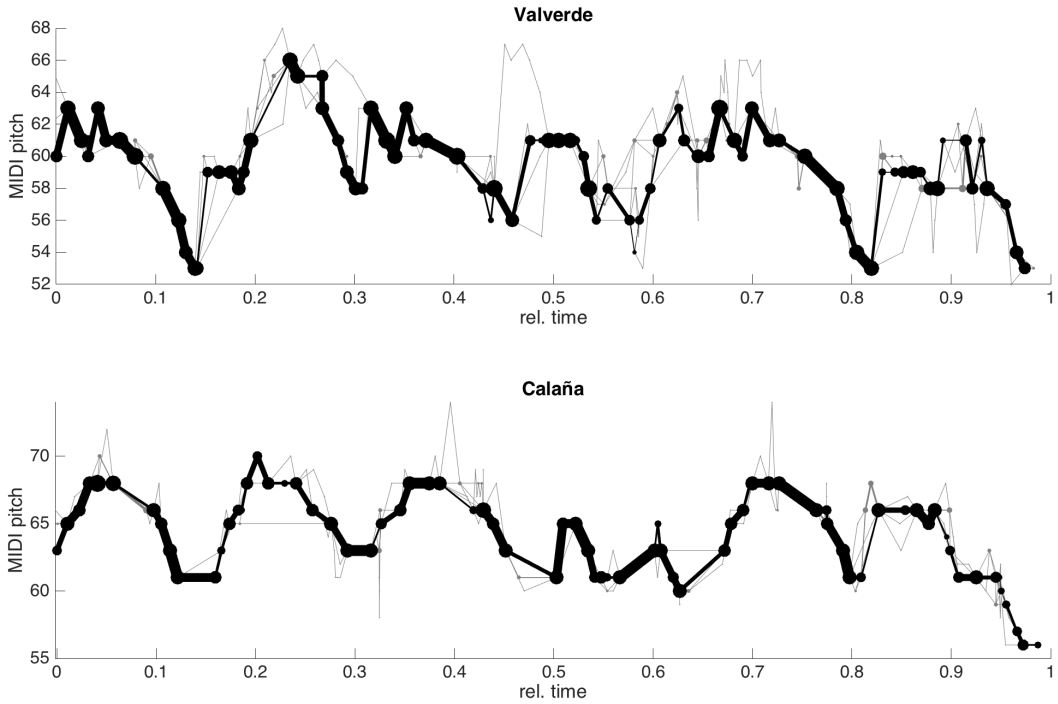


Fig. 4.26 Visualisation of the graph models extracted for *fandangos de Valverde* (top) and *fandangos de Calaña* (bottom). Circle sizes and line widths are proportional to vertex scores and edge weights, respectively. Vertices and edges constituting the template are marked black, others grey.

Furthermore, we can visually identify similarities among the melodic contours of the templates: The underlying melody of both, *fandangos de Valverde* and *fandangos de Calaña*, contains six phrases. In both cases, all phrases except the third, are characterised by a single upward movement followed by a fall of the melody. The third phrase consists of two such upward-downward movements. Analogously, the templates of the two *valiente* styles are similar to each other. Both contain five melodic phrases and the main difference between both styles can be observed in the third phrase, in which the *valiente de Huelva* melody exhibits a characteristic downward movement spanning an octave.

The visualisation furthermore allows us to analyse the local occurrence of melodic variation within a template. We can for example identify that in the *valiente de Alosno* sub-style the strongest variation occurs towards the end of the melody, between 0.8 and 1.0 on the relative time axis. While there seem to be several characteristic notes, which appear throughout most performances, performers tend to insert transitional and grace notes between them, which are not consistent across performances. We can observe a similar, yet less distinctive behaviour for the other *fandango* styles.

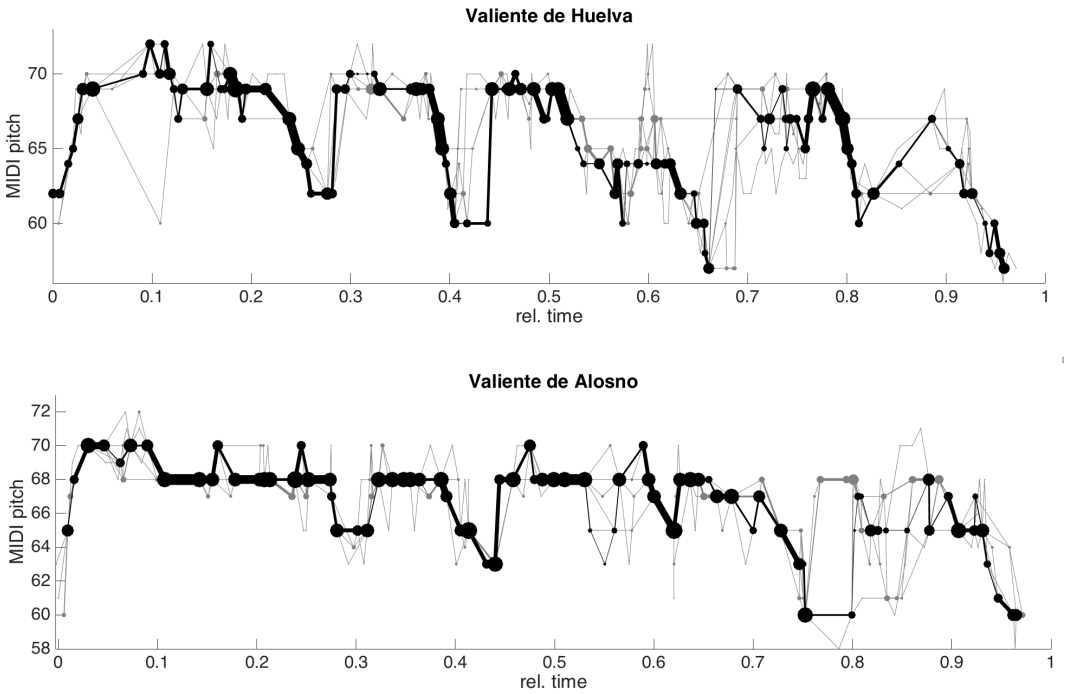


Fig. 4.27 Visualisation of the graph models extracted for the two *fandango valiente* styles: *Valiente de Huelva* (top) and *Valiente de Alosno* (bottom). Circle sizes and line widths are proportional to vertex scores and edge weights, respectively. Vertices and edges constituting the template are marked black, others grey.

style	melodic stability r
valverde	.62
calaña	.64
valiente de Huelva	.50
valiente de Alosno	.50

Table 4.4 Melodic stability for six different styles *(MT)*.

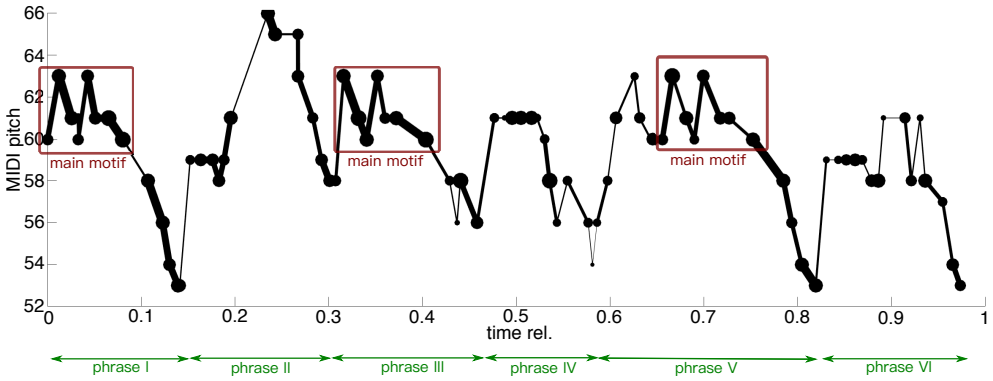


Fig. 4.28 Visualisation of the *valverde* template with annotations of phrases and occurrences of the main motif. Circle sizes and line widths are proportional to vertex scores and edge weights, respectively. Vertices and edges constituting the template are marked black, others grey.

Figure 4.28 shows the vertices and edges which form part of the melodic skeleton of the *valverde* sub-style together with manual annotations of phrase boundaries and occurrences of the main motif. It can be seen that the least stable sections are located at the second half of the fourth and during the last phrase. We furthermore observe that the characteristic melodic motif, which is unique to this sub-style, is largely preserved during performance.

After having demonstrated how the graphical model can be used to identify common trends among performances of the same style, we now proceed with an analysis of single performances with respect to the underlying melodic template. Given a performance transcription, we align it to a graph model and interpolate the onsets of unmatched notes as described in Section 4.4.2. We then visualise the extracted template together with the aligned performance transcription. This allows us analyse in how far a particular singer follows the template during performance and where and how variation is introduced.

An example of two *fandango de Valverde* performances is shown in Figure 4.29: The performance by *Paco Marín* (top) can be considered a typical interpretation of this style. The singer largely follows the underlying template and variation occurs in form of minor melodic deviations (i.e. the beginning of the last phrase around 0.85 on the relative time axis differs by one semitone) and insertion of grace notes (i. e. around 0.23). The second performance by singer *Corbacho* (bottom) is an example of a rather unconventional performance of this style. In addition to minor ornamentation, this performance exhibits major deviations in the melodic contour with respect to the template. The most salient aspect is the fact that the main motif is not preserved. Additional contour differences occur around 0.25 and 0.45 on the relative time axis.

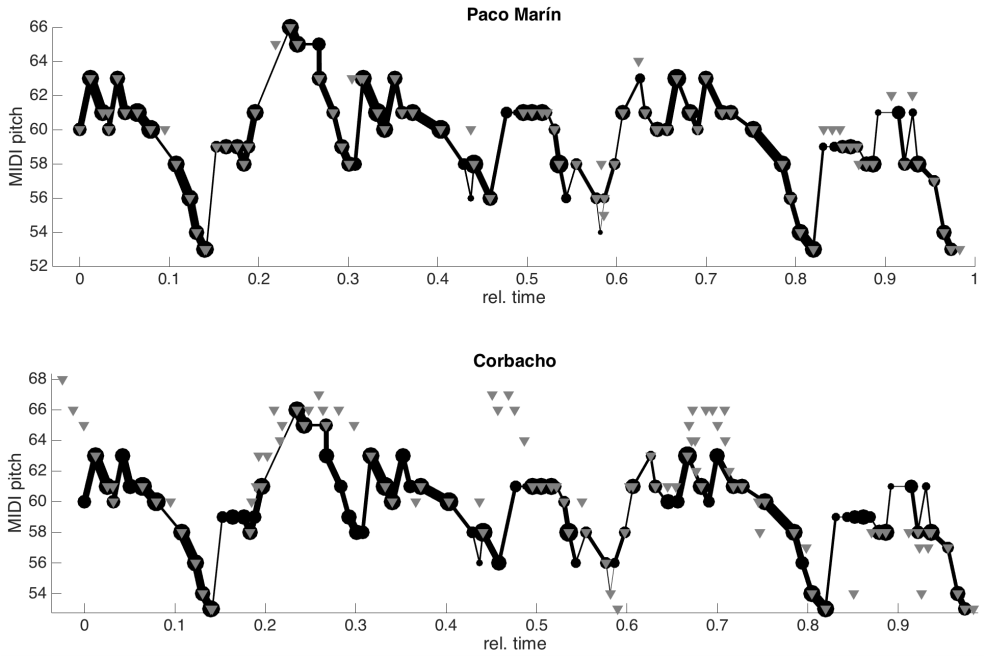


Fig. 4.29 Visualisation of two *valverde* performances with respect to the extracted template: *Paco Marín* (top) and *Corbacho* (bottom). The template is drawn in black and grey triangles indicate notes sung during performance.

4.4.5 Open Problems

An interesting aspect for future research is the evaluation of the quantitative evaluation of the proposed method on a large corpus of performances where the underlying melodic template is known. In this context, it would furthermore be of interest to investigate the influence of the alignment order. Also, while this work has mainly focused on the pitch domain for the specific case of *flamenco* music, future work includes the extension of the framework to the rhythmic domain as well as the application to other genres. The model construction can be further automated by automatically locating relevant segments in recordings containing various melodies or several repetitions of the same melody. Furthermore, the method should be evaluated in the context of a large-scale automatic melody classification task to gain more insight into its reliability and scalability.

4.5 Conclusions

In this chapter, motivated by the concept of constant reinterpretation melodies in flamenco music, we introduced the new computational task of melodic template extraction and presented two very distinct approaches.

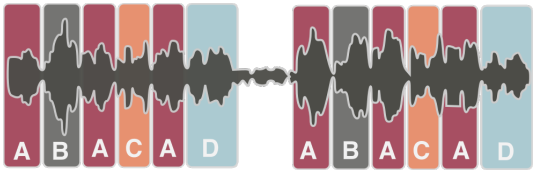
In Section 4.3, we formulated the task as a geometric optimisation problem which we introduced as the *Spanning Tube Problem*. Melodies are modelled as polygonal curves in the time-pitch space with m links and the goal is to compute a new polygonal curve, representative of the template, which fits a fixed number p of similar items at each point in time. The particular challenge of this task is that the parameter p corresponds to an amount of melodies, but does not refer to a specific set of melodies. We proposed an algorithm which solves the optimisation problem of finding the optimal tube width by performing binary search on a discrete set of candidate values. The obtained time complexity is $O(n^2 m \log n \log nm)$. We also showed how a particular case can be solved in linear time with a more elaborate approach. In a case study, we demonstrated how the spanning tube can be used to quantify the amount of variation among a set of melodies. In addition, several possible extensions to the algorithm and related open problems were identified.

A second approach to the same computational task was proposed in Section 4.4. We presented a progressive multiple sequence alignment procedure which operates on note-level transcriptions and yields a graph model holding information on the frequency of notes and note transitions across performances. We described in detail how we extract a melodic sequence from this model which approximates the melodic template inherent to the analysed performances and derived a metric for melodic stability. The algorithm extracts the template in $O(nm)$ with a pre-processing time of $O(n^2 m^2)$. We furthermore presented two strategies to estimate the melodic similarity between a given melody and multiple performances based on the proposed model. We demonstrated in an experimental evaluation that this method yields comparative performance to the computationally more expensive classification based on pair-wise comparison with all database instances. In addition, we showcased how visualisations of the graph model are a useful tool for comparative performance evaluation, providing insight into the amount and location of variation introduced by performers.

In the context of flamenco music, a major limitation for the development and evaluation of melodic template extraction methods is the lack of large annotated datasets for this task. This issue, which also applies to melody classification, has been previously been addressed in the conclusions of Section 2. It would furthermore be beneficial to compare automatically extracted template representations to templates defined by flamenco experts. Analysing the agreement among various experts and their criteria in the annotation process can provide valuable insights which can be included in computational models. Lastly, it would be interesting to investigate the potential use of proposed methods in the context of music education.

Chapter 5

Discovering Melodic Repetition



5.1 Introduction

Repetition is a fundamental concept in music, which allows the listener to perceive structure via the creation of temporal relationships within a composition. The perception of repetition, its influence on listener preference, and the related concept of perceived music similarity are frequently addressed topics in the literature [124, 196, 16, 147, 111]. Systems which can automatically discover repeated melodic patterns in a musical piece do not only provide a powerful tool for content-based description, retrieval and visualisation, but have also shown to yield interesting musicological insights, in particular when applied to largely undocumented non-Western music genres [78, 30].

Several computational studies have focused on the detection and analysis of melodic repetition in monophonic and polyphonic **symbolic music representations**. For an exhaustive review and comprehensive taxonomy, the reader is referred to [91]. Most such methods are not restricted to a specific hierarchical unit (i.e. phrases or motifs) and aim at the detection of melodic repetition in general, without setting any constraints on pattern length and boundaries. Those studies have revealed that only a fraction of all detected repetitions are actually perceptually relevant [91] and filtering out irrelevant patterns has turned out to be an important problem in such systems.

Approaches targeting the extraction of repetition directly from the audio signal have been mainly developed in the context of **music structure analysis** [138] and **audio-based pattern discovery** [33]. Music structure analysis refers to the task of segmenting a recording

into adjacent, non-overlapping segments and detecting repetitions among them. In the context of structural annotation of folk music, prior work has targeted the structural unit of stanzas [140, 139, 11] in singing recordings without accompaniment. Analogously to melodic pattern detection in scores, audio-based pattern discovery methods attempt to detect repeated segments of variable length. In [78], a large corpus of Indian Art Music is mined by comparing fixed-length segments using dynamic time warping. In an attempt to bridge the semantic gap between audio signals and symbolic representations, [29] applied a symbolic pattern detection approach to automatic polyphonic transcriptions. In [223], repeated themes are detected using techniques from the field of information dynamics. In [214], a silence-based segmentation scheme was used to identify cognitively meaningful units that can be efficiently used for tune family recognition.

In this chapter, we propose a system which automatically discovers **phrase-level repetition in folk singing recordings**, leading to an automatic structural annotation of a piece, as shown in Figure 5.2. The structure of European folk music is heavily based on repeated melodic sections at various hierarchical levels, e.g. *motifs*, *phrases* and *stanzas* [222]. The same observation can be made for certain flamenco styles, which are close to their folkloric origin. These include, for example, the *fandangos de Huelva*. While in the context of classical, jazz and popular music, the term *phrase* usually refers to the rather loose concept of a "complete" or "resolved" musical segment, in folk music, a *phrase* is a clearly defined, organisational unit [121]. A *stanza* or *verse* can be subdivided into several perceptually identifiable *phrases*, which can again be decomposed to smaller units such as *motifs* and *melodic cells*.

The task of detecting repeated sung phrases in folk music recordings differs from the previously studied scenarios in several aspects:

- The task at hand can be formulated as a pattern detection scenario, where the endpoints of the patterns correspond to phrase beginnings and endings. However, existing approaches to pattern discovery do not demand that the pattern endpoints are located at phrase boundaries. Instead, patterns can be of variable lengths and may even temporally overlap other patterns.

- We are specifically targeting repetition in the singing voice, which implies the additional challenge of detecting vocal segments. In contrast, in the symbolic and audio-based pattern detection task, repeated patterns are generally not restricted to a particular voice. Similarly, in the music structure analysis task, every frame of the audio file is assumed to belong to a segment, whereas in our scenario only vocal sections are relevant.

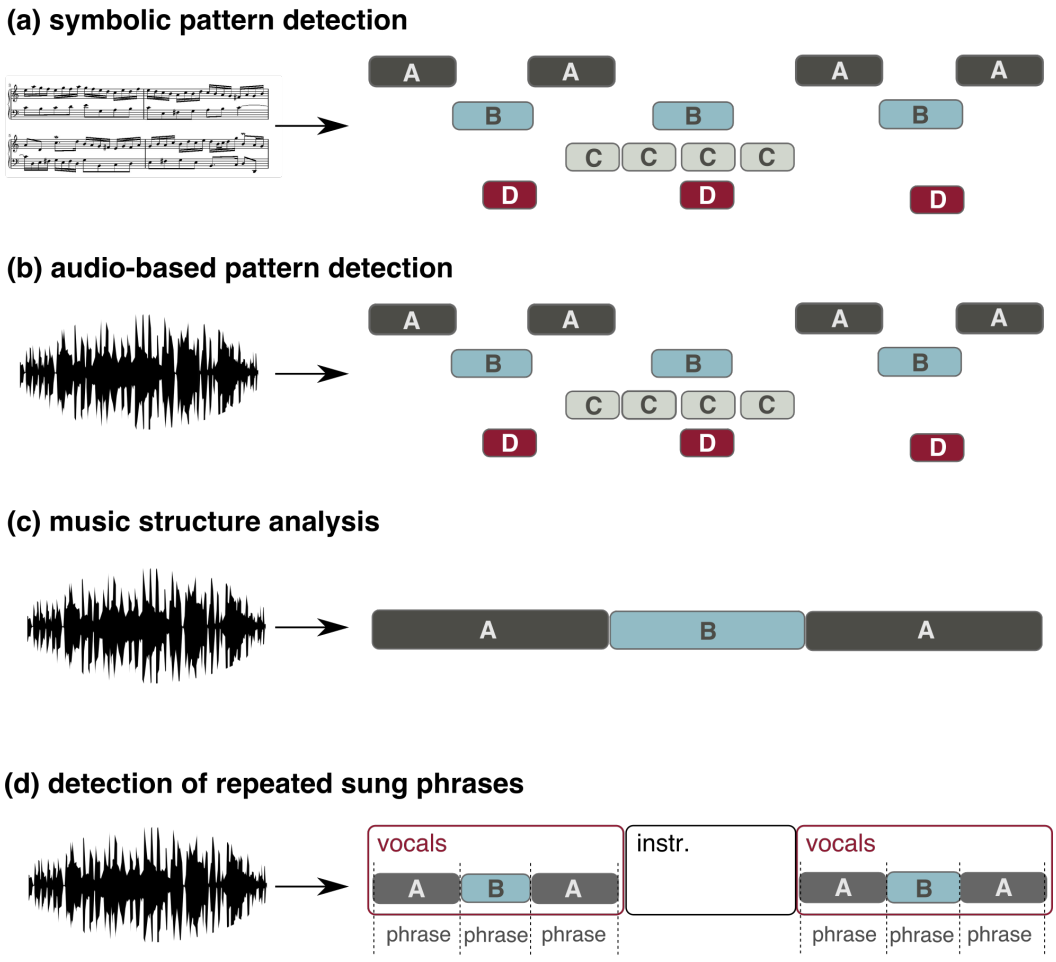


Fig. 5.1 Schematic view of the differences between symbolic pattern detection, audio-based pattern detection, music structure analysis and the discovery of repeated sung phrases.

- The proposed system operates on automatic performance transcriptions and we expect to encounter variation among instances of the same pattern, which is introduced by ornamentation, inaccurate intonation and transcription errors.
- In contrast to the work described in Chapter 3, which focused on the detection of a pre-defined melodic pattern, the task addressed in this chapter is entirely unsupervised. We do not assume any prior knowledge on the melodic content or the number of distinct patterns and their occurrences in the audio recording.

The work in [167] has explored phrase-level repetition in the context of sheet music in order to achieve a repetition-based segmentation of melodies. However, the automatic clustering and respective annotation of the repeated patterns as groups was not addressed by that method. A preliminary audio-based attempt to discover phrase-level repetition was

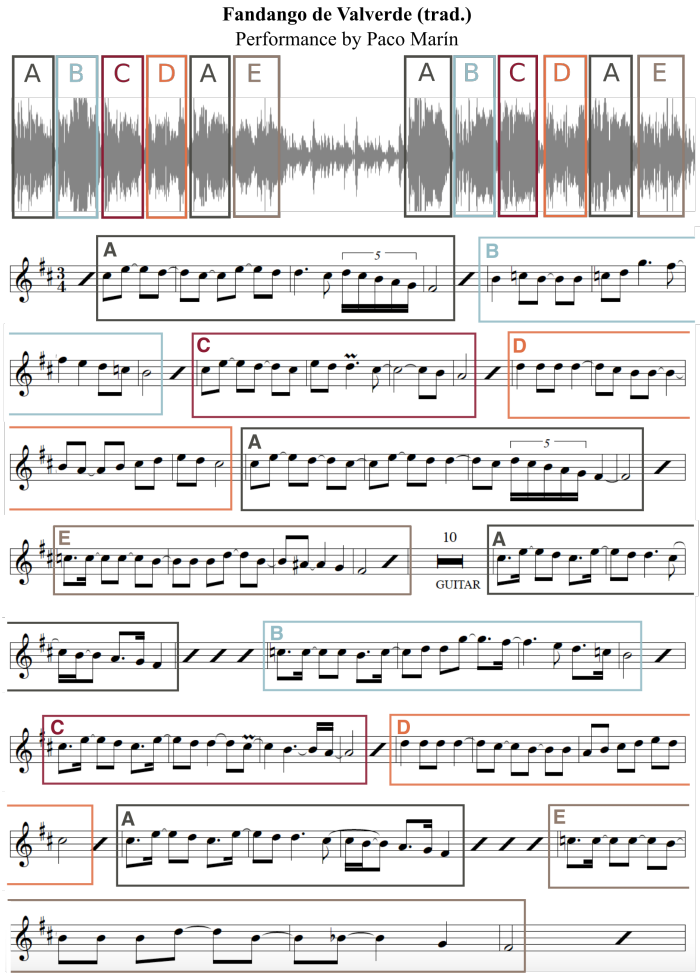


Fig. 5.2 Structural annotation of a *Fandango de Huelva* recording with respect to sung phrases: Audio annotation (top) and the corresponding phrases marked on a manual note transcription (bottom).

proposed in [106] in the context of accompanied flamenco singing: at a first stage, sung phrases were detected using a vocal detection algorithm and, then, at a second stage, pair-wise alignment of chroma-based representations of the detected vocal segments was performed and groups of similar phrases were formed using a frame-centric clustering scheme.

Figure 5.1 provides a schematic illustration of the differences between the task at hand and the related tasks of symbolic pattern detection, audio-based pattern detection and music structure analysis. An example of phrase-level repetition in the context of flamenco and the desired structural audio annotation is shown in Figure 5.2.

The proposed method and a detailed technical description of all its processing stages is provided in Section 5.2 and evaluation metrics for the task are introduced in Section 5.3. The results of an experimental evaluation in the context of three European folk traditions is presented in Section 5.4 and the chapter is concluded in Section 5.5.

5.2 Discovery of repeated sung phrases

In the following, we introduce a method for detecting repeated sung phrases directly from audio recordings that contain a cappella or accompanied singing performances. The system operates on automatically extracted, most likely noisy, note-level transcriptions of the singing voice melody from such recordings. The resulting symbolic representation contains only the melody of the singing voice, whereas instrumental interludes are encoded as silence. The note transcription stage employs off-the-shelf, state-of-the-art algorithms, which are expected to yield a varying percentage of transcription errors. Instead of focusing on reducing such errors at a post-processing stage, we deal with them during the subsequent stages of our approach. Furthermore, instances of a repeated pattern may exhibit melodic variation introduced by the performer, either in the form of intonation errors or consciously, as an expressive asset. Therefore, we design segmentation and clustering stages that exhibit a certain robustness to transcription errors and melodic variation among repetitions. Specifically, with a simple and efficient procedure, we first segment an automatically generated transcription into phrases by exploiting basic musicological characteristics of phrase boundaries in folk music. We then investigate a number of commonly used distance metrics to compute pair-wise melodic distances among the detected segments. Finally, a clustering scheme is employed to identify clusters (categories) of repeated phrases. The output of our system is an annotation of the audio recording (Figure 5.2, top), where phrase boundaries and repetitions have been automatically labelled.

Overall, the novelty of our method lies in the combination of the following elements: a new problem formulation and a new approach for solving the problem as a pipeline of building blocks, which operate on the intermediate layer of noisy transcriptions and not on the low-level layer of audio descriptors.

We provide a detailed evaluation of the proposed method on different European folk music traditions and investigate the influence of transcription and phrase segmentation errors on system performance. Our method yields convincing results and outperforms the approach in [106]. Furthermore, we demonstrate, via a comparative performance analysis and by analysing the behaviour on selected examples, that representative methods for the more generic tasks of structural analysis and pattern discovery fail to address the current task of phrase-level annotation.

An overview of the proposed system is shown in Figure 6.4. Starting from a raw audio recording, we first employ an automatic singing transcription algorithm to estimate a symbolic representation of the vocal melody, acknowledging that the resulting transcription will inevitably contain a percentage of errors. At a second stage, a novel, computationally efficient, phrase segmentation method, which exploits certain musical properties typically encountered in folk songs, segments the previously extracted transcription into a set of N subsequences corresponding to sung phrases. Then, all pair-wise melodic distances among the

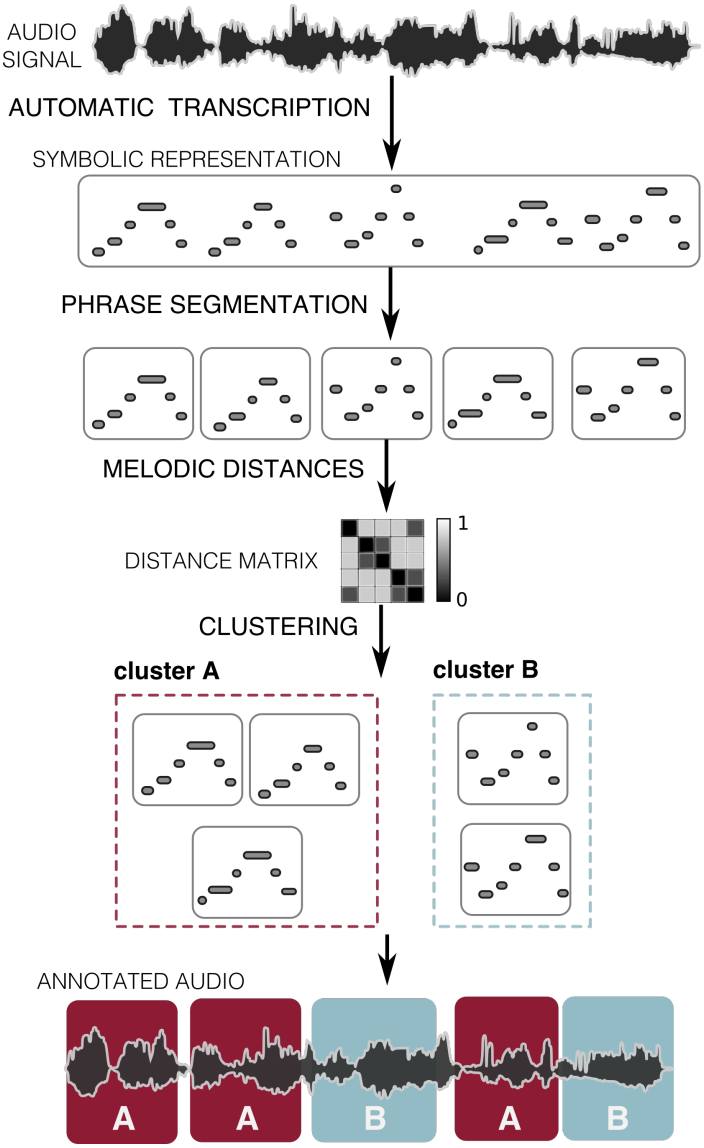


Fig. 5.3 Discovery of repeated melodic phrases: system overview.

detected subsequences are computed, resulting in a $N \times N$ distance matrix, $\mathcal{D}$. To this end, we investigate various distance measures commonly used in the context of music similarity. Finally, a standard clustering algorithm receives the computed distance matrix and groups the phrases into k categories, where k is automatically estimated on a song basis. As a result, each cluster corresponds to a prototypical melodic pattern and the members of the cluster to the occurrences of the pattern.

5.2.1 Music datasets

We investigated the automatic discovery of repeated sung phrases in three collections with distinct musical characteristics: a subset of the *Onder de groene linde* dataset (DFS) containing amateur recordings of dutch folk songs, a collection of *Fandangos de Huelva* (FH), a sub-genre of *flamenco* music, and a set of Greek *Rebetiko* recordings (REB). A musically trained individual annotated manually all three datasets regarding phrase boundaries and phrase repetitions. In addition, for the FH dataset, a flamenco expert with formal music education provided manual transcriptions of the singing voice melody. The adopted datasets are described below in more detail. In the following, the abbreviation ALL stands for the union of all three datasets. A summary of the number of phrases and repeated patterns and their distributions in the three datasets is given in Table 5.1. It is important to note, that throughout the datasets, all phrases are repeated at least once.

Dutch folk songs (DFS)

The *Onder de groene linde* audio dataset is part of the publicly available *Meertens Tune Collection* [212] which contains more than 7000 recordings of amateur singers performing traditional Dutch folk tunes without instrumental accompaniment. Recordings contain one or more verses of a tune. We selected a subset of 50 songs, for which a total of 1235 phrases were manually annotated to serve as ground-truth data. The annotated phrases are instances of 194 patterns, with each phrase appearing approximately 6 times on the average.

Fandangos de Huelva (FH)

The *Fandangos de Huelva* collection contains 11 commercial recordings and has been previously used to evaluate the algorithm in [106], which serves as a baseline method in this study. In all recordings, the singing voice is accompanied by guitar playing. We manually annotated 176 phrases, from which 52 distinct phrases can be extracted (on the average, approximately 3 repetitions per distinct phrase). For all songs, a flamenco expert additionally provided manual corrections of the automatically generated transcriptions, where pitch, segmentation and vocal detection errors were corrected. Such errors include melodic lines stemming from the accompaniment that were mistakenly transcribed as singing voice melody or singing voice

	DFS	FH	REB
number of recordings	50	11	30
number of phrases per song (min. / median / max.)	8/22/64	12/12/30	12/24/33
total number of phrases	1235	176	711
number of repetitions per phrase (min. / median / max.)	2/6/22	2/2/9	2/4/9
total number of unique patterns	194	52	178

Table 5.1 Statistics of the three datasets with respect to annotated phrases and pattern repetition

segments which were mistakenly classified as accompaniment and therefore not transcribed. These ground truth transcriptions were used in the glass ceiling analysis described in Section 5.4.4.

Rebetiko (REB)

In the scope of this study, we also gathered 30 recordings of Greek *Rebetiko* music, available on a video sharing platform, performed by the renown singer *Rita Abatzi*. Rebetiko is an art form of Greek urban songs which appeared towards the end of the 19th century and was shaped to the style that we know today until the third decade of the 20th century. It is a blend of elements from Greek music traditions and influences stemming from the Greeks of Asia Minor. In the recordings that we studied, the singing voice is accompanied by instrumentation including the violin, the bouzouki and the guitar. A total of 711 phrases were manually annotated, stemming from 178 distinct patterns, with each distinct pattern appearing 4 times on the average.

5.2.2 Automatic transcription

The first stage of our method employs an Automatic Music Transcription (AMT) algorithm to extract a symbolic representation of the melodic content of the audio recording. We aim to transcribe the melody of the singing voice from monophonic and polyphonic recordings. In our study, the term polyphonic refers to the case of a dominant singing voice in the presence of accompanying instruments. As it is mentioned in [8], best results are achieved with AMT systems which are specifically developed for the genre and instrumentation under study. We therefore employ different state of the art singing transcription systems which are selected based on their suitability for the music corpora at hand.

More specifically, the DFS dataset is transcribed using the *TONY* [127] transcription system, which has been developed for the transcription of monophonic sources. For the *flamenco* recordings of the FH dataset, which contain a large amount of melismatic, microtonal ornamentation and guitar accompaniment, we employ the *CANTE* [104] transcription system, which specifically targets singing transcription from accompanied flamenco recordings. The Greek *Rebetiko* corpus exhibits similar characteristics to the *flamenco* case with respect to ornamentation and pitch instability. Therefore, we apply again the system used for the FH dataset, but we replace the vocal detection stage (which targets the particular case of guitar accompaniment) with a generic vocal detector based on machine learning [188]. This detector has given convincing results for this type of instrumentation [158].

In all cases, the output of the AMT system is a symbolic representation in the form of a sequence of note events, where each note is described by its *onset time*, *duration* and *pitch value*. The *CANTE* method provides semitone-quantised MIDI values, while the *TONY* algorithm does not apply quantisation on the extracted frequencies. In the latter case, we quantise to the closest semitone.

5.2.3 Structural properties

In this section, we demonstrate three structural properties which serve as basic assumptions for the proposed phrase segmentation algorithm and, to this end, we analyse the manually annotated data with respect to phrase durations and occurrence of silences. To proceed, we first introduce some notation and definitions. Specifically, let us represent an automatic transcription as an ordered sequence, $\mathbf{x}$, of M notes,

$$\mathbf{x} = (x_1, ..., x_M)$$

The i-th note, $x_i = (x_i^o, x_i^d, x_i^p, x_i^s)$, is described by its onset time, x_i^o, note duration, x_i^d, pitch, x_i^p and silence, x_i^s, where the term silence stands for the time duration between the note offset and the onset of the next note. Our goal is to split a given transcription, X, into a set of N non-overlapping continuous segments corresponding to melodic phrases. Let $\mathcal{S} = \{S_1, ..., S_N\}$ be a segmentation of $\mathbf{x}$, where $S_j = (x_{L_j}, ..., x_{K_j}), 1 \leq j \leq N, S_j \in \mathcal{S}$, is a continuous subsequence of $\mathbf{x}$ (Figure 5.5 (b)), starting at the L_j-th note and ending at the K_j-th note.

We denote the sum of note durations of the j-th segment, S_j, with τ_{S_j},

$$\tau_{S_j} = \sum_{n=L_j}^{K_j} x_n^d \tag{5.1}$$

For the sake of simplicity, we will refer to τ_{S_j} with the term *phrase duration*, even though the silence between notes, although part of the phrase, has been ignored in this definition.

Furthermore, let $\hat{\tau}$ be the phrase duration, if a transcription, $\mathbf{x}$ (M notes long), is segmented into N segments of equal length, i.e.,

$$\hat{\tau} = \frac{\sum_i^M x_i^d}{N} \tag{5.2}$$

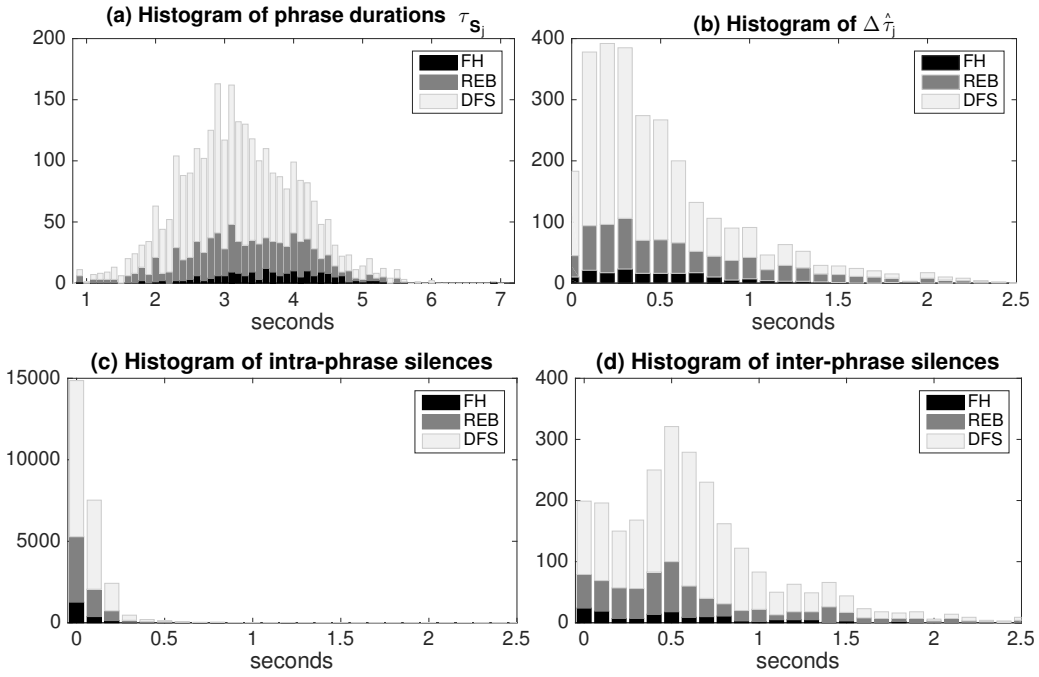


Fig. 5.4 (a) Histograms of phrase durations (τ_{S_j}), (b) histogram of absolute duration differences, $\Delta\hat{\tau}_j$, (c) intra-phrase silence durations, and (d) inter-phrase silence durations.

Figure 5.4(a) displays the histogram of τ_{S_j}, for the case of manually annotated phrases over all datasets under study. It can be seen that for the majority of phrases (92%) τ_{S_j} ranges between $\tau_{\min} = 2.0$ and $\tau_{\max} = 5.5$ seconds. Figure 5.4(b) shows the histogram of the absolute duration difference, $\Delta\hat{\tau}_j$, of manually annotated phrases from $\hat{\tau}$. It can be seen that $\Delta\hat{\tau}_j$ is smaller than 1 s for the majority of phrases (90%). It is worth noting that similar observations have been reported for Irish traditional tunes which contain two-bar phrases that are only occasionally fused or split as a means of expressive performance [129]. Further support to our analysis can be found in [196], where a phrase length rule, that favours phrases containing between 6 and 10 notes, was proposed in the context of folk music segmentation. These values were determined based on an analysis of the *Essen folk song collection*[1].

[1]http://www.esac-data.org

Figures 5.4(c) and 5.4(d) demonstrate that inter-phrase silence (Figure 5.4(d)) tends to be longer than intra-phrase silence (Figure 5.4(c)), indicating that phrase boundaries tend to coincide with vocal rests. This observation is also in line with one of the grouping preference rules of *Generative Theory of Tonal Music* [117], which states that phrase boundaries are likely to occur at rests that are longer than intra-phrase silences. Similar observations have been made in listening experiments [49] and data-driven studies [166, 61].

Consequently, we base the design of our phrase segmentation algorithm on the following three observations:

1. Phrases cannot be of arbitrary length and global lower and upper phrase duration bounds can be assumed for a given corpus.
2. Within a song, phrase duration exhibits a relatively small deviation.
3. The end of a phrase is likely to be followed by a vocal rest that is longer than the average inner-phrase silence.

In other words, our goal is to split a note transcription into a number of phrases which are of similar duration, assuming that the phrase length lies within a pre-defined duration range. Furthermore, long silences should more likely occur at phrase boundaries and less likely within a phrase.

5.2.4 Phrase segmentation

Based on the three structural properties described above, we aim for a segmentation, $\mathcal{S} = \{S_1, ..., S_N\}$, where:

(a) the τ_{S_j}'s exhibit a low deviation from the expected $\hat{\tau}$ value

(b) intra-phrase silences are short compared to inter-phrase silences

This is achieved with a computationally simple algorithm that combines the above objectives to produce the desired segmentation of a given transcription sequence. As the number of phrases, N, in a transcription, is not known in advance, we first define a segmentation algorithm given the number of phrases. Then, we compute a segmentation for each candidate value of N, assess the quality of each segmentation based on the criteria established above and, finally, select the highest quality segmentation.

We now describe the segmentation algorithm when the number of phrases, N, is assumed to be known (Figure 5.5). In order to identify the j-th phrase boundary, $j = 1, \cdots, N-1$, we assign a segmentation score, $\lambda_j(x_i)$, to each note, x_i, that quantifies its fitness as a cut-off candidate (Figure 5.5 (b)). The score of x_i is defined as

$$\lambda_j(x_i) = x_i^s \cdot \phi(\mathbf{x}, i, j) \tag{5.3}$$

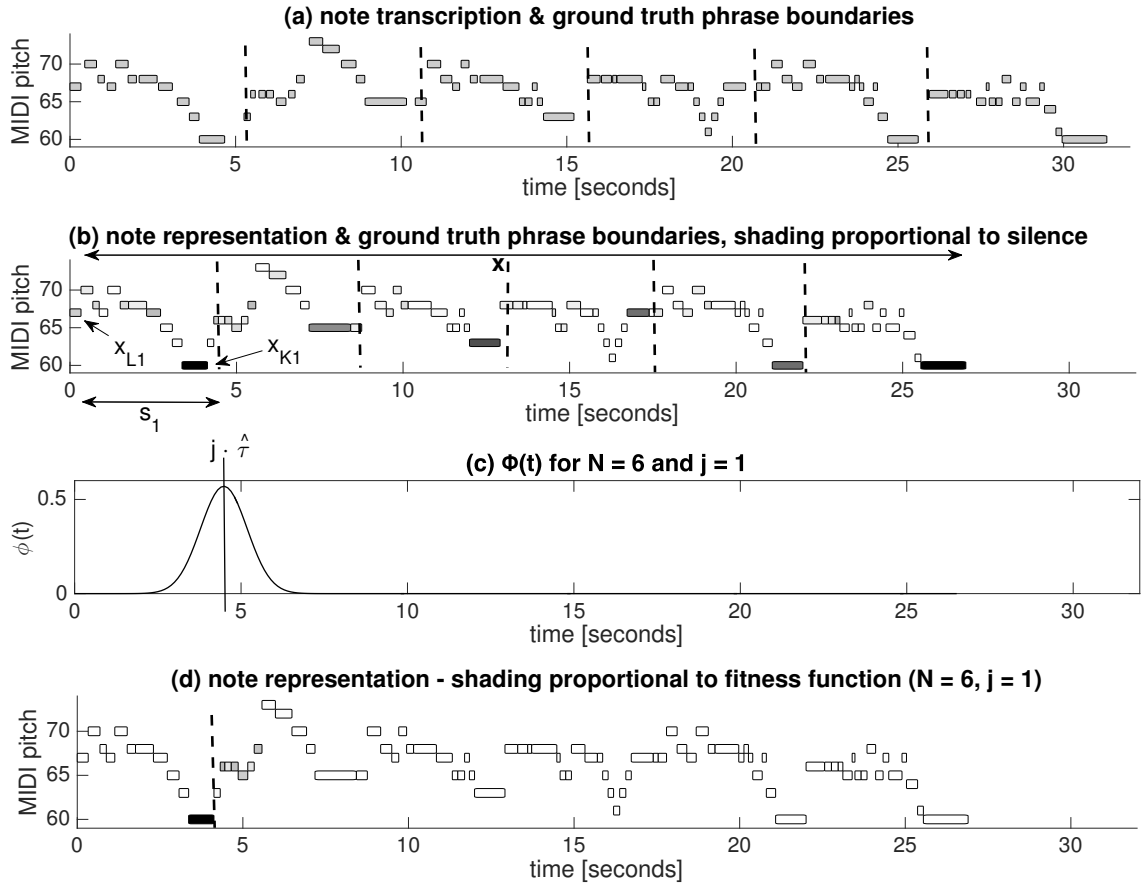


Fig. 5.5 Phrase segmentation: (a) automatic transcription, (b) note representation with marked ground truth phrase boundaries ($N = 6$) with shading proportional to x^s, (c) $\Phi(t)$ for $j = 1$ and $N = 6$, and (d) note representation and first estimated phrase boundary, shading proportional to $\lambda(x)$.

where ϕ is a Gaussian distribution with the mean corresponding to the optimal temporal segmentation position, $j \cdot \hat{\tau}$, and a standard deviation, $\sigma_{\hat{\tau}}$, which is estimated from the dataset (Figure 5.5 (c)), i.e.,

$$\phi(\mathbf{x}, i, j) = \frac{1}{\sqrt{2\pi}\sigma_{\hat{\tau}}} e^{-\frac{\left(\sum_{n=1}^{i} x_n^d - j \cdot \hat{\tau}\right)^2}{2\sigma_{\hat{\tau}}^2}} \tag{5.4}$$

The Gaussian is evaluated at the accumulated duration of the first i notes. It can be seen from Eq. (5.3) that the segmentation score is related directly to the objectives of equal phrase length and long inter-phrase silence. Specifically, function $\phi(\mathbf{x}, i, j)$ takes its maximum value at $j \cdot \hat{\tau}$. Segmenting at these positions for $j = 1, \cdots, N-1$ would yield phrases of equal length.

However, in practice, a segmentation at this exact location is usually not possible, since a boundary can only be located after a note. In fact, several possible phrase boundaries may be located in close proximity to $j \cdot \hat{\tau}$. Therefore, factor x_i^s, which denotes the silence duration after note x_i, ensures that high segmentation scores will be assigned to notes that end close to $j \cdot \hat{\tau}$ and are followed by a long silence. It can be inferred, that $\phi(\mathbf{x}, i, j)$ is evaluated on the accumulated note duration to which silences and instrumental interludes do not contribute. Furthermore, the contribution of factor x_i^s will force a phrase to end before long instrumental sections or silences. Thus, for each $j = 1, \cdots, N-1$, the j-th phrase boundary is placed after the note, $x_{i_{\max}}$, that maximises the segmentation score, i.e., $x_{i_{\max}} = \arg\max_i \{\lambda_j(x_i)\}$ (Figure 5.5 (d)).

We now elaborate on the computation of the estimate of the number of phrases, N. First, by using the aforementioned upper and lower phrase duration boundaries ($\tau_{\min}$ and $\tau_{\max}$) and $\hat{\tau}$, we reduce the candidate values for N to a small set of integers, $\mathcal{N}$. In other words, we only consider positive integer values for N which yield phrase durations that fall into a predefined phrase duration range.

For each $N \in \mathcal{N}$, we compute (using the algorithm above) an optimal segmentation $\mathcal{S}_N$ ($\mathcal{S}$ for short), where each segment is denoted, as before, with $S_j = (x_{L_j}, ..., x_{K_j}), 1 \leq j \leq N, S_j \in \mathcal{S}_N$. We then assign a cost, $c(N)$, to the extracted segmentation for N

$$c(N) = r_{\mathcal{S}} \cdot \sigma_{\tau,\mathcal{S}} \tag{5.5}$$

where $r_{\mathcal{S}}$ is the ratio of mean inner-phrase to mean inter-phrase silences, given by

$$r_{\mathcal{S}} = \frac{\frac{1}{M-N+1} \sum_{j=1}^{N} \sum_{i=L_j}^{K_j-1} x_i^s}{\frac{1}{N-1} \sum_{j=1}^{N-1} x_{K_j}^s} \tag{5.6}$$

and $\sigma_{\tau,\mathcal{S}}$ is the standard deviation of durations of the estimated phrases from their mean value. In line with the criteria established above, the cost $c(N)$ will be low if the durations of the resulting phrases exhibit low variation and silences within phrases tend to be shorter than silences at phrase boundaries. Consequently, the value in $\mathcal{N}$ with the lowest computed cost corresponds to the best phrase segmentation.

In our experimental evaluation, parameters $\tau_{\min}$ and $\tau_{\max}$ are estimated directly from the training data and $\sigma_{\hat{\tau}}$ is estimated via an exhaustive search procedure (see Section 5.3.3). Pseudo-code for the segmentation of a given transcription $\mathbf{x}$ into phrases $\mathcal{S}$ is given in Algorithms 5.1 and 5.2.

5.2.5 Melodic distance computation

As our method is targeting recordings of performances and not written scores, we expect that repetitions of phrases will not to be exact; they are likely to exhibit melodic and/or

Algorithm 5.1 Segmentation of a transcription $\mathbf{x}$ into N phrases

$N \leftarrow$ number of phrases
$M \leftarrow$ length of $\mathbf{x}$
$\mathcal{S} \leftarrow$ empty cell array of length N
prev $= 1$
for $j = 1$ to $N - 1$ **do**
  $\lambda \leftarrow$ array of length M filled with zeros
  for $i =$ prev to $M - 1$ **do**
    $\phi = \frac{1}{\sqrt{2\pi}\sigma_{\hat{\tau}}} \exp(-\frac{\left(\sum_{n=1}^{i} x_n^d - j \cdot \hat{\tau}\right)^2}{2\sigma_{\hat{\tau}}^2})$
    $\lambda(i) = x_i^s \cdot \phi$
  ind $= \arg\max(\lambda)$
  seg $= \mathbf{x}$[prev : ind]
  prev $=$ ind $+ 1$
  $\mathcal{S}(j) =$ seg
seg $= \mathbf{x}$[prev : M]
$\mathcal{S}(N) =$ seg
return $\mathcal{S}$

Algorithm 5.2 Phrase segmentation of a given transcription $\mathbf{x}$

$\tau_{\min} \leftarrow$ minimum phrase duration
$\tau_{\max} \leftarrow$ maximum phrase duration
$\tau_{\text{sum}} = \sum_i^M x_i^d$
$N_{\min} = \lceil \tau_{\text{sum}} / \tau_{\max} \rceil$
$N_{\max} = \lfloor \tau_{\text{sum}} / \tau_{\min} \rfloor$
$c \leftarrow$ array of length $N_{\max}$ with values +Inf
$\mathcal{S}_{\text{cand}} \leftarrow$ empty cell array of length $N_{\max}$
for $n = N_{\min}$ to $N_{\max}$ **do**
  $\mathcal{S} \leftarrow$SEGMENT_INTO_N_PHRASES($\mathbf{x}, n$) # Algorithm 5.1
  $r \leftarrow$ ratio of mean inner- to intra-phrase silences in $\mathcal{S}$ # eq. 5.6
  $\sigma \leftarrow$ standard deviation of phrase durations in $\mathcal{S}$
  $c(n) = r \cdot \sigma$
ind $= \arg\min(c)$
return $\mathcal{S}$(ind)

rhythmic variation because professional performers use variation as an expressive resource. It is also worth noting that amateur singers may reproduce a phrase incorrectly due to intonation and timing inaccuracies. Furthermore, the stage of automatic transcription may introduce pitch, note segmentation and vocal detection errors. However, irrespective of performance variation and errors, our goal remains to identify clusters of similar phrases. Therefore, after partitioning a transcription $\mathbf{x}$ into N phrases, $\mathcal{S} = \{S_1, ..., S_N\}$, we proceed to the computation of an $N \times N$ distance matrix, $\mathcal{D}$, such that $\mathcal{D}(i,j) = d(S_i, S_j)$, where $S_i, S_j \in \mathcal{S}$ and $i, j = 1, \ldots, N$. The distance function $d(S_i, S_j)$ assigns a dissimilarity value to the alignment of S_i against S_j and, therefore, matrix $\mathcal{D}$ holds all pair-wise distances among the detected phrases.

The computation of melodic (dis)similarity has been a popular research topic for many years, especially for melodies that are represented in symbolic (MIDI-like) formats. For a review of related work, the reader is referred to [137] and [218]. The respective problem for the case of audio signals has mainly been studied in the context of query-by-humming [97] systems. In this paper, we investigate four measures that can be used for the alignment of automatically generated transcriptions and which are commonly used in MIR systems.

Dynamic time warping (DTW)

DTW [108] techniques compute an optimal alignment path and respective matching cost between two sequences of feature vectors. In our study, we first convert the note sequence of each phrase to a one-dimensional representation. Specifically, assuming a short-frame length equal to 0.01 s, the i-th note of the phrase is converted to a sequence of (equal) pitch values, T_i frames long, where $T_i = \text{round}(\frac{x_i^d}{0.01s})$. This type of conversion, which is similar to the note duration encoding shown in Section 2.2.2, ignores intra-phrase silences. Then, if $S_i, S_j \in \mathcal{S}$ are the one-dimensional representations of two phrases with lengths I and J, respectively, we compute the optimal alignment path and infer the warping cost d_{warp} as a measure of their melodic distance. This procedure has been described in detail in Section 2.2.3.

Edit distance (EDIT)

The edit (or Levenstein) distance [118] computes the dissimilarity between two symbol sequences as the cost of the optimal way to transform one sequence to the other by means of insertions, deletions and substitutions. The edit distance has been frequently applied to estimate melodic dissimilarity between note sequences since the early days of MIR ([130, 132]). In order to apply the edit distance in this study, we again convert the note representation of a phrase to a sequence of pitch values, as it was done for the DTW algorithm.

Earth mover's distance (EMD)

The earth mover's distance was originally introduced in the field of image analysis [170] but it has also been used over the years to estimate melodic similarity in the context of various content-based music retrieval methods [204, 205]. To use the EMD measure in this paper, we follow the approach in [204]: melodies are represented as weighted point sets, where pitch x_i^p and onset x_i^o form the coordinates of points on a two-dimensional plane and note duration, x_i^d, becomes the weight of the point. Note onsets and note durations are normalised to the interval $[0, 1]$. EMD is a transportation distance which estimates the minimum amount of effort needed to transform one weighted point set to another.

Shape similarity (SHAPE)

The *shape-time* algorithm [208] is a geometric approach which represents melodies as interpolated curves on a pitch-time plane. Similarity among melodies is estimated through the shape similarity of their corresponding curve representations. This approach has been the best performing algorithm in the MIREX[2] symbolic melodic similarity evaluation framework throughout all editions from 2010 to 2015. Due to the fact that the output of the SHAPE algorithm is a similarity value, we map it to the interval $[0, 1]$ using min-max normalization and then we subtract the resulting value from 1, to produce a dissimilarity value.

5.2.6 Clustering

Starting from the phrase dissimilarity matrix $\mathcal{D}$, we now aim to identify groups (clusters) of similar phrases. Each cluster will contain instances (repetitions) of the same melodic sequence. Formulating this problem as a clustering task, our goal is to assign a label l_j to each phrase S_j in $\mathcal{S} = \{S_1, ..., S_N\}$, where $l_i \in \{1, ..., k\}$, and k is the number of clusters.

Here, we apply the well-known *k-medoids* [92] clustering algorithm, which estimates a partitioning of a distance matrix into k clusters by minimizing the sum of pairwise intra-cluster distances. Since the number of clusters, k, is unknown, we employ the *silhouette* validation method [169] to estimate the value of k.

We remove clusters containing a single instance, because in the data of our study each phrase is repeated at least once. Therefore, such segments either originate from phrase segmentation errors or from dominant instrumental lines that are mistakenly transcribed as vocals. We thus obtain a set, Ξ, of phrase-level clusters, $\Xi = \{\mathcal{Q}_1, ...\mathcal{Q}_k\}$, where the k-th cluster, $\mathcal{Q}_k = \{Q_{k_1}, ...Q_{k_m}\}$, contains m occurrences of the respective prototypical pattern.

[2]http://www.music-ir.org/mirex/wiki/MIREX_HOME

5.3 Evaluation metrics

In this section we describe the evaluation metrics that we applied in order to assess the quality of the proposed phrase segmenter and the system as a whole. As it was stated in Section 5.1, the task at hand differs from audio-based pattern detection and music structure analysis. Therefore, we modified standard evaluation metrics accordingly. The adopted metrics are based on the pattern discovery metrics that have been used over the past few years in the "*Discovery of repeated themes and sections*" task of the MIREX[3] evaluation framework. The MIREX metrics account for the fact that an algorithm may detect pattern occurrences that overlap, but not necessarily coincide, with the annotated ground truth phrases. The MIREX metrics define the following *cardinality score*, $s_c(P,Q)$, to measure the melodic similarity between two note sequences, P and Q

$$s_c(P,Q) = \frac{P \cap Q}{\max\{|P|,|Q|\}} \tag{5.7}$$

where the intersection $P \cap Q$ is the set of the note symbols which both sequences have in common and $|.|$ stands for sequence length. While this score is suitable for algorithms that analyse symbolic data or audio-based scenarios where the written score is available, the task at hand requires a minor modification because ground truth transcriptions are not available for all datasets under study. Therefore, instead of evaluating the intersection of two note-set representations, we assess their temporal intersection, where a segment $P = \{t_P^s, t_P^e\}$ is defined by its left and right endpoints, t_P^s and t_P^e, respectively (both measured in time units):

$$\frac{P \cap Q}{\max\{|P|,|Q|\}} \equiv \frac{\min\{t_P^e, t_Q^e\} - \max\{t_P^s, t_Q^s\}}{\max\{t_P^e - t_P^s, t_Q^e - t_Q^s\}} \tag{5.8}$$

In this way, mistakenly detected patterns in instrumental sections will yield a zero score and detected patterns with endpoints that do not coincide with phrase beginnings and endings will result in a reduced score. This metric was also used to assess the performance of the system in [106]. Having defined the modified score, we can now proceed to the description of the procedures that evaluate the phrase segmentation stage and the system as a whole.

5.3.1 Phrase segmentation

To evaluate the phrase segmenter, we ignore the clustering output and only take into account the endpoints of the detected and manually annotated phrases. In the related MIREX structural segmentation task, the output of a segmentation algorithm is evaluated based on precision and recall of detected boundaries which are within a tolerance range of a ground truth boundary. However, although structural segmentation aims at segmenting a stream

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

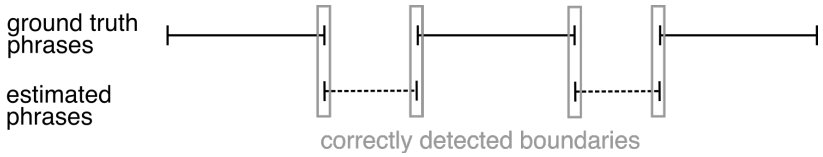


Fig. 5.6 Example of a misleading result when applying boundary detection evaluation to phrase segmentation.

into adjoining sections, sung phrases are not necessarily adjoining and might alternate with instrumental sections, which are not considered phrases. Consequently, evaluating boundary retrieval can be misleading in the context of phrase segmentation. An example is shown in Figure 5.6, where mistakenly estimating all instrumental interludes as phrases yields a boundary recall of 0.67 and a boundary precision of 1.0.

We therefore adapt the MIREX boundary retrieval metrics to the task at hand, by distinguishing between phrase start and end points. A detected phrase onset is accepted as a hit, if it is within a tolerance range around a ground truth phrase onset. Similarly, a phrase offset is accepted, if it is within a tolerance range around an annotated phrase offset. Here, we use a tolerance of 0.5 seconds, as suggested in [203]. Based on the correctly identified phrase on- and offsets, we compute the *phrase precision* ($\mathrm{Phr_P}$), *phrase recall* ($\mathrm{Phr_R}$) and *phrase f-measure* ($\mathrm{Phr_F}$).

In addition, we compute the median deviation in seconds between detected and estimated phrase boundaries as proposed in [203], while respecting again the distinction between phrase onsets and offsets. Specifically, we compute the *median true-to-guess deviation* ($\mathrm{Phr_{TTG}}$) as the median deviation between true phrase boundaries and the closest estimated boundary, and the *median guess-to-true deviation* ($\mathrm{Phr_{GTT}}$)) as the median deviation between estimated phrase boundaries and the closest ground truth boundary.

5.3.2 System as a whole

To evaluate the system as a whole, i.e., the joint performance of the segmenter, the melodic distance measure and the clustering algorithm, we adopt the metrics from the MIREX task on "*Discovery of repeated themes and sections*", but each metric is modified based on Eq. (5.8). The *establishment* metrics, i.e., establishment precision Est_P, recall Est_R and F-measure Est_F, measure the algorithm's capability of detecting that a given ground truth pattern appears at least once as a repetition throughout the recording. In the context of our formulation of the task at hand, the establishment metrics refer to the fact that a phrase is a member of a cluster containing more than one instances. The more detailed *occurrence* measures, i.e., occurrence precision Occ_P, recall Occ_R and F-measure Occ_F, evaluate how many repetitions of a given pattern are successfully detected. An occurrence is considered to be correctly detected if it overlaps with a ground truth pattern instance by more than 75%.

For more details regarding the MIREX definition, the reader is referred to the respective MIREX task description[4].

5.3.3 Cross-fold validation

All experiments described in Section 5.4 are performed in a 5-fold cross-validation scheme. Care is taken so that the instances of each dataset are uniformly distributed over the folds. At each run, we estimate the parameters $\tau_{\min}$ and $\tau_{\max}$ from the train partition as the 5th and 95th percentile of annotated phrase durations, respectively. The parameter $\sigma_{\hat{\tau}}$ is determined based on the highest achieved Est_F value on the train set in an exhaustive search over values ranging from 0.1 to 2.0 in steps of 0.1. The reported performance metrics refer to track-wise values which are first averaged within each run and then across runs. For the generation of genre-specific reports, the same procedure is applied (train and test folds contain data from all databases), but metrics are reported separately for each genre.

5.4 Results

In the sequel, we provide a detailed experimental evaluation of the proposed method. First, we evaluate the phrase segmentation stage by means of phrase precision, recall and f-measure and then, we compute the pattern discovery metrics for different melodic distance measures. We evaluate the system as a whole across datasets and folds, and investigate the influence of phrase segmentation and automatic transcription errors from a glass ceiling analysis perspective. In addition, we provide some examples of automatically generated annotations and give an overview of common errors. Finally, we show the relation of our repeated phrase discovery approach to neighbouring MIR tasks by analysing the output of state of the art algorithms for selected audio examples. We demonstrate that these algorithms are not suitable for the task of repeated sung phrase discovery via a comparative evaluation on the three datasets.

5.4.1 Phrase segmentation evaluation

Figure 5.7 shows the phrase evaluation metrics described in Section 5.3 for all four datasets. We can observe certain performance differences across genres. We obtain a phrase f-measure of 0.87 on the DFS dataset, and only 0.55 on the FH dataset and 0.57 on the REB collection. Through manual inspection we identified vocal detection errors in the two polyphonic datasets, FH and REB, as the main source of error. Specifically, dominant melodic lines from the accompaniment, e.g. from the violin in REB, are mistakenly transcribed, causing the

[4]http://www.music-ir.org/mirex/wiki/2016:Discovery_of_Repeated_Themes_%26_Sections

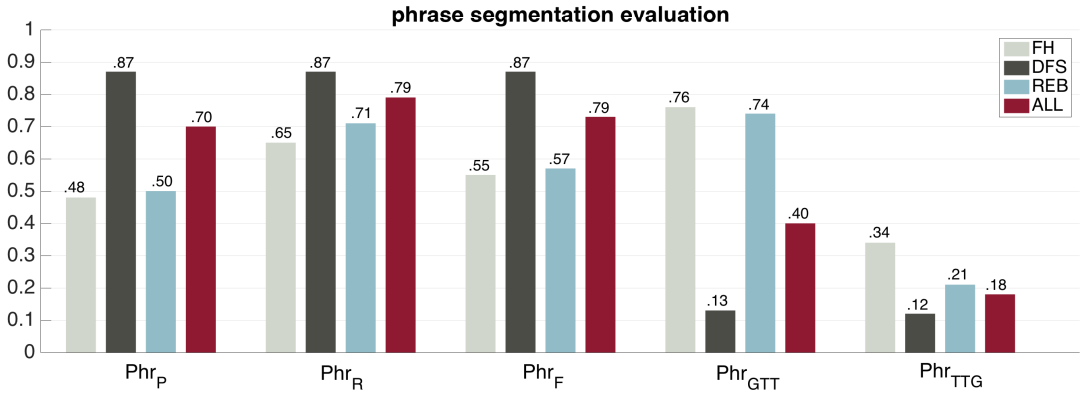


Fig. 5.7 Phrase evaluation metrics across datasets.

segmentation into sections of similar length to fail. Furthermore, for these two datasets, we observe much larger values for Phr_{GTT} compared to Phr_{TTG} ($\text{Phr}_{\text{GTT}} = 0.76$ vs. $\text{Phr}_{\text{TTG}} = 0.34$ for FH and $\text{Phr}_{\text{GTT}} = 0.74$ vs. $\text{Phr}_{\text{TTG}} = 0.21$ for REB). These values indicate a tendency towards over- rather than under-segmentation. On the other hand, for the DFS dataset, these two metrics are more balanced ($\text{Phr}_{\text{GTT}} = 0.13$ vs. $\text{Phr}_{\text{TTG}} = 0.12$).

5.4.2 Comparison of melodic distance measures

We now proceed to the comparison of the four melodic measures described in Section III-C and we investigate how they affect the clustering results. Specifically, we assess the overall system performance by means of the establishment and occurrence f-measures (Figure 5.8). Note, that the phrase segmentation stage is not affected by the choice of melodic distance measure and we therefore do not report the phrase evaluation metrics here.

First of all, it can be seen that all methods give satisfactory performance regarding establishment; they only differ at the second decimal digit. This is explained by the "mild" nature of the establishment measure: if a cluster is detected, even if it only contains one correct pattern and all the other patterns are wrong, the cluster will still contribute successfully to the establishment f-measure.

Secondly, the DTW, EMD and EDIT methods appear to be competitive, also in the case of the occurrence f-measure, which is a "harder"' measure compared to establishment. Again, the performance of these three methods only differs at the second decimal digit. The SHAPE method, on the other hand, has inferior performance from this perspective. It is worth noting that the SHAPE method was proposed in the context of perfect transcriptions, but we are using it in a noisy environment due to the transcription errors that our system has to deal with. Therefore, these errors affect the quality of the interpolated curve that is synthesized by the SHAPE method and this in turn affects its performance. In the light of these findings,

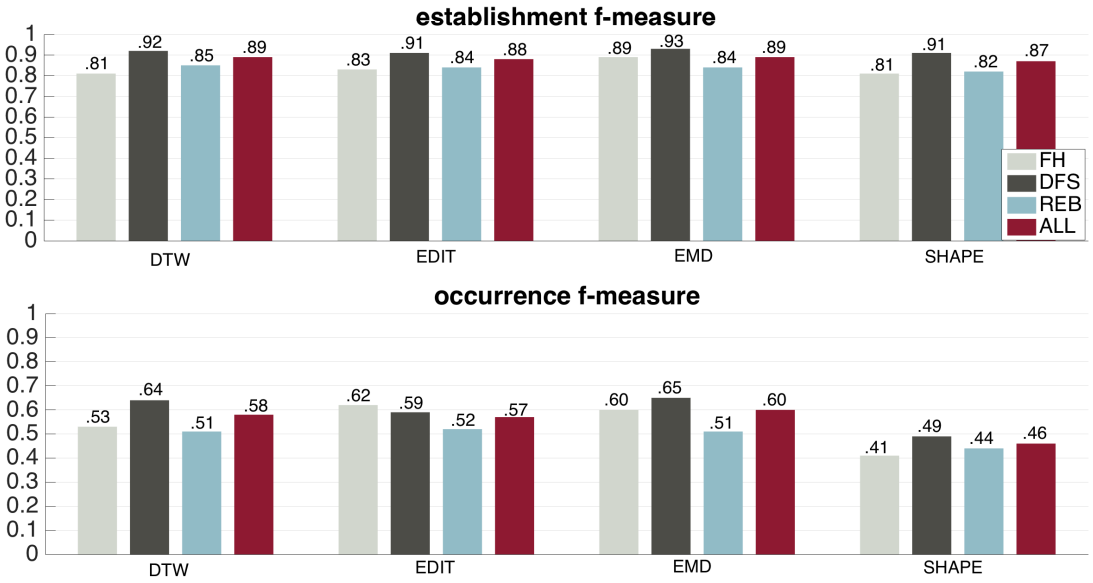


Fig. 5.8 Pattern evaluation metrics for different similarity measures.

we select the EMD method for presenting further performance results, because of its slightly better performance, over all datasets, regarding the occurrence f-measure.

5.4.3 Pattern discovery results across genres and folds

Table 5.2 shows the establishment and occurrence metrics for each one of the three datasets, when the system is evaluated as a whole using the EMD method, starting from automatic transcriptions and going through all the stages of the proposed pipeline. Again, it can be seen, that a higher performance is achieved for the DFS dataset compared to the FH and REB datasets. In particular with respect to the occurrence metrics, which evaluate the detection of all instances of a pattern, for the REB dataset, an F-measure of $\text{Occ}_F = 0.51$ is achieved, which is low compared to the $\text{Occ}_F = 0.65$ value of the DFS dataset. In a manual inspection of the results, we identified vocal detection errors and stronger melodic variation between instances of the same patterns as the main sources of error in the two polyphonic datasets. However, it has to be noted that we observe an improvement for the FH dataset ($\text{Est}_F = 0.89$ and $\text{Occ}_F = 0.60$), compared with the previously presented audio-based approach in [106] ($\text{Est}_F = 0.60$ and $\text{Occ}_F = 0.33$).

Table 5.3 shows the minimum value, maximum value, mean value and standard deviation of the three system parameters (τ_{min}, τ_{max} and $\sigma_{\hat{\tau}}$) as well as of the three performance f-measures (Phr_F, Est_F and Occ_F) across the five folds. It can be seen that τ_{min} and τ_{max} vary within a small range of less than 0.2 seconds and $\sigma_{\hat{\tau}}$ was estimated between 0.8 and 0.9. Furthermore, the standard deviation of all three performance measures across folds lies within the second decimal digit.

	FH	DFS	REB	ALL
Est-Pr	0.88	0.94	0.85	0.90
Est-Rec	0.90	0.92	0.86	0.90
Est-F	0.89	0.93	0.84	0.89
Occ-Pr	0.66	0.62	0.53	0.60
Occ-Rec	0.57	0.73	0.51	0.64
Occ-F	0.60	0.65	0.51	0.60

Table 5.2 Establishment and occurrence measures across genres

	min	max	mean	std
$\sigma_{\hat{\tau}}$	0.80	0.90	0.88	0.04
τ_{min}	1.82	1.95	1.86	0.05
τ_{max}	4.57	4.68	4.61	0.05
Phr_{F}	0.69	0.77	0.73	0.03
Est_F	0.86	0.92	0.89	0.03
Occ_F	0.54	0.66	0.60	0.05

Table 5.3 Estimated parameters and evaluation metrics across folds.

5.4.4 Glass ceiling analysis

We now aim to discover how errors inserted by the different system components influence the overall system performance. First, in order to assess the limitations due to phrase segmentation errors, we repeat the previous experiment but use manually annotated phrase boundaries (M-Ph) instead of automatically detected phrases (A-Ph) to compute pair-wise distances with the EMD measure. In order to allow for a direct comparison, we preserve the 5-fold cross-validation scenario, even though the system parameters, which form part of the phrase segmentation stage, are not required when manually segmented phrases are used.

Figure 5.9 shows that the improvement when using manually annotated phrases is larger for the REB dataset ($\text{Occ}_F = 0.51$ for A-Ph vs $\text{Occ}_F = 0.86$ for M-Ph) than for the DFS ($\text{Occ}_F = 0.65$ for A-Ph vs $\text{Occ}_F = 0.84$ for M-Ph) and FH ($\text{Occ}_F = 0.60$ for A-Ph vs $\text{Occ}_F = 0.75$ for M-Ph) datasets. As mentioned in Section 5.4.1, we identified through manual inspection mistakenly transcribed melodic segments during instrumental interludes as the main cause for phrase segmentation errors. This issue occurs less frequently in the FH dataset, because the vocal detection stage of the respective transcription algorithm is tailored to the guitar accompaniment present in *flamenco* music. The vocal detection task is more complex in the REB dataset, because the instrumentation varies across recordings and no suitable genre-specific method exists for this type of music.

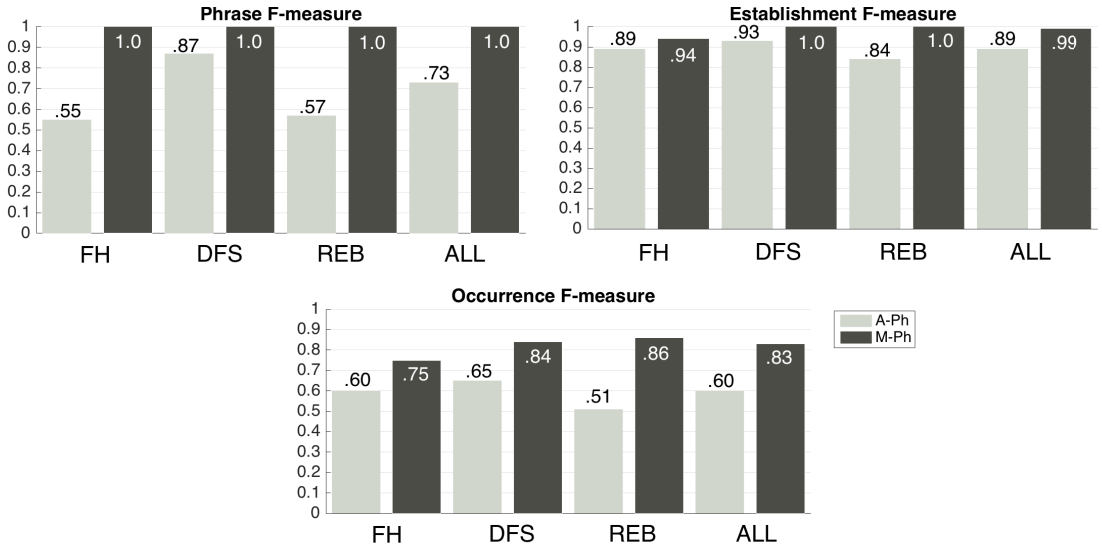


Fig. 5.9 Est_F (top) and Occ_F (bottom) for automatically segmented (light) and manually annotated (dark) phrases across datasets.

Furthermore, we observe that the establishment measures are close to 1.0 for M-Ph. These results are plausible, given that in folk music, all phrases are repeated at least once and consequently, each detected phrase forms part of a pattern. The only source of error in this scenario, with respect to pattern establishment, occurs if the clustering algorithm mistakenly creates clusters containing only a single instance. In this case, the respective phrase will not be considered part of any pattern, causing a decrease in Est_F. However, the phrase f-measure, which does not consider the clustering stage, results to 1.0 in this scenario.

Finally, we assess the influence of automatic transcription errors on the performance of the FH dataset, for which flamenco experts have manually corrected the automatic transcriptions. We repeat the previous experiment, where MT denotes the use of manually corrected transcriptions and AT refers to automatic transcriptions. The results in Table 5.4 show that when using manual transcriptions instead of automatic ones, the phrase f-measure increases from $\text{Phr}_F = 0.55$ to $\text{Phr}_F = 0.78$. However, the overall performance limitation due to errors introduced in the phrase segmentation stage appears to be stronger ($\text{Occ}_F = 0.66$ for MT-A-Ph vs $\text{Occ}_F = 0.94$ for MT-M-Ph) than due to automatic transcription errors ($\text{Occ}_F = 0.75$ for AT-M-Ph vs $\text{Occ}_F = 0.94$ for MT-M-Ph). It can be furthermore seen, that with perfect transcription and phrase segmentation, the achieved occurrence F-measure is $Occ_F = 0.94$. In this scenario, the remaining error is introduced during the melodic distance computation and clustering stages.

	Est_F	Occ_F	Phr_F
AT, A-Ph	0.89	0.60	0.55
AT, M-Ph	0.94	0.75	1.0
MT, A-Ph	0.91	0.66	0.78
MT, M-Ph	1.0	0.94	1.0

Table 5.4 Establishment and occurrence F-measure for the FH dataset for automatically (A-Ph) vs. manually annotated phrases (M-Ph) and automatic (AT) vs. manual transcriptions (MT).

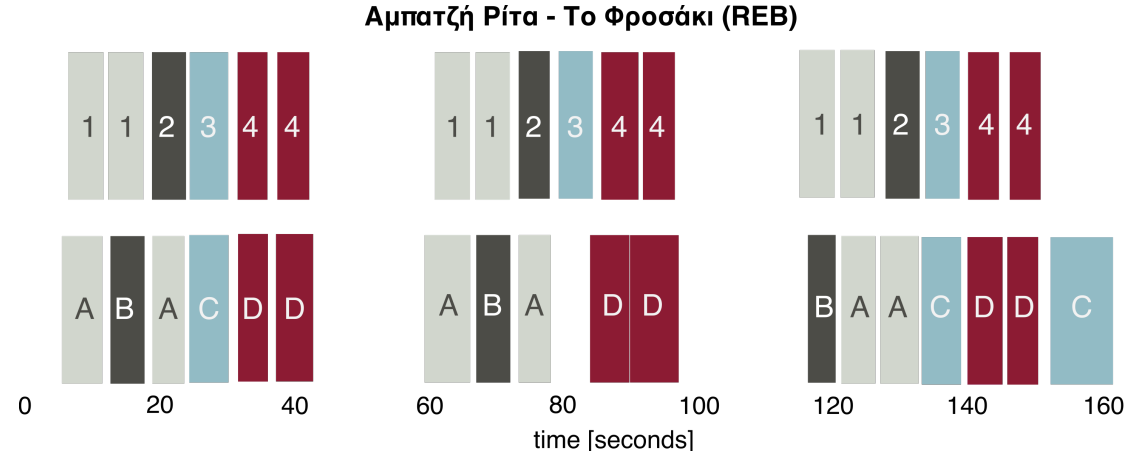


Fig. 5.10 First automatic annotation example, taken from REB: (top) ground truth and (bottom) annotated repetitions.

5.4.5 Examples and qualitative error analysis

An example of an automatic annotation of repeated phrases from a *rebetiko* recording is shown in Figure 5.10. It can be seen that the detected phrase boundaries largely correspond to the manual annotations. Ground truth patterns 3 and 4 are correctly identified as pattern a and b respectively. Ground truth patterns 1 and 2 are covered by d and c with confusion in the cluster membership. Vocal detection errors have caused a false positive phrase to be detected at the end of the recording, which was labeled as a. In this section of the song, the singer speaks, which caused the vocal detection algorithm to mistakenly classify this segment as voiced. Apart from inaccuracies caused by vocal detection errors, we observed a few frequently appearing error types which we briefly describe below.

In various examples, we observed that the segmentation stage detected sub-phrases or a group of two phrases instead of the actual phrase boundary. This corresponds to a correct segmentation on a different hierarchical level in the song structure. An example is shown Figure 5.11. In this particular example, the low intensity of the vocals with respect to the accompaniment caused several notes inside the vocal phrases to be missed by the

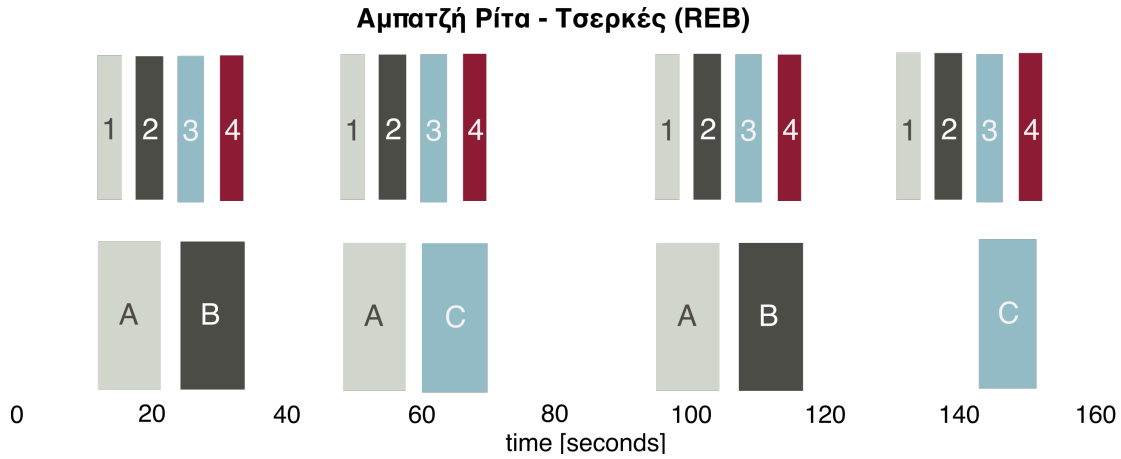


Fig. 5.11 Second automatic annotation example, taken from REB: (top) ground truth and (bottom) annotated repetitions.

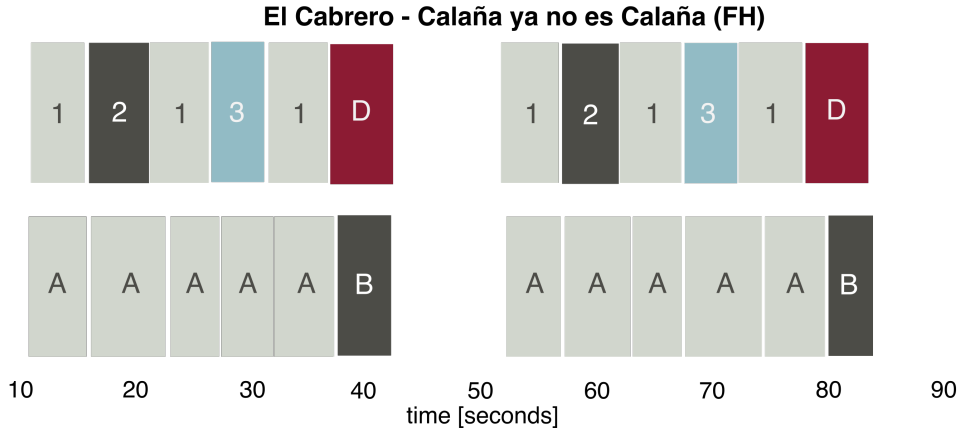


Fig. 5.12 Third automatic annotation example, taken from FH: (top) ground truth and (bottom) annotated repetitions.

transcription algorithm. Consequently, the accumulated note duration was estimated at a low value, causing a higher segmentation score when two adjacent phrases were fused. Here, it can furthermore be observed that the same vocal detection errors cause missing vocal segments (around sec. 130).

In addition, we observed some clustering errors, where two or more patterns are mistakenly grouped together. This scenario occurs in particular when one pattern is very dissimilar from a group of patterns which have at least a basic melodic movement in common. An example is shown in Figure 5.12 where ground truth patterns 1, 2 and 3 mistakenly received the common label *a*. Similarly, clustering errors occur when patterns show only minor melodic differences which are nevertheless perceptually significant due to different harmonic progressions in the accompaniment. This is the case for ground truth patterns 1 and 2 from the example shown in Figure 5.10.

5.4.6 Relation to pattern discovery and structural segmentation

In a final experiment, we investigate the suitability of state of the art audio-based pattern detection and structural segmentation algorithms for the task of detecting repeated sung phrases. To this end, we evaluate the algorithms listed below on our datasets and analyse their behaviour on three selected examples, after adapting available system parameters to the characteristics of this particular task. Furthermore, we compare the proposed system to a baseline method which operates on automatic transcriptions but performs a naive segmentation and label assignment.

- The audio-based pattern detection algorithm described in [146] (NF14). The tempo of each song, which is required by the algorithm, was estimated using the method described in [53]. All audio-files are resampled to 11025 Hz and the short-term spectrogram is extracted using a 290 ms long moving window, with an overlap of 50% between successive windows. This stage is followed by a constant-Q transform. As defined by the authors, the path score threshold parameter, θ, is set to the value of 0.3 and the number of eliminated diagonals, ρ, around a detected path in the similarity matrix is set to the value of 2. At the output of the system, instances which temporally overlap with other instances of the same pattern are discarded.

- The audio-based pattern detection method proposed in [223] (WD15). The method allows to specify a minimum pattern duration, which we set to τ_{min}. After downsampling the recordings to 11025 Hz, the short-term spectrogram is computed with a moving window, 743 ms long (11 ms hop size), followed by a constant-Q transform covering the frequency range from 27.5 Hz to 55125.5 Hz. The grid for determining the optimal VMO threshold, θ, ranges from 0.0 to 2.0 in steps of 0.01, as specified by the authors. We discard instances which overlap with other instances of the same pattern and extract the required tempo estimate with the method in [53].

- The structural segmentation approach proposed in [226] (WB10) which segments an audio recording into adjacent sections and assigns the same label to repeated sections. The extraction of the beat-synchronous chromagram follows the method and parameters in [54]. As recommended by the authors, the number of basis patterns is set to $K = 4$ and the minimum number of segments at the output is set to the value of 3.

- The method presented in [141] (MJG13), which detects repeated segments in audio recordings. After resampling all recordings to 22050Hz, the Chroma Energy Normalised Statistics are extracted with a feature rate of 2Hz. As recommended by the authors, the fitness tolerance parameter is set to $\delta = -2$ and the relative threshold to suppress small values in the self-similarity matrix is set to $\rho = 0.15$. The algorithm allows to

specify a lower and upper bound for the allowed segment duration, which we set to τ_{min} and τ_{max}, respectively.

- A naive baseline method (BL), which segments the note transcriptions at the note offsets closest to multiples of the median phrase duration (3.11 sec, computed over ALL). The resulting segments are assigned one out of k random labels, where $k = 4$ is the median number of clusters in the annotated ground truth data.

We furthermore compared the performance of the proposed method (denoted as PR), using the generic cross-fold validation setup, to our prior approach [106] (PKDM15) which was developed in the context of accompanied flamenco singing. Figure 5.13 shows the output of the different algorithms for three audio examples. The manually annotated ground truth is denoted GT. All system parameters, features and pre-processing methods not mentioned above, were applied as described in the respective publications.

As described in Section 5.1, pattern detection aims at discovering repetition in a music recording without restrictions regarding instrumentation or type of structural units. As a result, some of the repetitions detected by NF14 and WD15 are located in instrumental sections and others are located in singing sections but do not coincide with phrase start and end boundaries. Structural segmentation aims at segmenting the entire recording into labeled sections, where regions with the same labels are considered similar to each other. In the second example taken from the REB collection, the algorithm WB10 does assign a common label to two long vocal segments which are indeed instances of the same melody, but those segments span over more than two phrases. The algorithm also detects a repeated instrumental interlude between adjacent vocal segments. In the two accompanied examples taken from the FH and REB collections, the MJG13 algorithm detects repeated chord sections in the accompaniment.

In summary, we can conclude that while these four algorithms do detect repetition in the audio examples, the extracted patterns do generally not coincide with sung phrases. These findings also become apparent when analysing the establishment and occurrence f-measures for the different algorithms on all four datasets. Specifically, in Table 5.5 it can be seen that the proposed approach outperforms the baseline method for the task of detecting repeated sung phrases. In addition, among the referenced algorithms, NF14 yields the best results with $Est_F = 0.47$ and $Occ_F = 0.15$ on ALL. It can also be observed that the proposed method yields an improvement compared to PKDM, on both the flamenco dataset for which PKDM was originally proposed ($Est_F = 0.60$ and $Occ_F = 0.33$), as well as on the union of all three datasets ($Est_F = 0.42$ and $Occ_F = 0.15$). The note-based BL method yields a high establishment score ($Est_F = 0.75$) which outperforms the four referenced algorithms in this Section, but not the method we propose. This can be explained by the fact that, despite the naive segmentation and labelling process, the BL method benefits from the transcription system: since no notes are transcribed during long instrumental sections at the beginning and

	PR	PKDM15	NF14	WHD15	WB10	MJG13	BL
FH							
Est-F	**0.89**	0.60	0.47	0.25	0.33	0.22	0.73
Occ-F	**0.60**	0.33	0.11	0.02	0.03	0.05	0.16
DFS							
Est-F	**0.93**	0.38	0.50	0.42	0.17	0.15	0.75
Occ-F	**0.65**	0.13	0.16	0.07	0.01	0.00	0.16
REB							
Est-F	**0.84**	0.43	0.40	0.50	0.24	0.14	0.75
Occ-F	**0.51**	0.08	0.12	0.07	0.00	0.01	0.18
ALL							
Est-F	**0.89**	0.42	0.47	0.45	0.22	0.16	0.75
Occ-F	**0.60**	0.15	0.15	0.06	0.01	0.01	0.17

Table 5.5 Comparative evaluation of pattern discovery and structural segmentation: Establishment and occurrence f-measure.

end of the song, the BL method does not detect any patterns in those parts of the recording. It is however possible, that it detects patterns which span over interludes between vocal sections. This behaviour is apparent in the first two examples of Figure 5.13. However, it is important to note, that the random labelling scheme of the BL methods, results in a low occurrence score ($Occ_F = 0.17$).

5.5 Conclusions

We presented a complete system for detecting repeated sung phrases in folk music recordings, starting from the audio signal. A novel phrase segmentation algorithm was proposed, which operates on automatically generated note-level transcriptions. We investigated a variety of melodic distance measures to compute pair-wise distances among phrases and, by means of a standard clustering algorithm, clusters of similar phrases were discovered. Members of a cluster were interpreted as instances of the same repeated pattern.

In a detailed evaluation procedure, we assessed the performance of the proposed method on three genres with distinct musical characteristics. In a glass ceiling analysis, we demonstrated that phrase segmentation errors cause a stronger performance limitation than transcription inaccuracies. Furthermore, we provided various examples of automatic annotations and analysed frequently occurring errors. The proposed system outperforms the only existing audio-based approach to repeated phrase discovery in folk music recordings and, we have shown that state-of-the-art methods for the related tasks of audio-based pattern detection and music segmentation are not suitable for this task, despite non-exhaustive search for the

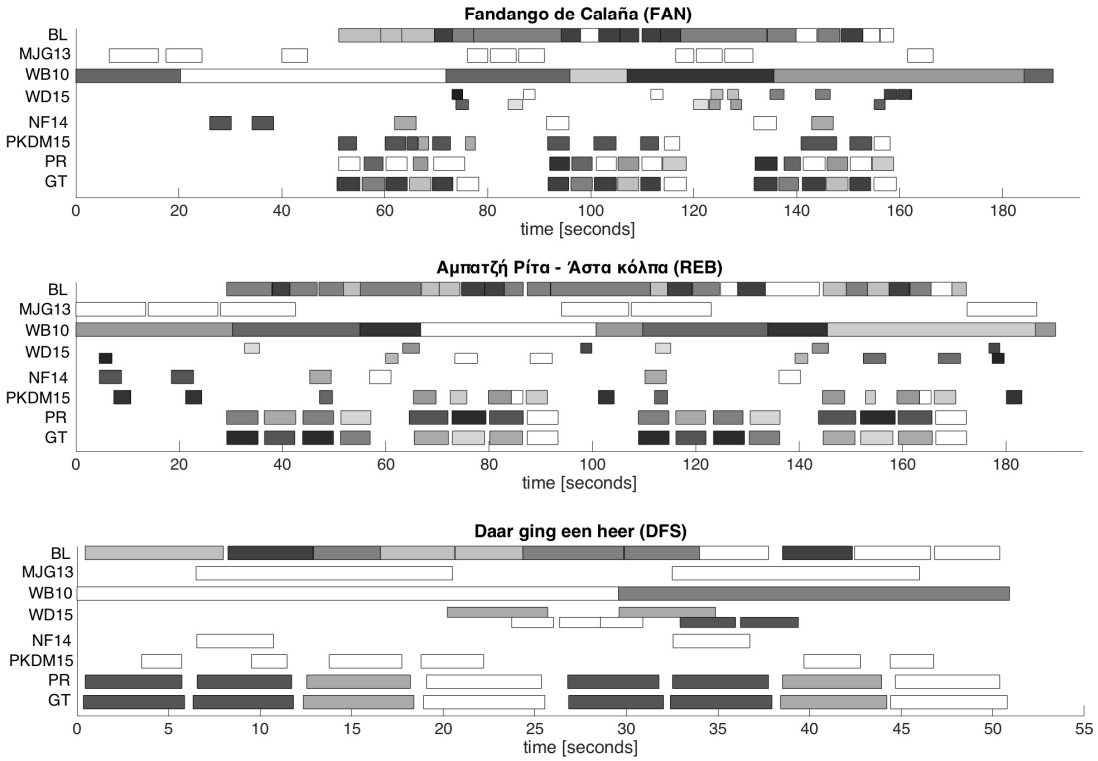


Fig. 5.13 Ground truth patterns (GT), output of the proposed algorithm (PR) and of three referenced systems (MJG13 [141], WHD15 [223], NF14 [146], WB10 [226] and PKDM15 [106]) for three audio examples. Shading indicates the cluster membership.

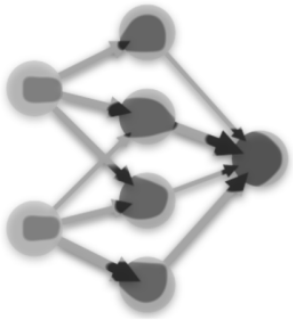
optimal settings per algorithm. Manual annotations and the source code of the system have been made available for the sake of reproducibility of research results.[5].

While the proposed method gives convincing results for the three studied datasets, there are a number of possible extensions, in particular in the context of flamenco music. Currently, the main limitation of the system is the fact that the musicological assumptions for the phrase segmentation algorithm do not generalise to all flamenco styles. In particular, many styles exhibit sung phrases which significantly vary in length. Consequently, a musicological study of phrase boundary characteristics in flamenco music and the development of a genre-tailored algorithm are logical directions for future work. In addition, the system could be employed in large-scale studies with the objective of discovering frequently occurring patterns and hidden structures or trends related to phrase-level repetition.

[5]http://ieeexplore.ieee.org/abstract/document/8100982/media

Chapter 6

Deep Learning for Flamenco Description and Discovery



6.1 Introduction

Over the last decades, **machine learning** has become an essential tool of modern technology which has made its way into an overwhelming number of applications that shape our every-day life, including product recommendation [153], object detection in images [116], speech recognition [83] and credit card fraud detection [13]. Moreover, cutting-edge artificial intelligence systems, like self-driving cars [12] and biometric authentication systems [109], heavily rely on machine learning methods.

In the context of classification problems, machine learning algorithms learn a decision surface on vector representations of a large amount of annotated examples. Such data-driven methods circumvent the need to explicitly program decision rules and can automatically discover complex relationships between the input data and the target classes. Moreover, in the context of data mining, machine learning algorithms are employed to efficiently unveil hidden trends, patterns and relationships in large amounts of data. For a comprehensive introduction to the topic, the reader is referred to [142].

A major limitation, when applying conventional machine learning to real-world data, is the need for a suitable representation of the input domain which encodes the relevant aspects of the input for a given classification problem. Handcrafting such features requires a significant amount of problem-specific engineering efforts and domain expertise. In addition, when defining a feature representation manually, one simply hopes it will hold the relevant information which enables the algorithm to distinguish between target classes, without knowing if the selected features are optimal in any sense.

In an effort to overcome these limitations, **deep learning architectures** [115] employ a cascade of multiple trainable layers, where each layer represents the input data with an increasing level of abstraction. In the context of supervised classification scenarios, the last layer either represents the target class directly, or a feature representation which can be fed into a conventional (shallow) algorithm for classification. In this way, deep learning architectures are not only capable of working on representations which are close to the input data, but are furthermore suitable to tackle computational tasks of high complexity and abstraction, as they are often encountered in computer vision and machine listening. While the mathematical foundations for deep learning have been around since the 1980s [171], recent advances in optimisation and parallel processing have made these architectures applicable to large real-world datasets at acceptable runtimes.

Figure 6.1 outlines the simplest deep learning architecture, the **multi-layer neural network** (ML-NN) or feed-forward network. The system is composed of an input layer, several (in this case two) hidden layers, and an output layer. The input layer contains a single N-dimensional input vector $x = (x_1, x_2, \ldots, x_N)$ (in this case a 3-dimensional vector) and an additional bias term, denoted with "1". Arrows depict trainable weight parameters and circles represent functions $f(z)$, which are referred to as *units*. In practice, the number of hidden layers and units in each layer, are important design choices which are usually taken on a problem basis.

Each unit maps its weighted input to a single output value, referred to as its *activation*. In particular, the activation $a_1^{(1)}$ of the first unit of the first hidden layer is computed as

$$a_1^{(1)} = f(z_1^{(1)}) = f(w_{11}^{(1)} x_1 + w_{21}^{(1)} x_2 + w_{31}^{(1)} x_3 + b_1^{(1)}), \tag{6.1}$$

where $w_{ij}^{(l)}$ denotes the weight connecting the i^{th} output of the $(l-1)^{\text{th}}$ layer (here the input layer) to the j^{th} unit of the l^{th} layer and $b_j^{(l)}$ stands for the weight connecting the bias to the j^{th} unit of the l^{th} layer. The output a_n^l refers to the n^{th} unit of the l^{th} layer. Similarly, the activation of the first unit of the second hidden layer is computed as

$$a_1^{(2)} = f(z_1^{(2)}) = f(w_{11}^{(2)} a_1^{(1)} + w_{21}^{(2)} a_2^{(1)} + w_{31}^{(2)} a_3^{(1)} + b_1^{(2)}). \tag{6.2}$$

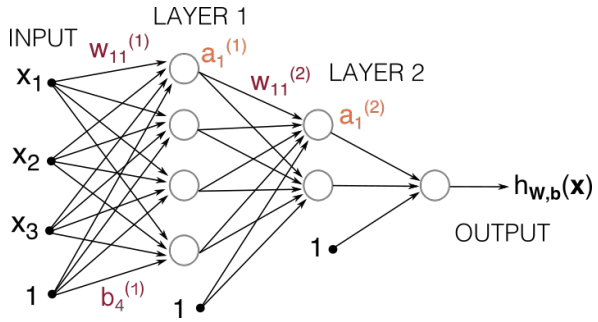


Fig. 6.1 Schematic illustration of a multi-layer neural network with two hidden layers, a 3-dimensional input layer and a single output unit. The first hidden layer contains 3 units and the second hidden layer contains 2 units.

Common choices for the activation functions $f(z)$ are the *sigmoid* and the *rectified linear function*.

Now, let all weights be contained in the matrix $\mathbf{W}$ and all bias weights in the vector $\mathbf{b}$. Similarly to the two previous equations, the output activation of the network $h_{\mathbf{W},\mathbf{b}}$ results to

$$h_{\mathbf{W},\mathbf{b}} = f(z_1^{(3)}) = f(w_{11}^{(3)} a_1^{(2)} + w_{21}^{(3)} a_2^{(2)} + b_1^{(3)}). \tag{6.3}$$

Training the network for a specific task consequently corresponds to finding the values for $\mathbf{W}$ and $\mathbf{b}$, such that the output $h_{\mathbf{W},\mathbf{b}}(x)$ computed for training samples $x^{(i)}$ best fits the corresponding annotated training labels $y^{(i)}$. For a single training instance, this objective leads to the cost function

$$J(\mathbf{W}, \mathbf{b}, x^{(i)}, y^{(i)}) = ||h_{\mathbf{W},\mathbf{b}}(x^{(i)}) - y^{(i)}||^2 \tag{6.4}$$

and for a training set of M training samples

$$\mathcal{D}_{\text{train}} = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(M)}, y^{(M)})\},$$

the cost function results to

$$J(\mathbf{W}, \mathbf{b}, \mathcal{D}_{\text{train}}) = \frac{1}{M} \sum_{i=1}^{M} J(\mathbf{W}, \mathbf{b}, x^{(i)}, y^{(i)}) \tag{6.5}$$

which corresponds to the mean squared error over all training instances. In practice, this cost function can be extended with a regularisation term which penalises large weight values in order to avoid numerical overflow and to furthermore prevent overfitting.

The optimisation problem is commonly solved with variants of the **stochastic gradient descent** method [14], where the necessary partial derivatives with respect to the weights are

computed using the **backpropagation** [171] algorithm. This process is described in further detail in the next section.

In the context of audio and music processing, deep learning based methods have shown to give promising results, and, in many cases, have been able to outperform state-of-the-art methods. In recent years, several MIR tasks have been approached from a deep learning perspective, including onset detection [179], instrument classification [79] music recommendation [209] and genre recognition [26].

This chapter focuses on a particular deep learning architecture, the **Convolutional Neural Networks** (CNNs) and their application to two flamenco-related tasks, singer identification and structural segmentation. The CNN architecture is reviewed in Section 6.2 and its application in a system for image-based singer identification in flamenco videos is presented in Section 6.3. In Section 6.4, a CNN-based system is proposed, which segments a flamenco recording into sections of consistent instrumentation. The resulting structural annotations are explored in a large-scale corpus analysis in Section 6.5, which gives rise a number of interesting applications and musicological observations.

6.2 Convolutional Neural Networks

Convolutional architectures extend the aforementioned feed-forward network with two additional types of layers, **convolutional** and **pooling** layers, at the beginning of the processing chain. Originally inspired by image processing, the goal of employing convolutional layers is to capture two-dimensional patterns in pixel arrays, such as edges or regions of homogenous colour values. While ML-NNs model an input image as a one-dimensional input vector, discarding information on the horizontal adjacency of pixels, CNNs take two-dimensional arrays as input and employ two-dimensional filters, or masks, to capture local phenomena which span over both dimensions. Over the past years, CNNs have been successfully employed to various image processing tasks, including face detection [119, 65] and object recognition [98].

More recently, the advantages of CNNs have been explored in the context of several MIR problems, including boundary detection [207], music recommendation [209] and instrument recognition [79]. In the case of music signals, the input of the network is a two-dimensional array, which combines several temporally adjacent instantaneous spectral feature vectors into a single feature matrix. In this way, CNNs can capture and learn short-term temporal patterns in the frequency spectrum and partially preserve the temporal signal evolution. It is worth noting, that while CNNs are assumed to be capable of working on raw or near-raw data, in the context of music signals, a spectral representation is commonly chosen as input. So called end-to-end learning, meaning the waveform is fed directly as input into

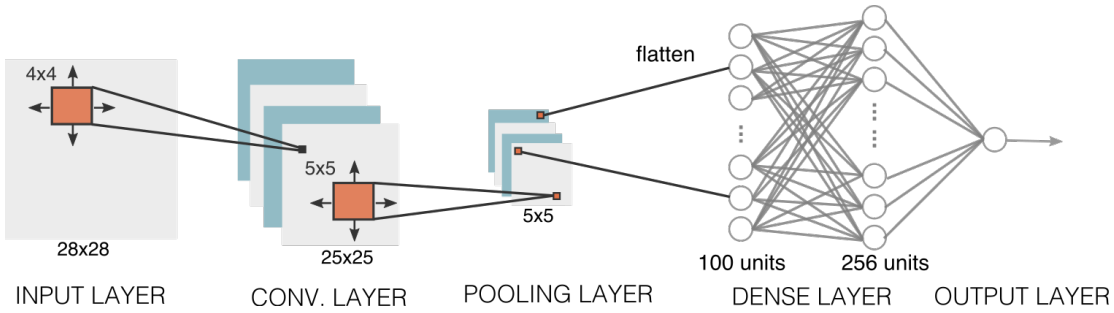


Fig. 6.2 Schematic illustration of a CNN with a convolutional layer containing four 4×4 filters, a 5×5 pooling layer, a dense layer with 256 units and a single output unit.

the network, have shown to yield inferior performance compared to methods operating on spectral representations [46].

6.2.1 Architecture

An example of a simple CNN architecture is shown in Figure 6.2. The input of the network is a 28×28 feature matrix, which could, for example, represent the pixels of a small image. The input layer is followed by a convolutional layer containing four convolutional filters. Each filter of size 4×4 maps the image to a 25×25 feature map. The filter can be thought of as a sliding mask which moves across the image and produces a single pixel for each position. Each cell of the filter is associated with a trainable weight and the produced pixel in the feature map corresponds to the weighted sum of the input pixels covered by the filter. Formally, the filter map $(I \otimes K)$ is computed from an image I and a filter K of size $k_1 \times k_2$ as follows:

$$(I \otimes K)_{i,j} = \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} I_{i+m,j+n} \cdot K_{m+1,n+1} \tag{6.6}$$

Figure 6.3 (a) shows an example of such a convolutional filter operation. Note, that strictly speaking, this process is not a two-dimensional convolution, but a cross-correlation, since the convolution would imply rotating the mask by 180 degrees. Similar to the units of the ML-NN, a bias is added to each value of the filter map and the result is passed through an activation function. It should be mentioned, that in convolutional layers, several input elements are connected to the same weight. This concept, also referred to as *weight sharing*, results in a lower amount of trainable parameters compared to the densely connected layers in ML-NNs.

The second building block shown in Figure 6.2 is a pooling layer. Similar to the concept of downsampling, pooling filters move across their input and reduce its size by performing local aggregation operations on adjacent matrix regions. Common pooling schemes compute the maximum value *(max pooling)* or the average *(mean pooling)* over the area covered by

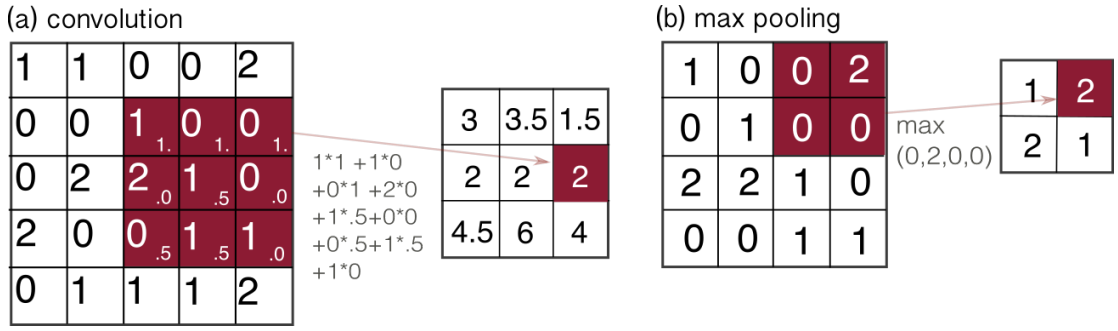


Fig. 6.3 (a) Example of a convolutional filter operation: A 5×5 feature matrix is mapped to 3×3 feature map using a 3×3 filter with stride $= 1$; (b) Example of a pooling operation with a 2×2 max pooling filter with stride $= 2$.

the filter. An example of a max pooling operation is shown in Figure 6.3 (b). In practice, several combinations of convolutional and pooling layers with variable numbers of filters can be cascaded.

For both, pooling and convolutional filters, hyper-parameters are the filter size and the stride with which the filter moves across the input matrix. In most cases, this input processing stage is followed by a number of densely connected layers, forming an ML-NN architecture as described above.

6.2.2 Training

As mentioned earlier, training a network for a specific problem corresponds to finding the optimal weights for all trainable parameters with respect to a cost function. Starting from a random parameter setting and using a standard gradient descend [81] approach, the weights in $\mathbf{W}$ and bias weights in $\mathbf{b}$ are updated in each iteration as follows:

$$w_{ij}^{(l)} := w_{ij}^{(l)} - \alpha \frac{\partial J(\mathbf{W}, \mathbf{b}, \mathcal{D}_{\text{train}})}{\partial w_{ij}^{(l)}} \tag{6.7}$$

$$b_{i}^{(l)} := b_{i}^{(l)} - \alpha \frac{\partial J(\mathbf{W}, \mathbf{b}, \mathcal{D}_{\text{train}})}{\partial b_{i}^{(l)}} \tag{6.8}$$

where the step size α is referred to as the *learning rate*. The partial derivatives over the entire training set $J(\mathbf{W}, \mathbf{b}, \mathcal{X}_{\text{train}})$ can be computed by averaging over gradients at individual samples:

$$\frac{\partial J(\mathbf{W}, \mathbf{b}, \mathcal{D}_{\text{train}})}{\partial w_{ij}^{(l)}} = \frac{1}{M} \sum_{i=1}^{M} \frac{\partial J(\mathbf{W}, \mathbf{b}, \mathbf{x}^{(i)}, y)}{\partial w_{ij}^{(l)}} \tag{6.9}$$

$$\frac{\partial J(\mathbf{W}, \mathbf{b}, \mathcal{D}_{\text{train}})}{\partial b_{i}^{(l)}} = \frac{1}{M} \sum_{i=1}^{M} \frac{\partial J(\mathbf{W}, \mathbf{b}, \mathbf{x}^{(i)}, y)}{\partial b_{i}^{(l)}} \tag{6.10}$$

The partial derivatives of $\mathbf{W}$ and $\mathbf{b}$ parameters are computed with the back-propagation [171] algorithm, which can be interpreted as an application of the chain rule of derivatives to neural networks. Let us assume, a forward pass has been computed for the training example $(x^{(i)}, y^{(i)})$, and the corresponding network output $h_{\mathbf{W},\mathbf{b}}(x^{(i)})$ and all activations throughout the network are known. First, without loss of validity, equation 6.11 can be rewritten as

$$J(\mathbf{W}, \mathbf{b}, x^{(i)}, y^{(i)}) = \frac{1}{2}||h_{\mathbf{W},\mathbf{b}}(x^{(i)}) - y^{(i)}||^2. \tag{6.11}$$

Note, that the added constant term does not affect the optimisation objective. We now define the *error signal* $\delta_n^{(l)}$ for the n^{th} unit of the l^{th} layer as the derivative of the cost function with respect to its input:

$$\delta_n^{(l)} = \frac{\partial J(\mathbf{W}, \mathbf{b}, x^{(i)}, y^{(i)})}{\partial z_n^{(l)}}. \tag{6.12}$$

The error signal measures the contribution of each unit to the overall error. For the units in the output of the network (layer L), the error signal can be directly computed as

$$\delta_n^{(L)} = \frac{\partial J(\mathbf{W}, \mathbf{b}, x^{(i)}, y^{(i)})}{\partial h_{\mathbf{W},\mathbf{b}}} \cdot \frac{\partial h_{\mathbf{W},\mathbf{b}}}{\partial z} = -(y - h_{\mathbf{W},\mathbf{b}}) \cdot f'(z_n^{(L)}). \tag{6.13}$$

The idea is now to propagate this result backwards through the dense layers of the network. The error signals of each unit is calculated as

$$\delta_i^{(l)} = \left(\sum_{j=1}^{s_{l+1}} w_{ij}^{(l+1)} \delta_j^{(l+1)} \right) \cdot f'(z_i^{(l)}) \tag{6.14}$$

where s_{l+1} denotes the number of units in the $(l+1)^{\text{th}}$ layer. In other words, $\delta_i^{(l)}$ receives weighted contributions from all the errors of units in the next layer to the right based according to weight parameters.

In order to propagate the error signals through pooling filters, we need to perform the aggregation function g of the respective filter on all outgoing error signals. The error matrix after back-propagation through the k^{th} filter results to

$$\delta_k^{(l)} = g(w_{ik}^{(l+1)} \delta_j^{(l+1)}; j = 1 \dots s_{l+1}) \cdot f'(z_k^{(l)}) \tag{6.15}$$

In case of max pooling, g passes the error only to the inputs which had the maximum value in the forward pass. For mean pooling, the error is distributed uniformly among all inputs.

Similarly, we can propagate the error signals through convolutional layers. Here, we exploit the commutative property of the convolutional filter operation and the error matrix after back-propagation through the k^{th} filter results to

$$\delta_k^{(l)} = \delta_k^{l+1} \otimes W_k \cdot f'(z_k^{(l)}), \tag{6.16}$$

where W_k is the weight matrix, or kernel, of the filter.

Finally, the partial derivatives with respect to the weight and bias parameters can be calculated using the error signals:

$$\frac{\partial J(\mathbf{W}, \mathbf{b}, x^{(i)}, y^{(i)})}{\partial w_{ij}^{(l+1)}} = a_j^{(l)} \delta_i^{(l+1)} \tag{6.17}$$

$$\frac{\partial J(\mathbf{W}, \mathbf{b}, x^{(i)}, y^{(i)})}{\partial b_i^{(l+1)}} = \delta_i^{(l+1)} \tag{6.18}$$

6.2.3 CNNs in practice

There exist several machine learning libraries, which implement CNNs among other common deep learning architectures, the most popular being *torch*[1], *keras*[2] (which builds upon *tensorflow*[3] and *theano*[4]) and *caffe*[5]. There are several common extensions and modifications to the fundamental architecture and training process described above, which result in a number of design choices to be taken in practice.

Activation functions

Given its differentiability and fixed output range, which can be directly interpreted as a class probability, the *sigmoid* function

$$f(z) = \frac{1}{1 + e^{-z}} \tag{6.19}$$

is commonly used for the output nodes of binary classification problems. In multi-class problems, the *softmax* function

$$f(z_j) = \frac{e^{z_j}}{\sum_{c=1}^{C} e^{z_c}} \tag{6.20}$$

is used, which ensures that the outputs of all C units in the output layer sum to 1. In hidden layers, the use of rectified linear units (*relu*)

$$f(z) = \max(z, 0) \tag{6.21}$$

has shown to significantly speed up the training process of large networks [98].

[1] http://torch.ch
[2] https://keras.io
[3] https://www.tensorflow.org
[4] http://deeplearning.net/software/theano/
[5] http://caffe.berkeleyvision.org

Loss functions

In binary or multi-class classification problems, the *cross-entropy loss* is frequently used instead of the mean squared error described in equation 6.5. Formulating the desired output $y^{(i)}$ as a categorical variable, where $y_j^{(i)} = 1$ if the j is the target class, the cross entropy loss is defined as

$$J(\mathbf{W}, \mathbf{b}, \mathcal{D}_{\text{train}}) = -\sum_{i=1}^{M}\sum_{c=1}^{C}[y_c^{(i)} h_{\mathbf{W},\mathbf{b},x^{(i)},c} + (1 - y_c^{(i)}) h_{\mathbf{W},\mathbf{b},x^{(i)},c}]. \tag{6.22}$$

The cross-entropy loss assigns a higher penalty to incorrect class predictions with a high output confidence than to false predictions with a lower confidence.

Optimisation

In order to avoid a full pass over the entire training set per update step, stochastic gradient descent [14] methods are commonly used, which update parameters based on single instances or small data batches. Recent variants, such as *Adagrad* [51], *Adadelta* [233] or *Adam* [96], furthermore provide adaptive learning rate schemes in order to avoid slow convergence or even loss of convergence in flat regions or around saddle points of the error function. In [34], it has been demonstrated that, given the high dimensionality of the optimisation problem, these aforementioned phenomena pose a stronger limitation to gradient-based learning methods than the existence of local minima in the cost function.

A network is commonly trained for a pre-defined number of passes through the full dataset (which can be broken into batches as described previously), referred to as *epochs*. Additionally, early stopping criteria can be employed, which end the iteration process if either the value of the cost function, or of the classification accuracy on the training set, does not improve by more than a specified amount within a certain number of epochs.

6.3 Image-Based Singer Identification in Flamenco Videos

After having reviewed the basic architecture and training process of CNNs, we present in this section a CNN-based application which can automatically identify the singer in a flamenco performance video among a set of candidates. The core of the proposed framework is a pre-trained CNN, which, instead of solving the classification problem directly, was trained to extract a feature vector with high discriminative power among faces, a so-called *embedding*, from an image. This generic architecture has the advantage that it can be employed in face recognition tasks without explicitly training it on a specific set of candidates. Instead, a shallow classifier can be trained on the extracted embedding, which is computationally

significantly more efficient than training the CNN itself. However, a series of standard, as well as problem-specific, image processing techniques are required to convert raw images, taken from the annotated image dataset and the unlabelled videos, into a suitable input for the CNN. In the sequel, the problem statement and all processing stages of the framework are described in further detail.

6.3.1 Singer Identification

The technologically challenging task of automatic singer identification is of crucial importance for the automatic indexing of large music databases. For the particular case of flamenco music there are several frequently occurring scenarios where the singer in a performance is unknown:

- In performance videos starring renown dancers, singers are usually considered in an accompanying role and in many cases only the name of the dancer is annotated. However, many respected singers spent the early years of their career accompanying dancers and discovering such videos could be beneficial for studying the evolution of a singer over time.
- Names are often not unique identifiers in the flamenco world. Singers may be referred to by both their stage name as well as their actual name and related singers, i.e. father and son, may be referred to by the same name
- More and more flamenco videos are submitted to popular multi-purpose video sharing platforms. However, such platforms do often not require to annotate the artist performing in the video. As a result, many flamenco videos are labeled by genre or style only.

Related work on automatic singer identification has so far been limited to the analysis of audio recordings and most approaches have formulated the task as a standard multi-class classification problem: A set of manually defined descriptors are extracted from annotated audio recordings and a shallow machine learning model is trained, which distinguishes among singers. In order to classify an unknown recording, the same features are extracted from the audio file and evaluated against the pre-trained model. [20] used a combination of Mel-frequency Cepstral Coefficient (MFCCs), Liner Prediction Mel-frequency Cepstral Coefficient (LPMCCs) and Gammatone Cepstral Coefficient (GTCCs) to train a Gaussian Mixture Model for each target class. Similarly, [202] trained GMMs based on a combination of MFCCs extracted from spoken and sung data of the candidate singers. Also using GMMs trained on MFCCs, [110] apply singing voice enhancement in a preprocessing stage and propose a novel method to propagate the uncertainty through the MFCC computation. In [183], multiple low-level descriptors are extracted from vocal and non-vocal segments to train four GMMs

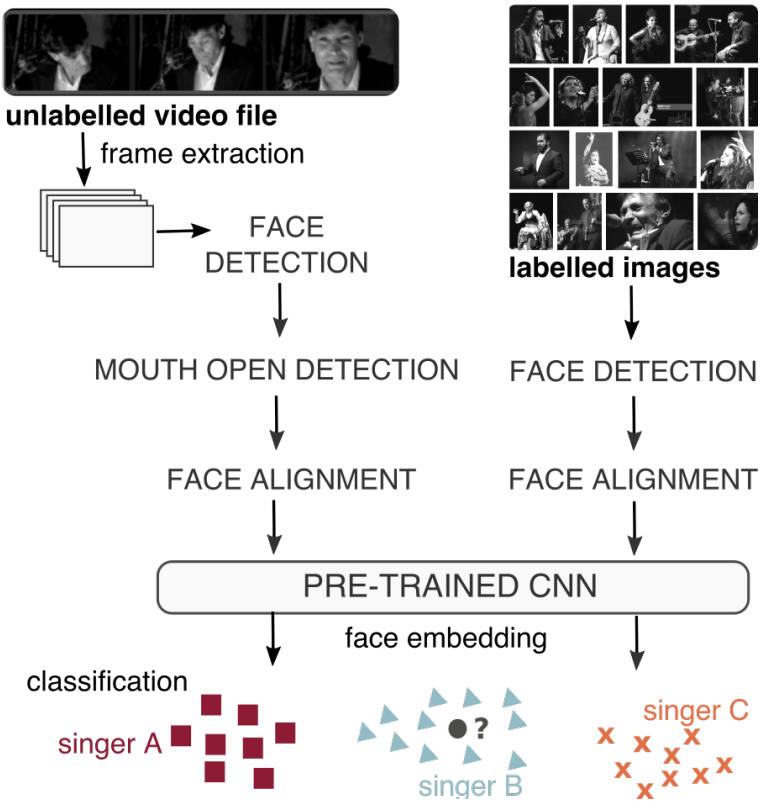


Fig. 6.4 Overview of the processing pipeline.

per singer, each focusing on a different performance aspect. In [235], the feature extraction of an unlabelled song is restricted to the beginning of the first singing voice segment. [62] fuse commonly used low-level timbre descriptors with fundamental frequency trajectories and [148] show that the additional vibrato-related features can increase performance. Methods for non-Western music traditions addressing genre-specific properties and challenges have been developed for carnatic [191], rebetiko [85] and flamenco music [103].

Analysing the aforementioned approaches reveals that they mainly differ in the use of descriptors and new features tend to yield only incremental improvement. In particular, spectral distortions in low audio quality audio recordings and the presence of dominant accompaniment instruments have shown to limit the performance of audio-based approaches.

6.3.2 Method

Motivated by the growing amounts of digitally available audio-visual performance recordings and the typically encountered scenarios described above, we present an image-based approach to singer identification in flamenco videos using state of the art face recognition technologies. The term face recognition refers to the task of automatically identifying a person based on a frontal image of his or her face. Given their non-intrusive nature and low-cost hardware

requirements, face recognition methods are an essential tool for biometric-based person authenticaction [133] and video surveillance [17], and have furthermore found application in multimedia indexing and video thumbnailing. For a compete review we refer the reader to [89].

The task of singer identification in music performance videos encompasses three major challenges:

1. Detect faces in the unlabelled video (face detection).
2. Decide if a face belongs to the singer or to one of the other musicians on stage.
3. Determine the singer's identity among a number of candidates in an annotated image database (face recognition).

The first and the third are well-studied image processing tasks and we employ state-of-the-art methods which have shown to give reliable performance in real-world scenarios. The second task is specific to the problem at hand and we propose a simple method to estimate if a detected face belongs to the singer or not.

An overview of the proposed method is depicted in Figure 6.4. Given a dataset of annotated images of candidate singers, we first employ a state of the art face detection algorithm to detect the face bounding box in each image. Then, all faces are aligned to a canonical pose using estimated landmarks and a series of affine image transformations. Each instance is then fed into a pre-trained CNN which extracts a feature vector from the image. To identify the singer of an unlabelled video file, we first detect all face bounding boxes in each frame. In order to decide if a detected face corresponds to the singer, we extract facial landmarks and estimate the amount of mouth opening. If the mouth is estimated to be open, we assume that the face inside the bounding box belongs to the singer and proceed as in the training stage: We align the face image to the canonical pose and extract its embedding using the pre-trained CNN. Then, we can predict the singer in the current image by comparing its embedding to all embeddings extracted from the annotated database in a k-NN classification. Finally, in order to assign a label to the video file, we perform a weighted voting scheme over all frame-wise estimates and their confidence values. Below, all processing stages are described in detail.

Face detection

Face detection is a well-studied problem in the computer vision community [234] and a a number of algorithms and pre-trained classifiers are readily available in open source image processing libraries. Here, we apply the histogram of oriented gradients (HOG) method, which was originally introduced by [32] in the context of human detection in video streams and has shown to give reliable performance for several object detection tasks, including face detection [23].

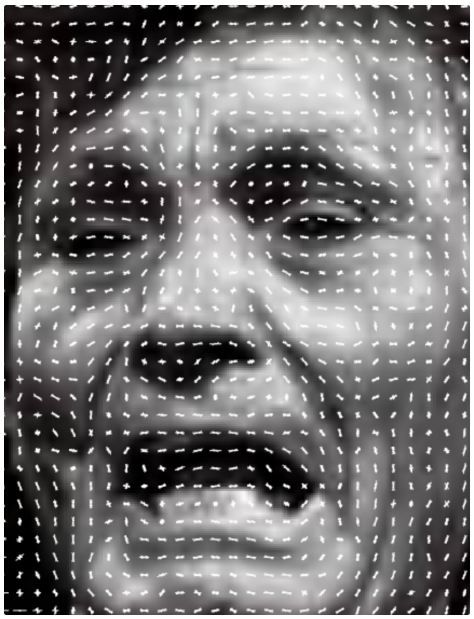


Fig. 6.5 Image with overlay of its HOG representation. The lengths of directed white lines are proportional to the respective angle histogram magnitudes.

The HOG representation of an image is generated by computing the brightness gradient for each pixel and then computing gradient histograms in cells of 16×16 pixels. Given the brightness value of a pixel p_{ij}, its horizontal brightness gradient can be computed as

$$g_x(p_{ij}) = -p_{i,j-1} + p_{i,j+1} \tag{6.23}$$

and similarly, its vertical brightness gradient as

$$g_y(p_{ij}) = -p_{i-1,j} + p_{i+1,j}. \tag{6.24}$$

The magnitude of the brightness gradient then results to

$$g = \sqrt{g_x^2 + g_y^2} \tag{6.25}$$

and its direction to

$$\theta = \arctan\frac{g_y}{g_x}. \tag{6.26}$$

Then, a gradient histogram is computed for each 16×16 cell, where the bins correspond to angle ranges and magnitudes correspond to the sum of gradient magnitudes falling into the respective bins. Commonly, an angle resolution of 20 degrees is used, which results in 9 features per cell. An example of a HOG representation of a facial image is shown in Figure 6.5.

Using a sliding window with multiple scales for each position, the local HOG representation can then be evaluated against a pre-trained model. This model solves a binary classification problem, where the target classes are *face* or *no face*. When a window setting is classified a face, its outline is returned as the corresponding face bounding box. Furthermore, duplicate detections in form of overlapping bounding boxes, are removed in a post-processing stage. Here, we used the implementation available in the *dlib* library [95] together with a linear support vector machine (SVM) classifier, which was trained on the *labeled faces in the wild* [114] dataset.

Landmark estimation and alignment

A bottleneck of face recognition systems is their lack of robustness towards pose variance. Therefore, a standard technique to improve performance is to transform input images to a reference pose. More specifically, the image is translated, scaled and rotated in such a way that specific landmarks, here the outer edges of the eyes and the nose, are located at a reference position.

Here, we apply the method proposed in [93] to estimate a set $\mathbf{S} = \{s_1, s_2 \dots s_{68}\}$ of 68 facial landmarks in each detected face bounding box. A landmark s_i is defined as the pair of x-y-coordinates of a specific facial feature, for example the the centre of the lower outline of the upper lip. The method models the task as a cascaded regression problem, where each regressor r_t predicts an update of a shape estimate $\hat{\mathbf{S}}$

$$\hat{\mathbf{S}}^{t+1} = \hat{\mathbf{S}}^t + r_t(I, \hat{\mathbf{S}}^t) \tag{6.27}$$

based on features extracted from the original image I which are indexed in relation to the current estimate $\hat{\mathbf{S}}^t$. For a complete technical description, we refer to [93]. Based on the estimated facial landmarks and the pre-defined landmarks of the canonical pose, the image can be transformed by applying standard scaling, translation and rotation operations.

A pre-trained ensemble of regression trees for facial landmark estimation and a routing performing the standard affine transformations are available in the *dlib* library [95]. An example of face detection, landmark estimation and alignment is shown in Figure 6.6.

Mouth open detection

In flamenco videos, we typically encounter, apart from the singer, various musicians on stage, including guitarists, dancers and percussionists. Consequently, it is necessary to decide if a detected face belongs to the singer. Here, we assume that detected faces with a wide mouth opening are most likely frontal shots of the singer.

To this end, we use the already estimated facial landmarks to compute a relative value for the amount of mouth opening. More specifically, compute the relative distance d between

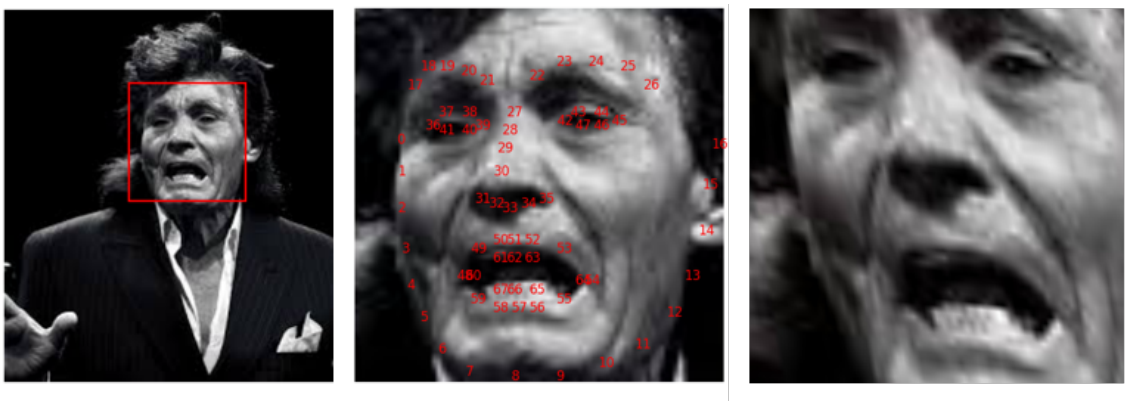


Fig. 6.6 (a) original image with face bounding box; (b) cropped image and face landmarks; (c) cropped and aligned image.

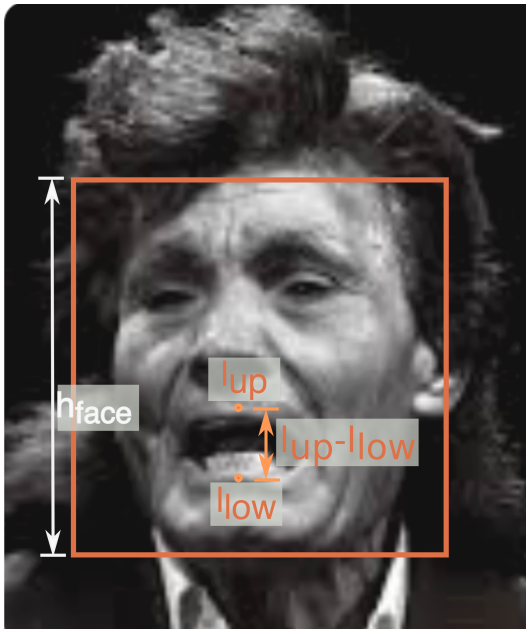


Fig. 6.7 Example of an estimated mouth opening of $d = 0.2$

the estimated facial landmarks corresponding to the centre of upper and lower lip, l_{up} and l_{low} respectively, with respect to the height of the face bounding box h_{face}:

$$d = \frac{l_{up} - l_{low}}{h_{face}} \tag{6.28}$$

An example for $d = 0.2$ is shown in Figure 6.7. We experimentally examined the value of d during various singing passages and determined $d > 0.15$ as a hard threshold for detecting a mouth to be opened.

Embedding

Once facial images have been extracted from video frames and pre-processed as described previously, the task is now to identify the individual in the image among a set of candidates. Having outlined the advantages of CNNs for image analysis, a straight-forward approach would be to train a CNN for classification among candidates in a database. However, in order to obtain a robust model, large amounts of annotated data a required, ideally in the range of various thousands of instances, or more, per class. Collecting and annotating such large training sets for each particular task requires an immense amount of manual effort, which is not feasible in most real-world applications. Furthermore, in order to extend the model with additional classes, a complete re-training of the network is necessary. It should also be mentioned, that training CNNs on large image collections usually requires long processing times. The network in [181], was trained for more than $1,000$ hours on a dataset holding more than 100 million instances.

In order to avoid problem-specific training, recent efforts in face recognition and verification have focused on developing generic models, which can be readily applied to a variety of problem-specific datasets. Instead of treating the face recognition task as a multi-class classification problem, such networks learn to extract a feature representation from a given input image, which has high discriminative among facial images of different people. Here, we employ *faceNet* [181], a CNN, which maps input images to 128-dimensional feature vectors. In the euclidean space, distances between these vectors directly correspond to facial similarity and images of the same person are assumed to be located close to each other and distant from images of other individuals.

This is achieved by forming triplets $\{x^a_{(i)}, x^p_{(i)}, x^n_{(i)}\}$, where $x^a_{(i)}$ denotes the n^{th} training image *(anchor)*, $x^p_{(i)}$ is an image of the same person *(positive)*, and $x^n_{(i)}$ is an image of a different person *(negative)*. The objective is to obtain a vector representation at the output of the network $h_{\mathbf{W},\mathbf{b}}(x^{(i)})$, referred to as *embedding*, such that

$$||h_{\mathbf{W},\mathbf{b}}(x^{(i)}_a) - h_{\mathbf{W},\mathbf{b}}(x^{(i)}_p)||^2_2 + \alpha < ||h_{\mathbf{W},\mathbf{b}}(x^{(i)}_a) - h_{\mathbf{W},\mathbf{b}}(x^{(i)}_n)||^2_2, \tag{6.29}$$

$$\forall (x^{(i)}_a, x^{(i)}_p, x^{(i)}_n) \in \mathcal{D}_{\text{train}},$$

where α is a margin threshold parameter, which all triplets should satisfy. This learning objective translates to the cost function

$$J(\mathbf{W}, \mathbf{b}, \mathcal{D}_{\text{train}}) = \sum_{i=1}^{M} \left(||h_{\mathbf{W},\mathbf{b}}(x^{(i)}_a) - h_{\mathbf{W},\mathbf{b}}(x^{(i)}_p)||^2_2 - ||h_{\mathbf{W},\mathbf{b}}(x^{(i)}_a) - h_{\mathbf{W},\mathbf{b}}(x^{(i)}_n)||^2_2 \right), \tag{6.30}$$

subject to

$$||h_{\mathbf{W},\mathbf{b}}(x^{(i)}_a)||^2_2 = 1 \tag{6.31}$$

The additional constraint restricts the embedding to the hypersphere, in order to avoid numerical overflow due to very large embedding coordinates.

In practice, in order to reduce the computational complexity, not all possible triplets are passed through the network in each epoch. Since instances which already satisfy Eq. 6.29 do not contribute significantly to the parameter optimisation, only triplets which yield a particularly poor result with respect to Eq. 6.29 are selected.

This cost function is optimised with a CNN similar to the *GoogLeNet* [192] architecture, containing, apart from a convolutional and pooling layer at the input, several cascaded *inception* layers. These modules perform several convolution and pooling operations with different kernel sizes in parallel and concatenate the output [27]. In the proposed framework, we employ the publicly available *openFace* implementation [4] of *faceNet*. The included model was trained on the union of the faceSCRUB [145] and CASIA webFace [232] datasets, holding a total of approximately 500k images.

Classification

Given a raw video file, we extract image frames in intervals of one second and generate the embedding of faces for which the mouth was estimated to be open. We evaluate each of these feature sets against the embeddings of a labeled database in a weighted k-nearest neighbour (k-NN) classification scheme [155]. We chose the rather simple k-NN method due to the sparsity of the less than 500 data-points in the 128-dimensional feature space and the fact that extracted embeddings lie on an Euclidean space where distances are directly proportional to face similarity.

For a given detected open-mouth face q_i we initialise the confidence vector $c(q_i) = [c_{q_i=1}, c_{q_i=2}, ..., c_{q_i=C}]$ with zeros, where C denotes the number of ground truth classes. We add the the value $1/k$ to the element corresponding to the annotated class of the k^{th} neighbour.

Let $Q = \{q_1, q_2, ..., q_U\}$ be the set of U detected open-mouth faces and $c(q_i)$ holds the confidence values $c_{q_i=j}$ of frame q_i belonging to class j. The accumulated confidence $\mathbf{c}_j$ for class j results to

$$\mathbf{c}_j = \sum_{i=1}^{U} c_{q_i=j} \tag{6.32}$$

and the label l is finally assigned as

$$l = \underset{j}{\operatorname{argmax}}\ \mathbf{c}_j. \tag{6.33}$$

6.3.3 Datasets

In this study, we use three datasets to train, evaluate and benchmark the proposed framework. We use a collection of annotated images from the candidate singers which are used as a reference in the k-NN classification scheme. The framework performance is assessed based on

a collection of flamenco videos. Finally, we gathered an additional dataset of audio recordings to compare the proposed approach to an audio-based baseline method.

Annotated image collection

In the scope of this study we gathered training dataset containing images of flamenco singers. For 10 singers, 3 female and 7 male, we gathered 50 publicly available images each. Images in which no face was detected were discarded, leaving a total of 478 images in the training set.

Video collection

We gathered a total of 30 videos, 3 videos of each singer in the training database. All videos were taken from online video sharing platforms and apart from the singer at least one more person is seen on stage. The quality ranges from amateur mobile recordings to professional video clip and live performance recording productions. The contained material includes live concerts, private gatherings, excerpts taken from documentaries and music clips.

Baseline audio collection

In order to compare our approach to state of the art audio-based singer identification methods (Section 6.3.4), we gathered an additional 10 audio tracks for each singer. The recordings were taken partly from the CorpusCOFLA [102] database and partly from private collections.

6.3.4 Experimental evaluation

Baseline method

State of the art audio-based singer identification methods, i.e. [235] and [202], follow a common processing framework: A machine learning model is trained on audio descriptors extracted frame-wise from an annotated database. For each frame in the unlabelled audio recording, the same features are extracted and evaluated against the learned model. Finally, the label is assigned based on a majority vote among the frame classifications.

Here, we implemented a baseline approach following this framework. From the annotated recordings in the audio training database, we first extract singing voice segments using an unsupervised method proposed by [158], which has given reliable results for flamenco recordings. From these segments we then extract the MFCCs in non-overlapping windows of 50ms length.

As in [235], we train a GMM for each singer and investigate different values for the number of components β. In the test stage, we extract the same features from the audio track of each unlabelled video, evaluate against the pre-trained GMMs and assign a label based on a frame-wise majority vote.

Results

The results of the experimental evaluation by means of correctly classified instances are shown in Table 6.1. The audio-based baseline method achieves 73.3% classification accuracy among the 10 candidates in the dataset. This is in line with the results reported in [103] where 86.7% were achieved among 5 candidates. The proposed image-based approach achieves a significantly higher accuracy of 90% for all investigated values for k.

classifier	**accuracy**
baseline, $\beta = 2$	66.7%
baseline, $\beta = 4$	73.3%
baseline, $\beta = 8$	70.0%
proposed, $k = 1$	90.0%
proposed, $k = 3$	90.0%
proposed, $k = 5$	90.0%

Table 6.1 Experimental results for audio- and video-based singer identification.

6.3.5 Discussion and future work

While the presented framework has shown potential for the task of automatic singer identification, we identify several possible extensions which can be the subject of future work. The hard threshold for the mouth opening detection, which has been determined empirically, can be replaced with a machine-learning based approach, possibly directly extending the face detection classifier to only consider faces with an open mouth. Such a system can potentially also support singing voice detection tasks and can help to identify relevant frames for the creation of video thumbnails. The experimental evaluation could be significantly scaled by automating the image annotation process. This could be achieved by employing web-mining techniques to retrieve images of specific singers in an automatic, or at least semi-automatic, manner. Finally, an obvious extension would be the fusion of audio- and video-based approaches to increase performance and cater scenarios where one of two domains is of low quality.

6.4 Structural Annotation using a multi-label CNN

In this section, we present an application of CNNs in the context of structural segmentation of flamenco music recordings. In literature, the term structural segmentation, or structural analysis [138], generally refers to a segmentation of an audio file into organisational units

which may be repeated throughout the song, such as *intro*, *verse*, *bridge* or *chorus*. While similar concepts exist in some flamenco styles, the focus of the proposed approach is on a different musical dimension, that is instrumentation.

More specifically, we present an automatic annotation system which segments each music recording into sections that are consistent with respect to instrumentation. This is feasible, because in classical flamenco recordings, instrumentation is in most cases limited to singing with guitar accompaniment and rhythmic hand-clapping (referred to as *palmas*). Some styles are traditionally performed a cappella, and, in rare cases, performances can be instrumental.

Based on these characteristics, we focus on detecting the presence of four essential instrumental components:

- vocals
- palmas
- solo strummed guitar
- solo picked guitar

With respect to the presence of guitar playing, we specifically focus on solo sections only and discard the accompaniment while the singing voice is present. In flamenco, guitarists commonly take on a background role when the singing voice sets in and play at a significantly lower intensity compared to solo sections. The guitar accompaniment during vocal sections is therefore often limited to sparse chords played at low volume and in many recordings the guitar is barely audible while the singing voice is present. Initial experiments have shown that the accuracy for detecting picked and strummed guitar decreases by approximately 10% when guitar accompaniment in the presence of vocals are considered.

Figure 6.8 depicts schematically the outcome of the proposed annotation task.

Fig. 6.8 Schematic illustration of the segmentation of a flamenco recording with respect to instrumentation.

While this form of annotation is in itself is of value for the automatic indexing and visualisation of large flamenco collections, we show in Section 6.5 how the annotations can be mined to identify their potential in related computational tasks and to gain genre-specific musicological insights. To this end, we conduct a large-scale exploratory analysis of a music corpus containing more than 1500 commercial flamenco recordings [102].

6.4.1 System overview

A schematic overview of the proposed segmentation method is depicted in Figure 6.9. In order to detect sections of consistent instrumentation within a recording, we first extract a compact spectral representation of the audio signal in short-term windows. More specifically, we extract a two-dimensional feature matrix at each window position, which holds the log-mel-spectrogram of several short-term frames around the current time instance. Each feature matrix is then passed to a CNN with several convolutional, pooling and feed-forward layers. The output of the network consists of four sigmoid units, each of which indicates the presence or absence of one of the four instrumental components.

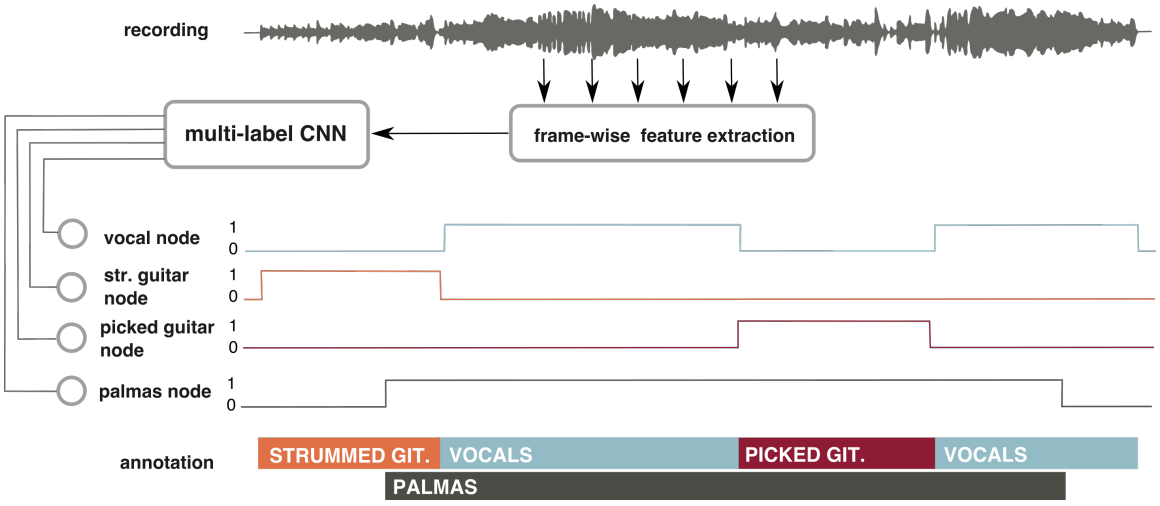


Fig. 6.9 Schematic view of the proposed segmentation method.

Given the large number of available feature matrices in the training database in relation to the number of songs, we train the network on a fixed number of randomly selected instances and apply data augmentation techniques in order to increase the variety of training examples and prevent overfitting.

The feature extraction and data augmentation methods used in the presented approach are loosely based on the the findings described in [180], where various techniques were compared in the context of a binary vocal detection task. We furthermore adopt the design of the convolutional and pooling layers used in [180] (which is a scaled-down version of the architecture described in [185]) and modify the output layer to suit the multi-label task at hand.

Below, technical details of all stages are described in further detail.

6.4.2 Pre-processing and feature extraction

The proposed method operates on short-term spectral features extracted from the audio signal. During training and prediction, the network is presented with a single set of features representing a given point in time. Therefore, before a signal is fed to a CNN, a short-term feature extraction stage extracts a sequence of feature vectors from its time-domain representation.

First, each audio recording is re-sampled to 22050 Hz and converted to mono format. The signal is then parsed with a moving window, 1024 samples long (46.4 ms), with a hop size of 315 frames (14.3 ms). At each window position, the Discrete Fourier Transform (DFT) is computed and passed as input to a mel filter-bank that yields 80 mel-band energy coefficients. Each mel-filter computes a weighted sum of the DFT coefficients that lie inside its frequency range [161]. The filter-bank covers bands ranging from 27.5 to 8000 Hz. We furthermore take the logarithm of the energy coefficients in order to reduce the dynamic range of the values. In addition, each band is normalised to zero mean and unit standard deviation, where both mean and standard deviation are computed over the entire training set.

After the short-term feature extraction stage has been completed, a subsequence of 57 successive feature vectors is aggregated each time to form a 80×115 image representation, that will serve to assign a classification label to the time instant corresponding to the middle of the subsequence. As a result, the short-term feature sequence generates a sequence of 2-D images. Each image reflects the evolution of spectral content (image height) over a neighbourhood of frames (image width) around the time instant for which a classification decision will be made. Given the short-term window step of 1024 samples and the sampling frequency of 22050 Hz, an image width of 115 frames corresponds to $\approx 1.6s$ of audio, i.e., to ≈ 0.8 s of audio around the time instant to which the classification label will be assigned. This approach is inspired by the two-dimensional pixel matrices that are used in image classification systems and has shown to give promising results in various audio analysis tasks (see for example [207, 79, 82]). The feature extraction process is depicted in Figure 6.10.

6.4.3 Data augmentation

A straightforward method of training the network would consist of extracting all available feature matrices from a given training set and passing them, together with the corresponding labels, to the network. This approach has the drawback that the network is presented with a large number of training instances per song, compared to the number of available recordings. More specifically, using a hop-size of 14.3 ms yields over 12500 training examples for a 3-minute recording. The number of available songs is however expected to be far lower, given that the training stage requires manual annotations (here we train on 80 manually annotated recordings). As a consequence, the network is prone to learning the timbre, melodic and

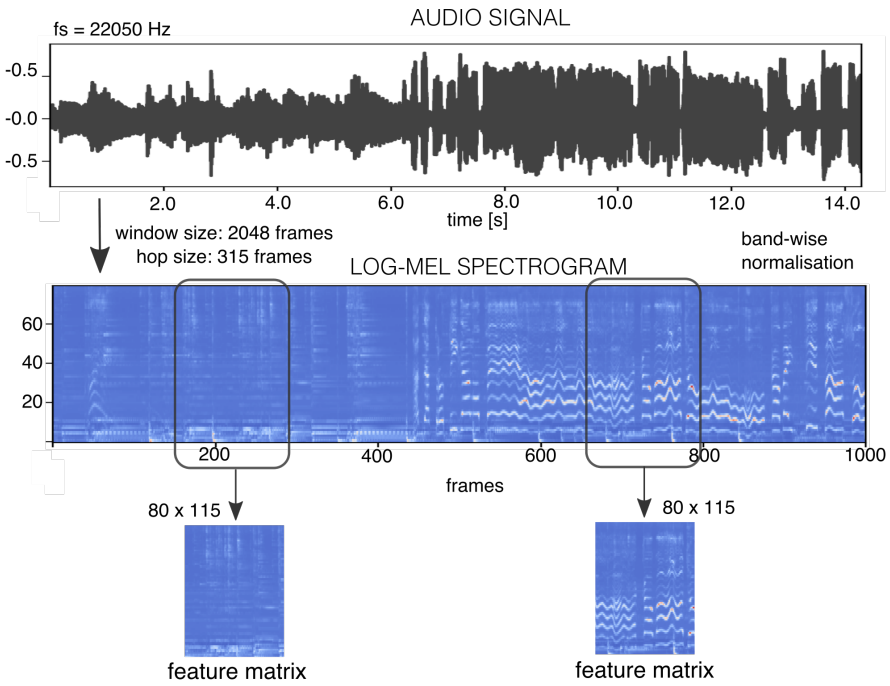


Fig. 6.10 Illustration of the feature extraction stage.

harmonic content of these songs instead of generic characteristics which generalise well to unseen data.

Therefore, we limit the total number of training instances to 40000, which are randomly drawn from the training data. We assume, that this amount covers a representative sample of the training space. We furthermore apply a set of randomised data augmentation techniques to each feature matrix to increase the variety presented to the network and thus prevent overfitting. The three augmentation strategies applied here are intended to mimic natural variety among a large number of recordings with respect to pitch range, key transposition, recording quality, timbre and tempo. These methods have shown to increase the detection accuracy and prevent overfitting in the context of vocal detection [180].

More specifically, we employ the following three augmentation stages to the spectrogram of each training instance:

- **Frequency-domain filtering** using a gaussian-shaped filter response following the equation

$$f(x) = d \cdot e^{(0.5 \cdot \frac{(x-\mu)^2}{\sigma^s})} \tag{6.34}$$

  where μ corresponds to a randomly chosen frequency bin between 150 Hz and 8 kHz, σ is randomly chosen between 5 and 7 semitones, and the attenuation d is randomly chosen between +10 and −10 dB.

- **Time stretching** with a randomly chosen stretching factor between 0.8 and 1.2.

- **Pitch shifting** with a randomly chosen stretching factor between 0.8 and 1.2.

As described in [180], both time stretching and pitch shifting can be efficiently implemented as affine image transformations on the two-dimensional spectrogram representation and frequency filtering is achieved by simply multiplying each frame of the spectrogram with the randomly generated frequency response.

6.4.4 CNN architecture

The CNN architecture employed in this study, as shown in Figure 6.11, is comprised of three main blocks: Each 128×22 input matrix is first passed through two consecutive convolutional layers with 64 and 32 convolutional masks (feature detectors) of size 3×3, and then subsampled in a 3×3 max pooling scheme. The second block consists of two further convolutional layers with 128 and 32 masks, again followed by a 3×3 max pooling layer. The output of each convolutional operation is passed through a *relu* function. The resulting $32 \times 11 \times 7$ tensor is then unfolded (flattened), yielding a 4928×1 one-dimensional representation, which is subsequently fed as input to a standard, fully connected feed forward neural network. This network consists of two hidden layers, of 256 and 64 units, and a *sigmoid* output layer holding the four units described above.

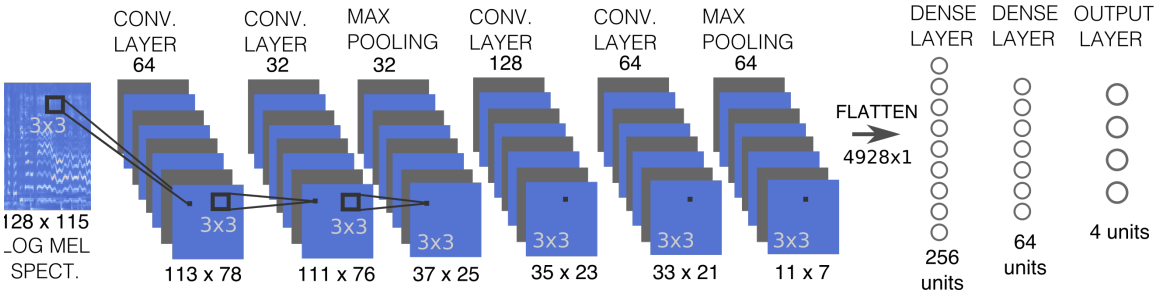


Fig. 6.11 Illustration of the CNN architecture.

As mentioned earlier, we aim at detecting solo strummed and picked guitar sections and discard the guitar playing technique when the singing voice is present. We furthermore assume that (as it is the case in the vast majority of classical flamenco recordings) only a single guitar is present. Consequently, the three classes vocals, strummed and picked guitar, are conceptually speaking mutually exclusive. The presence of the palmas is however independent of the presence of the other three components. We therefore model the problem as a multi-label classification task and do not hardcode the aforementioned mutual exclusivity into the network. We instead assume that the network will learn this property and we deal with ambiguous output in the post-processing stage described below. This design choice has the additional advantage that it allows us to detect silence, which is encoded in the output as all sigmoid units predicting the zero class.

Initial experiments have furthermore shown that the presented multi-label architecture yields similar performance that of to an ensemble of four CNNs, where each CNN solves a binary task, focusing on a single instrumental component.

6.4.5 Post-processing

As it was previously described, when an input image is given, the output at the four sigmoid units is interpreted as the multi-label prediction for the time instant corresponding to the middle of the respective frame subsequence. Given the mutual exclusivity of vocals, strummed and picked guitar by definition, it is necessary to define a procedure for the case of ambiguous output, when more than one of the classes is predicted at a given time instance. In this case, we simply set the label of the node with the highest activation to one and all remaining nodes to zero. Note that, as it was described above, the case when none of the classes is predicted as true is valid, since it corresponds to silence.

6.4.6 Classifier training

The proposed segmentation system is trained and evaluated on a set of 100 recordings which were manually annotated in the scope of this study. All songs are commercial flamenco recordings taken from the *corpusCOFLA* [102] collection. The annotated dataset was split into training (80 songs), validation (10 songs) and test (10 songs) sets. In order to create a realistic scenario, the splits are artist-filtered, meaning that no artists appears in more than one split. These standard measures are taken to reduce the risk that the classifiers will overfit to the timbral characteristics of a particular singer or recording style.

After the feature extraction stage is carried out on the aforementioned dataset, the network is trained for a maximum of 100 epochs using the *Adam* [96] algorithm for optimisation of the mean squared error over the training set. During each training epoch, the training images are shuffled and are grouped to form mini-batches (128 images per mini-batch). An early-stopping criterion is also used, which terminates the training procedure if the loss-value does not decrease significantly (at least by a value of 0.01) during a patience-period of 5 epochs.

6.4.7 Baseline methods

We evaluate the performance of the proposed method for each of the target classes separately by computing the binary classification accuracy as the percentage of correctly classified frames for a particular class over the test set.

In order to assess the advantage of using a deep network and to estimate the difficulty of the task itself, we furthermore compare the proposed method to the performance of an ensemble of shallow classifiers. More specifically, we train a separate classifier based on the

Method	vocals	palmas	strummed guitar	picked guitar
proposed	97%	96%	93%	95%
baseline	88%	77%	66%	69%

Table 6.2 Experimental results: Classification accuracy in % for all four tasks.

vocal detection method described in [188] for each of the four tasks. The method uses a based on a Gaussian Mixture Model (GMM) per class, trained on frame-level mel-frequency cepstral coefficients and log frequency power coefficients. For a detailed description, we refer to [188]. We furthermore smooth the resulting decision sequence with a median filter of 0.5s length.

All baseline methods and the CNN approach are trained and evaluated using the same split into train, validation and test set.

6.4.8 Experimental results

The results of the proposed CNN-based system are shown in Table 6.2 together with the performance obtained using the GMM-based baseline method. For all four tasks, the deep convolutional architecture yields a binary classification accuracy above 90%. The performance of the shallow classifier is significantly lower, in particular for the two tasks involving guitar playing techniques: The obtained classification accuracy for strummed guitar detection is only 66% and for picked guitar detection 69%. In comparison, the deep network yields classification accuracies of 93% and 95%, respectively.

These results indicate that the proposed system can generate automatic reliable annotations which can set the basis for related computational tasks and can be used in data-driven exploratory studies (see Section 6.5). The experiment has furthermore demonstrated the capabilities of CNNs to solve complex audio-based classification tasks, where standard shallow algorithms yield unsatisfactory results.

6.4.9 What did the networks learn?

A heavily criticised aspect of deep learning, and to a certain extend of machine learning algorithms in general, is the fact that the resulting computational model is not interpretable by humans. While all learned weights of a pre-trained deep architecture are accessible, the sheer number of parameters, which can be in the range of various millions, and the increasing level of abstraction throughout the layers, make it impossible for humans to understand what the network learned and which characteristics of the data it considers when making a decision. As a result, deep networks have gained a reputation of being "well-performing black boxes".

This aspect is problematic in various ways. While deep architectures can solve many complex problems, they do not provide the means for scientists to gain deeper knowledge of the problem itself. As a result, the problem solution does not yield a better understanding of the domain. In addition, the lack of transparency makes it difficult for engineers to estimate the robustness of a model with respect to unseen data characteristics and the development and fine tuning of deep networks is often reduced to numerous iterations of trial and error.

Recently, a number of visualisation techniques have been proposed in the image processing community in order to overcome these limitations and allow developers and users to better understand how a deep network operates and what it has learned [184, 55, 237]. In the context of image classification, *activation maps* or *salience maps* [237], have proven to be a useful tool in validating that a network bases its decision on relevant pixel areas within an input image.

Another, more generally applicable strategy, is *activation maximisation*. The idea is to artificially generate images, which, when passed as input to the network, will cause a high activation in a specific feature map in a particular convolutional layer [55, 184]. In this way, it is possible to estimate which characteristics of the input image are targeted by the corresponding convolutional filter. Case studies on networks trained for image classification[6] and handwritten character recognition [55] have revealed that first layer filters detect elementary structural components, such as edges of different orientation and shape. Deeper layers tend to target more complex textures, consisting of combinations of the basic shapes learned in prior layers.

Here, we apply the *activation maximisation* strategy to the audio domain and attempt to gain insight into the underlying mechanisms of the trained multi-layer CNN by generating input feature matrices which maximise filter maps of the first and third convolutional layer. Visualisations of the original input space (Figure 6.12) show that the presence of the targeted instrumental components can not only be heard, but also visually identified in the feature representation. We therefore directly follow the method described in [55] to investigate in how far the structure of our network is similar to the that of networks trained for image classification and handwritten character recognition.

Let x be the network input and $\mathbf{A}_n^{(l)}$ be the feature map of size $(I \times J)$ created by the n^{th} filter of the l^{th} convolutional layer. Assuming that the network is trained and all parameters are fixed, the values observed in the feature map $\mathbf{A}_n^{(l)}[i,j](x)$ only depend on the input image x. Mathematically speaking, we aim to generate an image x^*, s. th.

$$x^* = \max_x \sum_{i,j} \mathbf{A}_n^{(l)}[i,j](x). \tag{6.35}$$

[6] https://blog.keras.io/how-convolutional-neural-networks-see-the-world.html

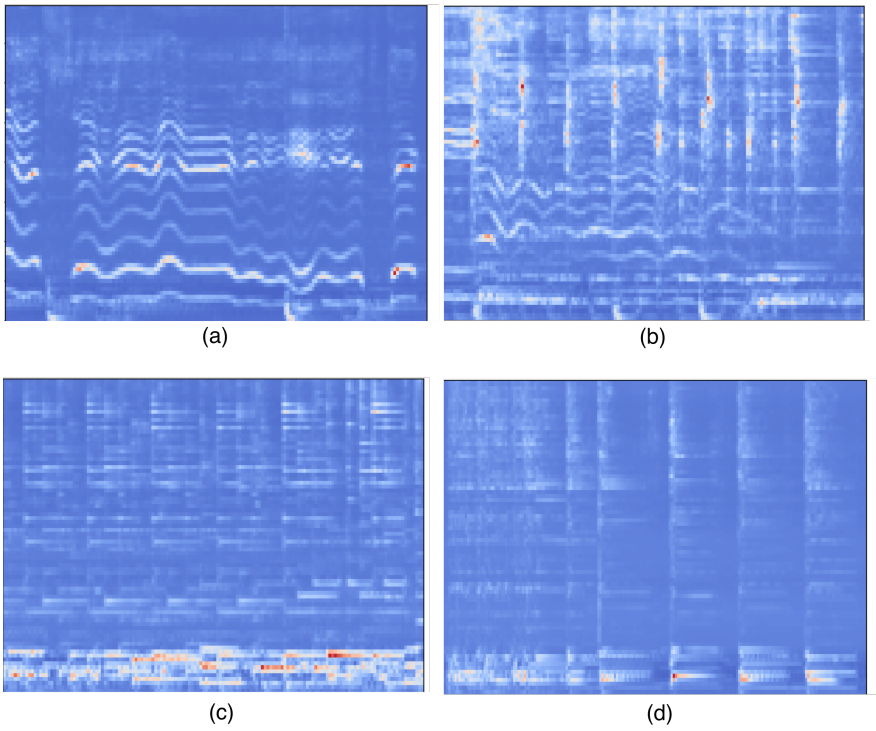


Fig. 6.12 Log-mel features extracted from audio excepts containing (a) vocals, (b) vocals and palmas, (c) picked guitar and (d) strummed guitar.

While this is a non-convex optimisation problem, we attempt to find at least a local maximum by employing *gradient ascend* based on the score function

$$\xi(x) = \sum_{i,j} \mathbf{A}_n^{(l)}[i,j](x) - r \cdot ||x|| \tag{6.36}$$

where the second term is an $l2$-norm with regularisation weight parameter r. Starting with random pixel values for x, at each step, we compute the gradient $\frac{\delta\xi(x)}{\delta x}$ w.r.t. the input image and update the pixel values as follows

$$x := x + \beta \cdot \frac{\delta\xi(x)}{\delta x} \tag{6.37}$$

where β is the learning rate. The values for parameters β and r were determined empirically by observing convergence for different settings. The regularisation parameter was set to $r = 1 \cdot 10^{-5}$ and the learning rate was set to $\beta = 0.001$ for maximising the first layer activation and to $\beta = 0.1$ for maximising the third layer activation.

Figures 6.13 and 6.14 show the resulting images for several convolutional filters of the first and third layer, respectively. Similar to the aforementioned image processing task, it can be seen that the first layer filters appear to focus on basic shapes, such as different types of edges.

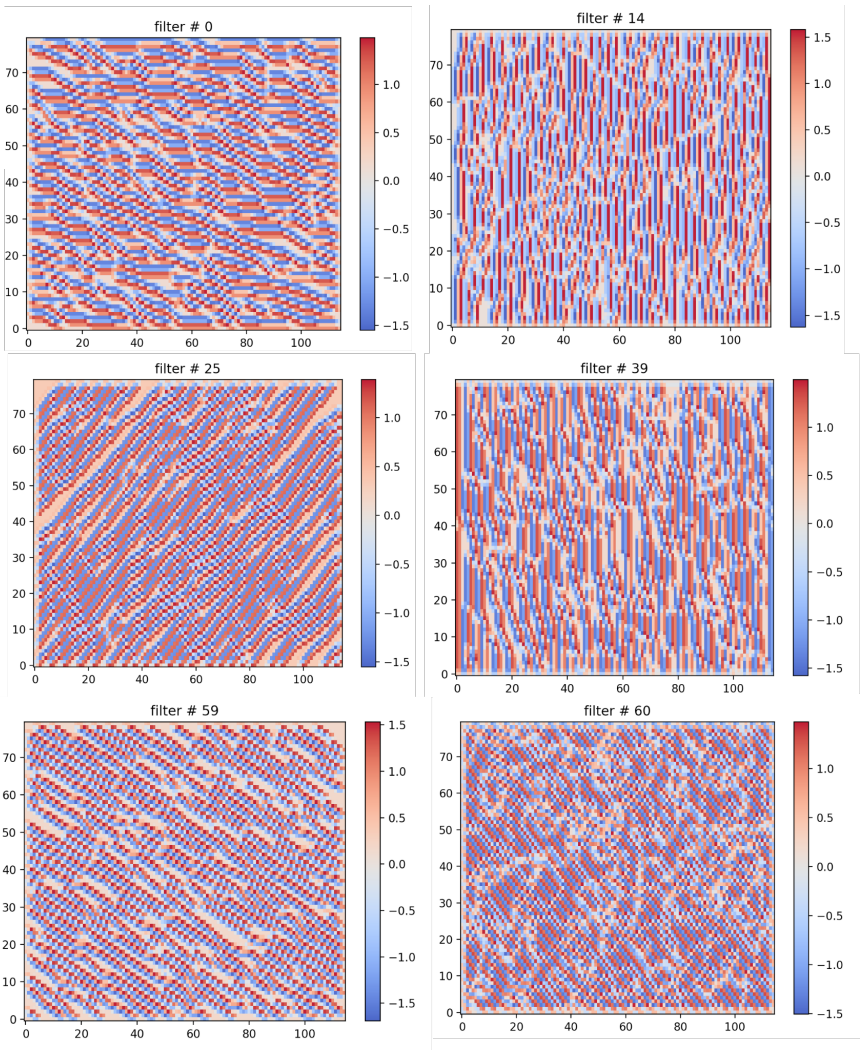


Fig. 6.13 Input feature matrices which maximise selected first layer filter maps.

The third layer filters seem to combine these elements into more complex textures. It is interesting to see that some of these textures show strong similarities with instrument-specific spectral patterns: The wave-shaped patterns in filter 1 bear a striking resemblance to the parallel continuous contours caused by vocal vibrato (Figures 6.12 (a) and (b)). Furthermore, the patterns in filters 34 and 53 are somewhat similar to the spectra produced by strummed guitar sections (Figures 6.12 (d)), where the percussive onset produces vertical lines across the spectrum, followed by parallel horizontal lines caused by the notes contained in the chord and their harmonics.

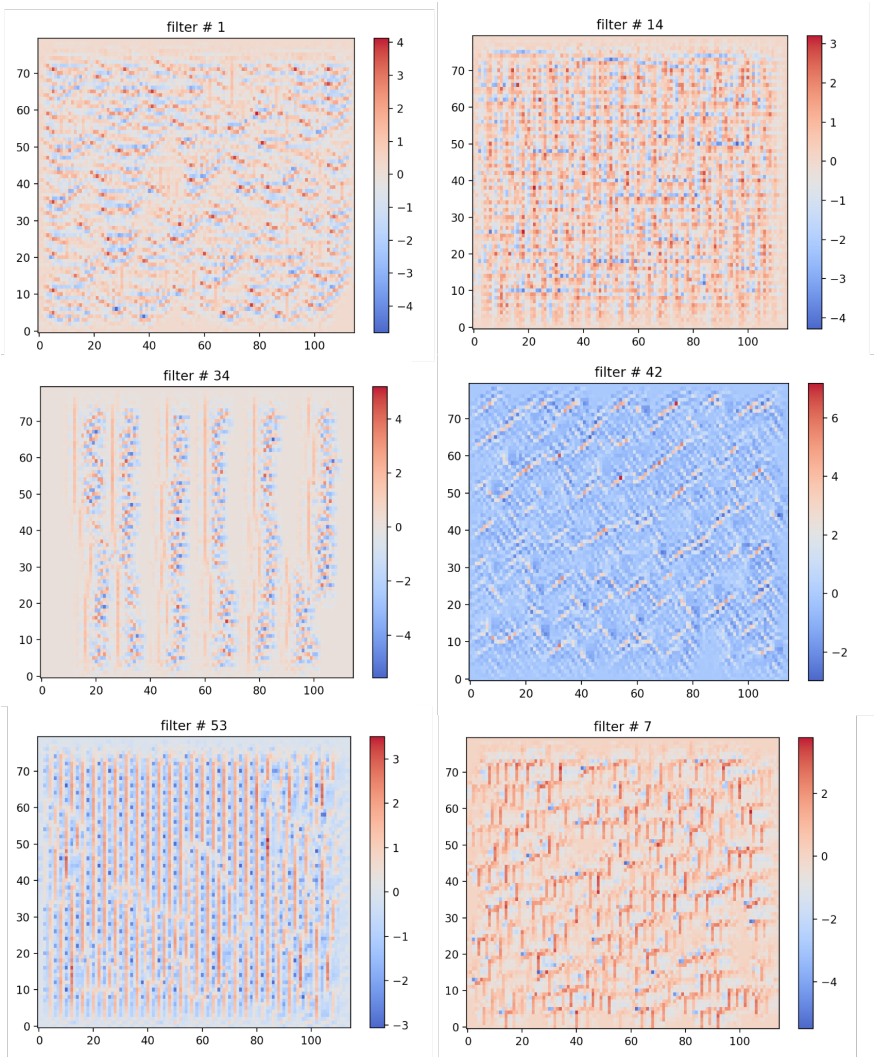


Fig. 6.14 Input feature matrices which maximise selected third layer filter maps.

6.4.10 Discussion

The proposed structural segmentation backend has shown to give reliable results for the task of detecting sections of consistent instrumentation. For all three subtasks, our experiments revealed, that the CNN-based approach significantly increases the performance compared to a baseline method using a shallow classifier. Despite the common conception of deep learning algorithms being "black boxes", we have managed to identify several characteristics of the target classes which the classifiers appear to have learned. In this analysis, it has become apparent, that the sort-term temporal information contained in the matrix representation of the input domain holds valuable information for all four classification problems.

6.5 Mining automatic structural annotations

In Chapter 1 it has been outlined, how the growing availability of digital music collections, in the form of symbolic music representations and audio recordings, along with recent advances in symbolic and audio-based MIR, has enabled researchers to scale descriptive music analysis [227, 10] from small collections of a few manually annotated items to large corpora of music content. For the particular case of non-Western music traditions, which often lack formal documentation, data-driven studies are a powerful means to reveal distinctive musical characteristics, validate musicological research hypotheses and identify meaningful features for the automatic organisation of digital collections.

In this section, we employ the previously described structural segmentation backend to explore a large corpus of flamenco music recordings. More specifically, we demonstrate how the automatically extracted annotations can be used to

(a) augment editorial meta-data with meaningful content-based descriptors and visualisations

(b) discover genre-specific characteristics and evaluate their potential for open MIR tasks

(c) illustrate music-theoretical concepts in a large-scale data-driven context

Our goal is to focus on computationally feasible analysis experiments and tasks that would require immense effort when conducted manually.

Data-driven studies have already shown significant potential in discovering knowledge and hidden correlations within and among music genres. The work in [151], which was based on a learned dictionary of vocal pitch contour elements, showed that singing styles with geographical proximity, exhibit similar characteristics. In [30], a comparison of melodic pattern occurrences between genre-specific corpora and an anti-corpus, revealed distinctive patterns for different folk song traditions. Using Subgroup Discovery techniques, the method in [194] generated a set of interpretable rules which can associate folk song transcriptions with their origin. An analysis of manually annotated harmonic progressions in rock music [35] revealed a number of frequently occurring chords and chord sequences, as well as evolutionary trends over several decades. In [219], a large-scale corpus study of ragtime scores led to the empirical validation of certain hypotheses that are related to the occurrence of rhythmic patterns.

Here we present, for the first time, an exploratory study of a large flamenco corpus [102], from a computational analysis perspective. In particular, we focus on the musical structure with respect to instrumentation and its relation to style and tonality. To this end, we extract automatic structural annotations using the backend described in the previous section and perform a data-driven mining study which addresses a number of computational and musicological aspects. In particular:

(a) We detect instrumental and a cappella recordings based on the estimated percentage of vocal and non-vocal frames.

(b) We explore intra and inter-style similarity based on song-level instrumentation statistics and investigate the usefulness of these features for automatic style recognition tasks.

(c) We verify music-theoretical assumptions on tonality with respect to style and instrumentation, by correlating pitch class profiles with mode templates.

First, we show that the visualisation of the automatically extracted structure of a recording yields a concise overview of its content and can reveal interesting musical properties. Then, we compute global statistics over the corpus with respect to instrumentation and describe how the backend can be used to automatically identify a cappella and instrumental recordings. At a next stage, we examine how styles differ with respect to instrumentation and respective structure and investigate the potential use of the segmentation backend in automatic style classification systems. In addition, we illustrate the relationship between style, instrumentation and tonality via an example of three popular styles with distinct harmonic characteristics.

The results of this study can be further exploited by several potential applications for automatic indexing and content-based retrieval in digital flamenco collections, can provide valuable clues towards the unresolved problem of automatic style detection [100], and can reveal interesting musicological observations.

6.5.1 Corpus

We focus on Version 1.1 of *corpusCOFLA*[7] [102], which has been introduced in Chapter 1 as a large collection of commercial recordings. In its most recent update, the annotated styles, which exhibit varying levels of detail and sub-style distinction, were classified into 86 categories and artist names were assigned unique identifiers. An overview of the statistics of the corpus is given in Table 6.3 and the ten most frequently occurring styles are shown in Table 6.4.

6.5.2 Visualization of the structural annotations

Before proceeding with the analysis of the corpus as a whole, we show on the basis of two examples, how the output of the classifiers can provide a compact, content-based visualization of a flamenco recording.

Figure 6.15 (a) presents the automatic segmentation of the song *Me fuí detras de los míos*, performed by *Manuel Sordera*. This particular recording, which contains two styles,

[7]http://www.cofla-project.com/?page_id=170

no. recordings	1594
no. styles	86
tracks without style information	109
tracks containing multiple styles	104
no. singers	364
no. anthologies	10

Table 6.3 Statistics of the *corpusCOFLA*.

style	**no. of recordings**
Fandangos	203
Soleares	150
Siguiriyas	124
Bulerías	120
Malagueñas	73
Tangos	59
Alegrías	43
Granaínas	39
Tarantas	37
Tientos	34

Table 6.4 10 most frequently occurring styles in the *corpusCOFLA*.

starts with the *tientos* style and ends with a *tango*. It is not uncommon that these two styles are encountered in a single performance, in this particular order. As a matter of fact, they are closely related styles because they follow a common harmonic progression and rhythmic structure. However, a *tango* is performed at a faster tempo with stronger rhythmic accentuation.

The visualization illustrates some further interesting differences between these two styles in this recording. The *tientos* part, which lasts from the beginning until approximately second 220, is characterized by long vocal segments, with guitar interludes alternating between picked and strummed sections. With the onset of the *tangos* part, the *palmas* set in to emphasize the underlying rhythmic pattern and the vocal sections become shorter. The guitar accompaniment is limited to strumming, which, due to its percussive nature, further increases the accentuation of the rhythm. Figure 6.15 (b) shows the same form of representation for a recording containing only *tientos*. Since in this recording, there is no modulation to *tangos*, there are no *palmas* and the guitar moves between picked and strummed sections until the end of the recording.

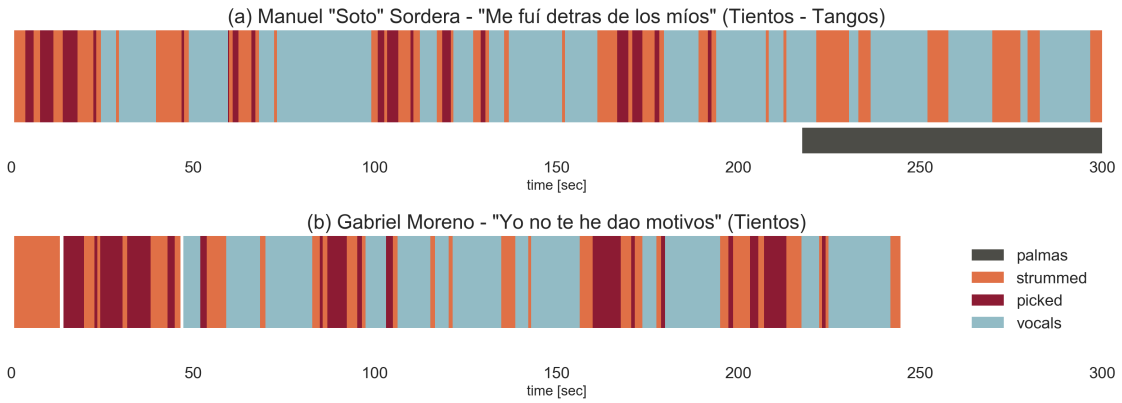


Fig. 6.15 Example of visualizing structure: (a) *tientos - tangos* and (b) *tientos*.

The proposed visualization can be combined with other content-based descriptors that cover different musical aspects. For example, in Figure 6.16, structural annotations are combined with descriptors related to melodic repetition. Specifically, the arcs connecting vocal segments correspond to the five highest similarity values among all vocal sections. The thickness of the arc is proportional to the respective similarity value, which is computed using the alignment method in [106]. It can be seen that there are two groups of similar vocal sections: The first one is formed by sections 2, 5 and 8, and the second consists of groups 7, 10, 13 and 16.

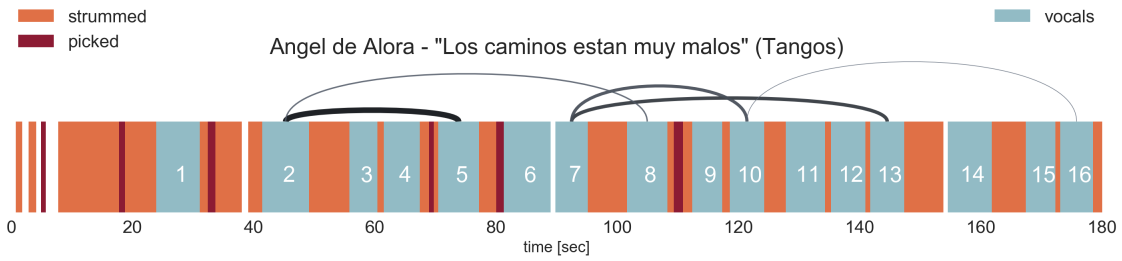


Fig. 6.16 Example of visualizing structure and vocal section similarity.

6.5.3 Global statistics

The extraction of the automatic annotations for all the recordings of the collection enables the computation of global statistics related to the presence of different instrumentation scenarios. Table 6.5 shows the percentage of non-silent frames in which vocals, palmas, picked guitar and strummed guitar were detected.

It can be seen, that only 57% of the analyzed frames are estimated to contain vocals. This is an interesting observation, because flamenco is known to be heavily centred around

	% of all non-silent frames
vocals	57%
palmas	20%
strummed guitar	24%
picked guitar	19%

Table 6.5 Global instrumentation statistics of the *corpusCOFLA*.

the singing voice, the core element and origin of the flamenco tradition. However, at the same time, this is a meaningful result, because many performances contain long solo guitar introductions and interludes. The remaining 43% of solo guitar sections can be further divided to 24% of strummed and 19% of picked guitar. Furthermore, *palmas* are detected in 19% of all frames. Even though there are no strict rules about the presence of *palmas*, they are unlikely to be found in some of the styles played in slow tempo, including *fandangos*, *soleares*, *siguiriyas* and *malagueñas*, which comprise a large part of the corpus.

6.5.4 Discovery of instrumental and a cappella recordings

Although a flamenco music performance usually contains both singing voice and guitar accompaniment, certain styles, which belong to the *tonás* family, are traditionally performed a cappella. These styles are of particular interest to musicological studies, since they are assumed to represent the origin from which flamenco evolved to its present form. Consequently, a number of musicological and computational studies have targeted their particular characteristics [19, 198, 105]. However, assembling a representative corpus of a cappella flamenco recordings is not a trivial task, because the editorial metadata of commercial recordings do not contain information on the existence or absence of guitar accompaniment and style annotations are most often missing in online repositories and audio sharing platforms. As a support to this claim, so far, there only exists one such dataset of 72 manually assembled recordings of the *tonás* family[8].

Here, we aim to automatically discover a cappella recordings within the *corpusCOFLA* collection. To this end, we begin by assuming that, due to the absence of the guitar, the non-silent frames of an a cappella recording will be mostly classified to the vocal class. Therefore, to retrieve the a cappella recordings, we first set a high threshold for the percentage of non-silent frames in which vocal are detected in a recording and we evaluate the results. Then we lower the threshold by 1% each time and we repeat the retrieval operation. In every iteration, we evaluate the results by reporting the retrieval precision, which indicates the

[8]https://www.upf.edu/web/mtg/tonas

fraction of relevant instances among all retrieved recordings. We also note the total number of retrieved songs.

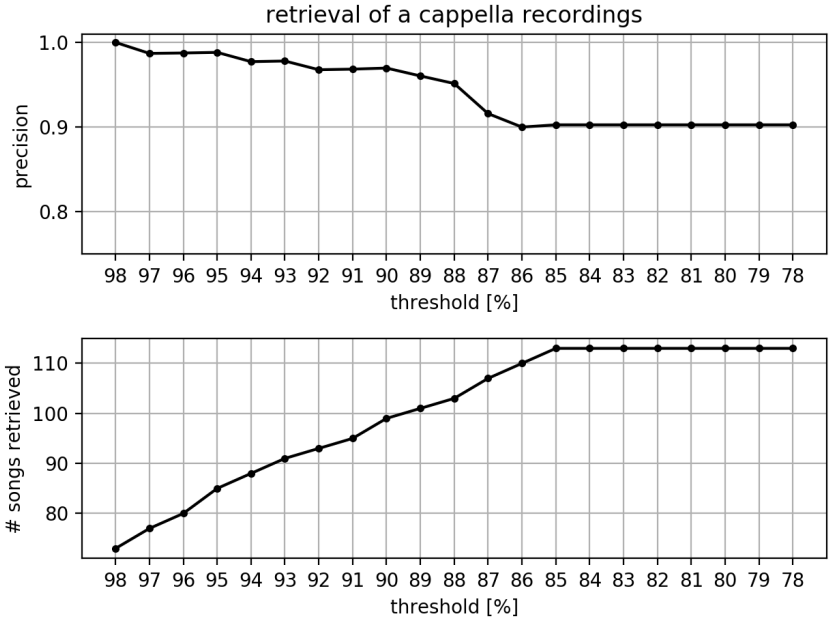


Fig. 6.17 Discovery of a cappella recordings: Retrieval precision and total number of retrieved songs for varying thresholds.

The results in Figure 6.17 show that, when the threshold value is 98%, none of the retrieved recordings contain instrumental accompaniment and are therefore true hits. In this way, we identify 72 a cappella recordings. When the threshold is further decreased, some more a cappella recordings are retrieved to the expense of an increasing number of false positives. A manual inspection showed, that the false positives are performances which contain a guitar background in the presence of vocals, i.e., not in solo guitar interludes, hence the high percentage of vocal frames. This is also due to the setup of our vocal classifier, according to which, a frame is classified in the vocal class as long as it contains vocals, i.e, even when there exists simultaneous guitar accompaniment. For a threshold value of 78%, we retrieve a total of 110 recordings, out of which 102 are performed a cappella.

Similarly, we approach the discovery of instrumental recordings by analysing songs with a low percentage of vocal frames. As in the previous experiment, we compute the retrieval precision for varying threshold values. In the case of instrumental recordings, the threshold refers to the maximum allowable percentage of vocal frames, i.e., the percentage below which a recording is considered to be instrumental. The results in Figure 6.18 show that for threshold values below 17%, all of the retrieved recordings are instrumental. In this way, we were able to identify 10 instrumental recordings in the corpus. Increasing the threshold however does not yield any further true positives. As mentioned earlier, instrumental performances are rare

in classical flamenco and it can be stated that solo guitar pieces have only recently gained popularity [189].

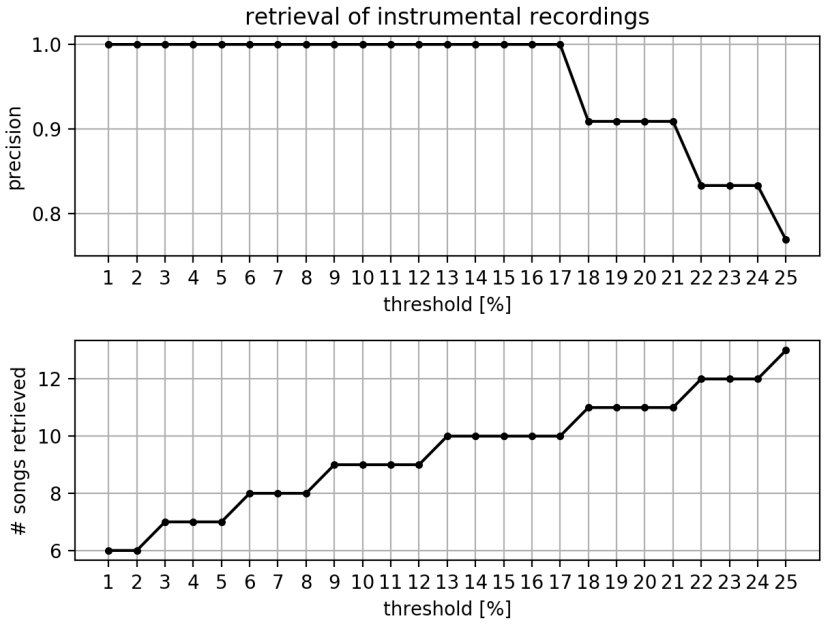


Fig. 6.18 Discovery of instrumental recordings: Retrieval precision and total number of retrieved songs for varying thresholds.

These two experiments have shown that the structural annotations can be used for the semi-automatic (if perfect precision is required) and automatic (if a percentage of false alarms can be tolerated) identification of a cappella and instrumental recordings in large flamenco collections. The resulting metadata represent an important augmentation to the commonly provided editorial meta-data scheme, which usually lacks this type of information. Furthermore, the creation of collections of a cappella recordings with minimal manual intervention is of great importance for scaling existing computational approaches. Furthermore, the automatic tagging of instrumental recordings is important for genre-specific recommender systems, because solo guitar flamenco recordings may appeal to a different audience than classical flamenco, which mainly evolves around the singing voice.

6.5.5 Differences in instrumentation across styles

Flamenco styles are distinguishable by a complex set of harmonic, melodic and rhythmic characteristics. Computational approaches to automatic style classification have so far been limited to a small number of sub-styles which can be identified based on common melodic templates [43, 135]. This concept has been explored in detail in Chapter 2. However, the

generic problem of automatically detecting the style of a flamenco recording remains largely unresolved [100].

The occurrence of different instrumentations across styles has not yet been formally studied. Therefore, in order to gain a better insight into style-specific characteristics and explore the potential use of the proposed structural segmentation backend for the purposes of automatic style classification, we investigate if certain types of instrumentation are more likely to occur in particular styles. To this end, each song is represented as a feature vector, where each feature dimension stands for the percentage of frames containing one of the four instrumental components: vocals, *palmas* and strummed and picked guitar. This representation is then used to assess differences across styles on a multi-dimensional plane. In the scope of this study, we focus on the ten most frequently occurring styles of the corpus (Table 6.4) and recordings containing multiple styles are excluded.

In a first experiment, we visualise pairs of styles with respect to the song-level percentages of *palmas*, vocals and picked guitar, and, for some cases, we observe that pairs of these features exhibit style-specific behaviour. Two examples are shown in Figure 6.19. The left scatter plot shows the percentage of picked guitar vs. the percentage of *palmas* for the styles of *bulerías* and *malagueñas*. It can be seen that the two styles are almost linearly separable on the respective two-dimensional plane. While most *bulerías* exhibit a high percentage of *palmas*, the *palmas* are largely absent in the *malagueñas* style. Regarding the guitar playing technique, picking is dominant in *malagueñas*, whereas the *bulerías* mainly contain strummed guitar.

Using our prior knowledge on a cappella styles, we now compare recordings of *bulerías* to the *tonas* family, which encompasses most a cappella styles of flamenco music. Figure 6.19 (right) presents the joint distribution of the percentage of vocals and *palmas* for the two classes. As expected, the a cappella recordings contain a high percentage of vocal frames, mostly above 90% and do not contain *palmas*. Some outliers can be explained by the fact that one of the a cappella styles, the *martinete*, is usually accompanied by stokes of a hammer on a anvil. A manual inspection revealed that this accompaniment was often mistakenly classified as either *palmas* or picked guitar (due to the harmonic nature of the produced sounds). The combination of the low percentage of vocals and the presence of *palmas* in the *bulerías* leads again to a practically linearly separable visualisation of the two styles.

Based on these observations, we now aim to understand how well this feature representation of a single vector per recording can discriminate the ten most frequent styles in the corpus. Therefore, we compute pair-wise Euclidean distances among all involved recordings. We then employ multi-dimensional scaling (MD) [156] to the resulting similarity matrix to estimate a two-dimensional layout of the data points. In the resulting space, recordings with similar feature values are placed in proximity, whereas recordings with different characteristics are placed further apart.

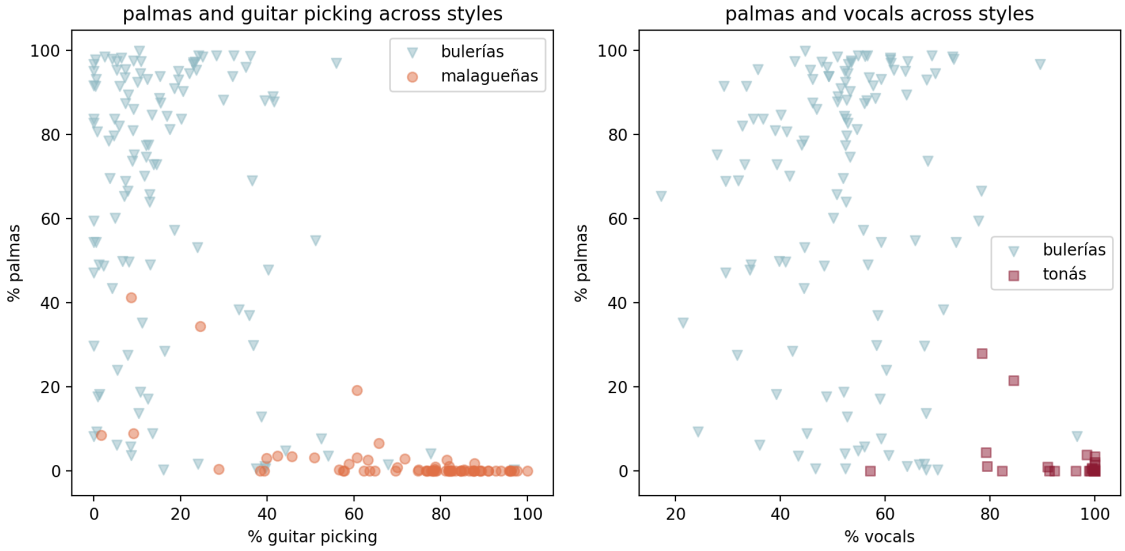


Fig. 6.19 left: % of picked guitar frames (x-axis) and % of *palmas* frames (y-axis) for *bulerías* and *malagueñas*; right: % of vocal frames (x-axis) and % of *palmas* frames (y-axis) for *bulerías* and *tonás*.

The computed layout is shown in Figure 6.20. It can be seen that two clusters are formed, one of which contains mainly recordings belonging to the *bulerías* style. The remaining styles are located in a second cluster. Within this group, we observe a slight tendency of the *malagueñas* forming a tight subgroup. The *algerías* and *tangos* appear to spread over both groups. Reducing the input space to a smaller number of styles reveals further sub-structures in the larger cluster. As an example, Figure 6.21 shows the MDS-layout for *siguiriyas*, *malagueñas* and *bulerías*. Here it can be seen that three style-specific clusters are formed.

The visually identified separation into two large groups can be quantified by applying an unsupervised clustering to the data. More specifically, we employ the k-means algorithm [80] and set the number of clusters to $k = 2$. We then analyse the distribution of recordings per style across the two clusters. The result is shown in Table 6.6, where it can again be seen that the *bulerías* are largely located in the second cluster. *Algerías* and *tangos* are spread over both clusters and the remaining styles are largely contained in the first cluster. Increasing the number of clusters did not yield any further style-specific separation.

6.5.6 Tonality across instrumentation and styles

In a last experiment, we illustrate flamenco-specific music theoretical concepts about the relationship between style, instrumentation and tonality in a large-scale data analysis environment. In flamenco, we encounter, apart from major and minor mode, a third mode, often referred to as the *flamenco mode*. Most styles are restricted to one of these three modes.

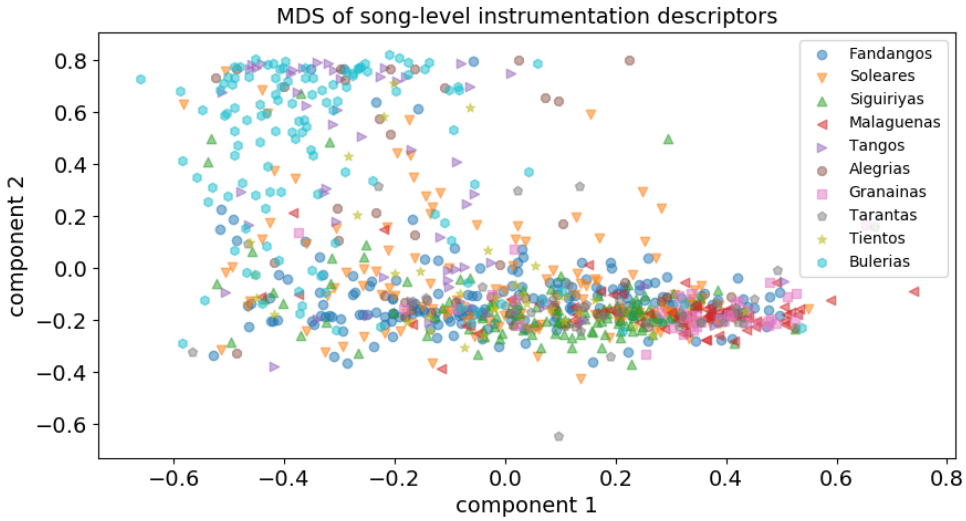


Fig. 6.20 MDS layout of pairwise distances between song-level descriptors.

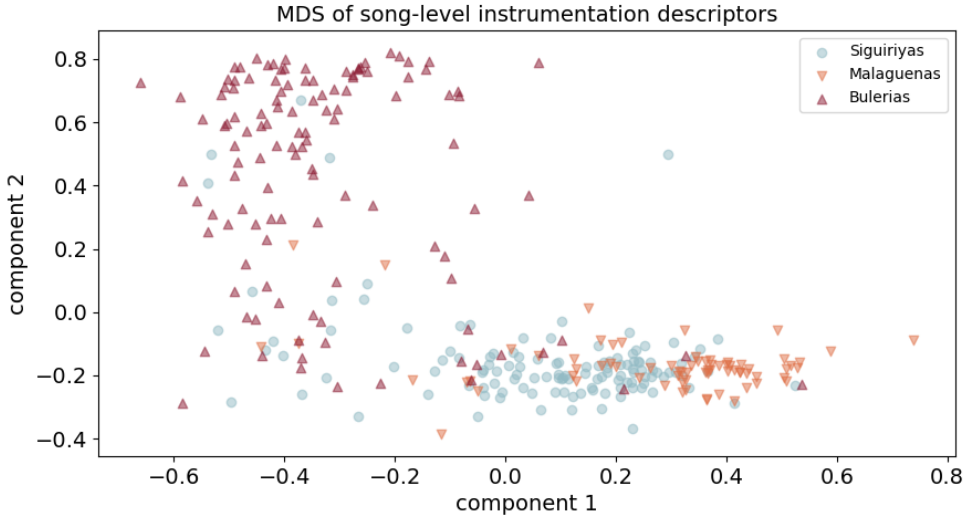


Fig. 6.21 MDS layout of pairwise distances between song-level descriptors.

For example, alegrías, are set in major mode, and *seguiriyas* are set in *flamenco mode*. The *fandangos* are an exception, because they are described in the literature as bimodal, in the sense that guitar solo sections are performed in *flamenco mode*, while singing voice sections are performed in major [58]. Here, we study this phenomenon in a data-driven approach on a larger scale. Specifically, we analyse the estimated tonality for three styles, i.e., the *fandangos*, *seguiriyas* and *alegrías*, in vocal, strummed and picked guitar sections.

A common method to determine the tonality of a piece is to extract its *pitch class profile* and compare it to tonality-specific *pitch class templates* [197]. Pitch class profiles quantify

style	cluster 1	cluster 2
Fandangos	95%	5%
Soleares	85%	15%
Siguiriyas	96%	4%
Bulerías	21%	79%
Malagueñas	97%	3%
Tangos	44%	56%
Alegrías	58%	42%
Granaínas	97%	3 %
Tarantas	92%	8%
Tientos	82%	18%

Table 6.6 Outcome of the k-means clustering: Style-wise distribution across clusters.

the occurrence of scale degrees throughout a composition. The statistical occurrence of scale degrees is closely related to the perceived tonality [25] and pitch class templates are prototypical scale degree occurrence patterns, specific to a particular tonality. These are either extracted from large annotated corpora or are estimated through listening experiments [107, 197]. If the pitch class profile of a given piece exhibits a high correlation with the pitch class template, it is likely that the piece will be perceived as belonging to the tonality associated with the template.

In line with this approach, we estimate the tonality of a flamenco recording by correlating its pitch class profile with tonality-specific templates. To this end, we first extract *harmonic pitch class profiles* (HPCP) [67] on a frame-level basis (non-overlapping frames of length 4096 samples), and then average over vocal, strummed and picked guitar segments. In this way, we create three pitch class profiles for each recording.

For the major mode, we use the template from [197]. Furthermore, we extract a flamenco mode template from all recordings in the corpus belonging to the *soleares* style. This process requires key normalisation per recording, in order to account for possible transposition among performances. Therefore, we normalise each pitch class histogram to a randomly selected reference example, by computing the correlation between circularly shifted versions of the histogram and the reference. For each recording, the pitch shift yielding the highest correlation is applied to all HPCP vectors.

We then compute the correlation between the pitch class profiles of the recordings under study and both templates. Here, we again normalise each profile to the key of the respective template. The results, broken down into styles and instrumentation, are shown in the form of kernel density plots in Figure 6.22. High values on the x-axis correspond to a high correlation with the major mode template and consequently indicate a strong major mode identity. Similarly, high y-values correspond to a strong flamenco mode identity.

It can be seen, that across instrumentation, the *algerías* show a higher correlation with the major mode template, and the *siguiriyas* with the flamenco mode template. This observation is in line with theoretical studies [57]. For the style of *siguiriyas*, we observe less variance for picked guitar sections, compared to vocal and strummed guitar sections. In the plots displaying the *fandango* recordings, we can see the bi-modality of the style. Specifically, both picked and strummed guitar sections exhibit a higher correlation with the flamenco mode template. In contrast, the vocal sections yield a higher correlation with the major mode template. The correlation with the flamenco mode is however weaker than for the case of the *siguiriyas*.

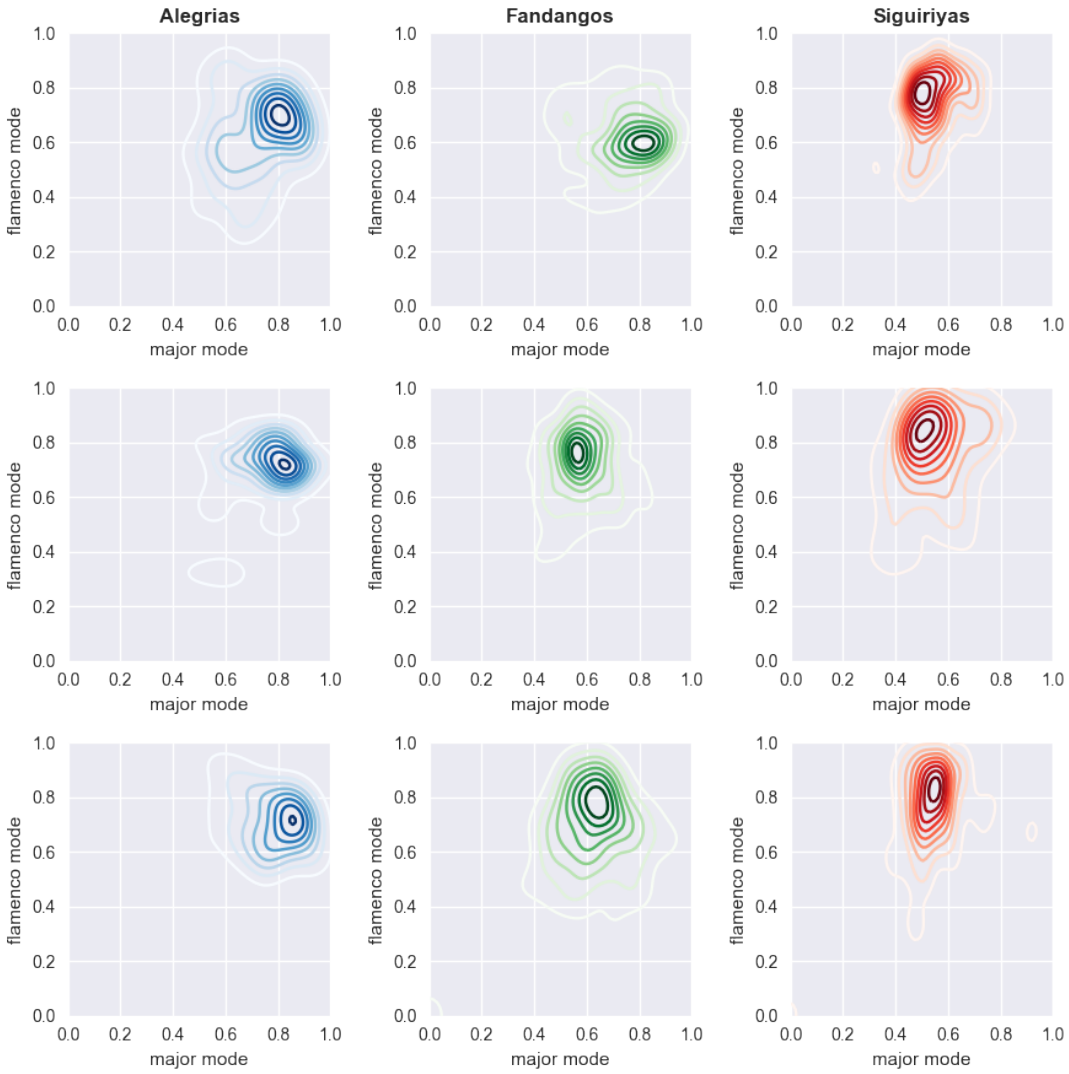


Fig. 6.22 Kernel density plots displaying the correlation of pitch class profiles with pitch class templates across styles and instrumentation.

6.5.7 Discussion

This exploratory corpus study has demonstrated the potential of mining automatic structural annotations of a large flamenco collection. We have shown, how the visualisation of regions of consistent instrumentation provides a compact content-based description of a recording which can reveal interesting musicological observations. We furthermore demonstrated, how the automatic annotations can be used to efficiently retrieve instrumental and a cappella recordings in an automatic or semi-automatic manner. Investigating the relationship of instrumentation and style gave several indications for existing correlations which could be potentially exploited in the open problem of automatic style recognition. In addition, we verified music theoretical assumptions on the relationship between style and tonality in a data-driven manner.

6.6 Conclusions

In this chapter, we explored the application of CNNs, a deep learning architecture, in the context of automatic content-based description and discovery of flamenco music. This work was motivated by the recently reported successful application of CNNs, and deep learning in general, to various computer vision and machine listening tasks. After reviewing theoretical foundations and implementation aspects of CNNs, we presented their application to two flamenco-related tasks, singer identification in flamenco videos and structural segmentation of flamenco music recordings.

The proposed image-based singer identification system relies of a number of state-of-the-art image processing and computer vision technologies, which are readily available in open source libraries. However, it has been shown, that their application to the task at hand goes beyond a trivial re-use of existing techniques, since problem-specific adaptations were necessary and the integration of several processing blocks required a certain level of domain-specific knowledge. The core of the framework is a pre-trained CNN which extracts an embedding with high discriminative power among faces from a given input image. This network was developed and trained for the particular purpose of being re-used in face recognition and authentication tasks, without the need to re-train on a problem-specific candidate set. While the proposed method has given promising results, we have identified a number of possible extensions and open problems. In general, the use of computer vision for flamenco description and analysis, is an interesting research area in itself, which should be further pursued in the future.

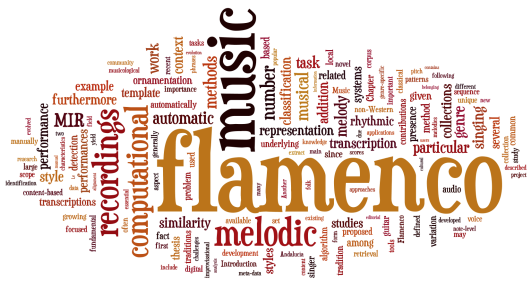
In a next step, focusing on the audio domain, we proposed a structural segmentation backend using a multi-label CNN. The method, which detects regions of consistent instrumentation in flamenco music recordings, has shown to outperform a baseline method which uses an ensemble of shallow classifiers. More importantly, it has demonstrated the potential

of deep learning for the analysis of flamenco music recordings in general, and future work could be directed at applying such techniques to similar tasks in the context of flamenco and other non-Western music traditions.

Moving towards the area of data mining, we then applied the developed segmentation backend to a large corpus of commercial flamenco recordings and explored the resulting automatic annotations in a data-driven study. Our goal was to enable the realisation of musicological findings that would otherwise require time-consuming manual procedures, and verify, at a large scale, via computational means and a data-driven approach, existing musicological observations. In addition, we aimed at discovering the potential of this type of annotations for related indexing and classification tasks. This computational study, which revealed a number of trends and correlations, is the first of its kind in the context of flamenco music.

Chapter 7

Summary and future work



Motivated by the growing availability of digital flamenco collections and the current shortcomings of MIR systems, which often do not generalise well to non-Western music traditions, this thesis addressed several challenges in the context of automatic content-based description of flamenco music recordings. The conducted computational studies and proposed methods encompass various musical aspects and focus on a several tasks, which are not only of importance for the automatic organisation of large flamenco collections, but do furthermore provide the means for data-driven studies which aim at a deeper understanding of the genre.

The catalysts for designing genre-specific solutions are diverse: In some cases, the particular characteristics of flamenco music imply computational challenges, which are simply absent in other genres. In other cases, the underlying musical assumptions of existing MIR systems do either not hold for flamenco music, or, the incorporation of genre-specific knowledge can simplify the problem formulation and thus increase system performance.

The first three chapters of this thesis focused on computational tasks related to the flamenco-specific phenomenon of reoccurring, often style-specific, melodies. In the past, this characteristic has been addressed from a computational perspective in the context of melody classification and several methods have been proposed, all following a common framework. In Chapter 2, existing work was extended by providing a systematic study of signal representations with different levels of abstraction. Furthermore, several evaluation metrics were proposed for supervised and unsupervised melody recognition scenarios. While

this work focused on classifying a given melody among a set of candidates, the related task of melody retrieval refers to the challenge of locating a given melodic segment in a large corpus. In Chapter 3, a novel method was proposed, which addresses this task in the context of flamenco music. Given that scores are usually not available, the algorithm operates on automatic transcriptions and specifically considers the presence of heavy ornamentation, which is typically encountered in flamenco music. Finally, Chapter 4, approached the novel task of extracting the underlying melodic template from a set of performances of the same melody. To solve this problem, two approaches were presented. The first relies on techniques of computational geometry and approximates the template as a piece-wise linear function. The second uses multiple sequence alignment to construct a graph model of the set of performances, which allow us to extract the template in form of a discrete pitch sequence.

Chapter 5 addressed the challenge of discovering repeated sung phrases in folkloric flamenco styles and other European folk music genres with similar structural characteristics. After identifying shortcomings of existing methods with respect to this task, a novel method was proposed, which first segments an automatic transcription into phrases and then detects repetitions among them. The system performance, and its limitations, were assessed in an experimental study involving three music genres.

Chapter 6 explored the potential of deep learning methods in the context of flamenco description and discovery. A first study addressed the often encountered missing singer annotation in flamenco video collections. To this end, an image-based singer identification framework was proposed, which automatically detects the face of the singer in flamenco videos and recognises his or her identity among a set of candidates. A second application used a multi-label CNN to segment a flamenco music recording into regions of consistent instrumentation. The resulting annotations were then explored in a large-scale computational study, which revealed a number of interesting musicological observations and identified potential applications of the proposed method in related MIR tasks.

While all of the proposed methods showed promising results with respect to accuracy and efficiency under lab conditions, a significant amount of software engineering and scaling efforts are required to make them applicable to large-scale real-world applications. However, these necessary steps lie largely outside the scope of interest, skills and funding of MIR research and do consequently require the collaboration and support of stakeholders, like music archives and public institutions [37]. Future research initiatives and funding proposals in the area should consider this aspect in order to transform the existing research code into user-level applications.

Another caveat of current computational approaches to flamenco music is the limited availability of data. Machine learning, and in particular deep learning, which has shown potential in the work described in Chapter 6, heavily relies on large annotated datasets. However, gathering and annotating such large amounts data manually, is a tedious, time-consuming

task. Future work could therefore explore web-mining and crowd-sourcing techniques which could at least partially automate the data collection process.

Lastly, the work conducted in this thesis has opened up a number of promising research directions, which can be pursued in future work. One example is the incorporation of the image domain into the computational analysis of flamenco music. The *centro andaluz del flamenco* alone holds 1,300 video recordings of flamenco performances and many more can be found in open online repositories. Combining audio and video analysis can potentially yield more robust systems and allows furthermore to investigate the relation between both domains, for example when studying emotion and expression. In addition, image analysis could be used to study flamenco dancing, an aspect of the genre which has so far not been approach from a computational perspective. A major, so far unresolved, challenge in computational flamenco description is the automatic recognition of the style. While several promising leads and features were identified in the scope of this thesis, the general problem remains open. Given the importance of style affinity in the categorising flamenco recordings, the complexity and multi-dimensionality of style-specific characteristics, and the lack of formal definition of the concept, this task is probably the most crucial, and at the same time ambitious objective of future research.

References

[1] Ahuja, R. K., Mehlhorn, K., Orlin, J., and Tarjan, R. E. (1990). Faster algorithms for the shortest path problem. *Journal of the ACM (JACM)*, 37(2):213–223.

[2] Aichholzer, O., Caraballo, L. E., Díaz-Báñez, J. M., Fabila-Monroy, R., Ochoa, C., and Nigsch, P. (2015). Extremal antipodal polygons and polytopes. *Graphs and Combinatorics*, 31:321–333.

[3] Aloupis, G., Fevens, T., Langerman, S., Matsui, T., Mesa, A., Nuñez, Y., Rappaport, D., and Toussaint., G. (2006). Algorithms for computing geometric measures of melodic similarity. *Computer Music Journal*, 30(3).

[4] Ambos, B., Ludwiczuk, B., and Satyanarayanan, M. (2016). Openface: A general-purpose face recognition library with mobile applications. Technical Report CMU-CS-16-118, CMU School of Computer Science.

[5] Barba, L., Caraballo, L. E., Díaz-Báñez, J. M., Fabila-Monroy, R., and Pérez-Castillo, E. (2016). Asymmetric polygons with maximum area. *European Journal of Operational Research*, 248(3):1123–1131.

[6] Bastian, M., Heymann, S., and Jacomy, M. (2009). Gephi: An open source software for exploring and manipulating networks. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*.

[7] Bayard, S. (1950). Prolegomena to a study of the principal melodic families of british-american folk song. *Journal of American Folklore*, 63(247):1–44.

[8] Benetos, E., Dixon, S., Giannoulis, D., Kirchhoff, H., and Klapuri, A. (2013). Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434.

[9] Benson, D. J. (2008). Music: a mathematical offering. *The Mathematical Intelligencer*, 30(1):76–77.

[10] Beran, J. (2003). *Statistics in musicology*. CRC Press.

[11] Bohak, C. and Marolt, M. (2016). Probabilistic segmentation of folk music recordings. *Mathematical Problems in Engineering*, 2016:11.

[12] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

[13] Bolton, R. J. and Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical science*, 17(3):235–249.

[14] Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of the International Conference on Computational Statistics (COMPSTAT)*, pages 177–186. Springer.

[15] Bountouridis, D., Brown, D. G., Wiering, F., and Veltkamp, R. C. (2017). Melodic similarity and applications using biologically-inspired techniques. *Applied Sciences*, 7(12):1242.

[16] Bradley, I. L. (1971). Repetition as a factor in the development of musical preferences. *Journal of Research in Music Education*, 19(3):295–298.

[17] Burton, A. M., Wilson, S., Cowan, M., and Bruce, V. (1999). Face recognition in poor-quality video: Evidence from security surveillance. *Psychological Science*, 10(3):243–248.

[18] Caballero, Á. Á. (2004). *El cante flamenco*. Alianza Editorial.

[19] Cabrera, J. J., Díaz-Báñez, J. M., Escobar-Borrego, F. J., Gómez, E., Gomez Martin, F., and Mora, J. (2008). Comparative melodic analysis of a cappella flamenco cantes. In *Proceedings of the 4th Conference on Interdisciplinary Musicology (CIM)*.

[20] Cai, W., Li, Q., and Guan, X. (2011). Automatic singer identification based on auditory features. In *Proceedings of the 7th International Conference on Natural Computation (ICNC)*, pages 1624–1628.

[21] Caraballo, L. E., Díaz-Báñez, J. M., and Pérez-Castillo, E. (2015). Finding unknown nodes in phylogenetic graphs. In *International Conference on Bioinformatics and Biomedical Engineering*, pages 403–414. Springer.

[22] Castro Buendía, G. (2015). *Génesis musical del Cante Flamenco*. Ed. Libros con Duende, S.L.

[23] Cerna, L., Cámara-Chávez, G., and Menotti, D. (2013). Face detection: Histogram of oriented gradients and bag of feature method. In *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, Las Vegas, USA.

[24] Chan, W. S. and Chin, F. (1996). Approximation of polygonal curves with minimum number of line segments or minimum error. *International Journal of Computational Geometry and Applications*, 6(01):59–77.

[25] Chew, E. (2000). *Towards a mathematical model of tonality*. PhD thesis, Massachusetts Institute of Technology.

[26] Choi, K., Fazekas, G., Sandler, M., and Cho, K. (2017). Convolutional recurrent neural networks for music classification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2392–2396.

[27] Chollet, F. (2016). Xception: Deep learning with depthwise separable convolutions. *arXiv preprint arXiv:1610.02357*.

[28] Clarke, E. and Cook, N. (2004). *Empirical musicology: Aims, methods, prospects*. Oxford University Press.

[29] Collins, T., Böck, S., Krebs, F., and Widmer, G. (2014). Bridging the audio-symbolic gap: The discovery of repeated note content directly from polyphonic music audio. In *Proceedings of the 53rd International Audio Engineering Society Conference on Semantic Audio*. Audio Engineering Society.

[30] Conklin, D. (2010). Discovery of distinctive patterns in music. *Intelligent Data Analysis*, 14(5):547–554.

[31] Cowdery, J. R. (1984). A fresh look at the concept of tune family. *Ethnomusicology*, 28(3):495–504.

[32] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Computer Society Conference on Vision and Pattern Recognition*, pages 886–893, San Diego, CA, USA.

[33] Dannenberg, R. B. and Hu, N. (2003). Pattern discovery techniques for music audio. *Journal of New Music Research*, 32(2):153–163.

[34] Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS)*, pages 2933–2941, Montreal, Canada.

[35] de Clercq, T. and Temperley, D. (2011). A corpus analysis of rock harmony. *Popular Music*, 30:47–70.

[36] De Poli, G., Rodà, A., and Vidolin, A. (1998). Note-by-note analysis of the influence of expressive intentions and musical structure in violin performance. *Journal of New Music Research*, 27(3):293–321.

[37] de Valk, R., Volk, A., Holzapfel, A., Pikrakis, A., Kroher, N., and Six, J. (2017). MIRchiving: Challenges and opportunities of connecting mir research and digital music archives. In *Proceedings of the 4rd International Workshop on Digital Libraries for Musicology*, pages 25–28, Shangai, China. ACM.

[38] Díaz-Báñez, J. (2017). Mathematics and flamenco: An unexpected partnership. *The Mathematical Intelligencer*, 39(3):27–39.

[39] Díaz-Báñez, J.-M. (2013). On math problems in the study of flamenco singing. *La Gaceta de la Real Sociedad Matemática Española*, 13(3):513–541.

[40] Díaz-Báñez, J.-M., Farigu, G., Gómez, F., Rappaport, D., and Toussaint, G. T. (2004). El compás flamenco: a phylogenetic analysis. In *Proceedings of BRIDGES: Mathematical Connections in Art, Music, and Science*, pages 61–70.

[41] Díaz-Bañez, J. M., Farigu, G., Toussaint, G., Gómez, F., and Rappaport, D. (2005). Similaridad y evolución en la rítmica del flamenco: una incursión de la matemática computacional. *Gaceta de la Real Sociedad Matemática Española*, 8(2):489–509.

[42] Díaz-Báñez, J. M., Korman, M., Pérez-Lantero, P., Pilz, A., Seara, C., and Silvera, R. I. (2015a). New results on stabbing segments with a polygon. *Computational Geometry: Theory and Applications*, 48(1):14–29.

[43] Díaz-Báñez, J. M., Kroher, N., and Rizo, J. C. (2015b). Efficient algorithms for melodic similarity in flamenco singing. In *Proceedings of the 5th International Workshop on Folk Music Analysis (FMA)*, pages 56–60, Paris, France.

[44] Díaz-Báñez, J. M. and Mesa, A. (2001). Fitting rectilinear polygonal curves to a set of points in the plane. *European Journal of Operational Research*, 130(1):214–222.

[45] Díaz-Báñez, J. M. and Rizo, J.-C. (2014). An efficient DTW-based approach for melodic similarity in flamenco singing. In *International Conference on Similarity Search and Applications*, pages 289–300. Springer.

[46] Dieleman, S. and Schrauwen, B. (2014). End-to-end learning for music audio. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6964–6968. IEEE.

[47] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

[48] Douglas, D. H. and Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122.

[49] Dowling, W. J. (1973). Rhythmic groups and subjective chunks in memory for melodies. *Attention, Perception and Psychophysics*, 14(1):37–40.

[50] Downie, J. S. (2003). Music information retrieval. *Annual review of information science and technology*, 37(1):295–340.

[51] Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7):2121–2159.

[52] Duda, R. O., Hart, P. E., and Stork, D. G. (2012). *Pattern classification*. John Wiley and Sons.

[53] Ellis, D. P. (2007). Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60.

[54] Ellis, D. P. and Poliner, G. E. (2007). Identifying cover songs with chroma features and dynamic programming beat tracking. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[55] Erhan, D., Bengio, Y., Courville, A., and Vincent, P. (2009). Visualizing higher-layer features of a deep network. Technical Report 1341, University of Montreal.

[56] Feng, D.-F. and Doolittle, R. F. (1987). Progressive sequence alignment as a prerequisite correct phylogenetic trees. *Journal of Molecular Evolution*, 25(4):351–360.

[57] Fernández, L. (2004). *Flamenco Music Theory: Rhythm, Harmony, Melody, Form*. Mel Bay Publications.

[58] Fernández-Marín, L. (2011). La bimodalidad en las formas del fandango y en los cantes de levante: origen y evolución. *La Madrugá*, 5(1):37–53.

[59] Flossmann, S., Grachten, M., and Widmer, G. (2009). Expressive performance rendering: Introducing performance context. In *Proceedings of the Sound and Music Computing Conference*, pages 155–160, Porto, Portugal.

[60] Fournier, H. and Vigneron, A. (2011). Fitting a step function to a point set. *Algorithmica*, 60(1):95–109.

[61] Frieler, K. (2014). Exploring phrase form structures: European folk songs. In *Proceedings of the International Workshop on Folk Music Analysis (FMA)*.

[62] Fujihara, H., Goto, M., Kitahara, T., and Okuno, H. G. (2010). A modeling of singing voice robust to accompaniment sounds and its application to singer identification and vocal-timbre-similarity-based music information retrieval. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):638–648.

[63] Gamboa, J. M. (2005). *Una historia del flamenco*. Espasa.

[64] Gamboa, J. M. and Núñez, F. (2007). *Flamenco de la A a la Z: diccionario de términos del flamenco*. Espasa Calpe Mexicana, SA.

[65] Garcia, C. and Delakis, M. (2004). Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 26(11):1408–1423.

[66] Ghias, A., Logan, J., Chamberlin, D., and Smith, B. C. (1995). Query by humming: musical information retrieval in an audio database. In *Proceedings of the 3rd ACM International Conference on Multimedia*, pages 231–236. ACM.

[67] Gómez, E. (2006). Tonal description of polyphonic audio for music content processing. *INFORMS Journal on Computing*, 18(3):294–304.

[68] Gómez, E. and 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.

[69] Gómez, E., Bonada, J., and Salamon, J. (2012a). Automatic transcription of flamenco singing from monophonic and polyphonic music recordings. In *Proceedings of the III Interdisciplinary Conference on Flamenco Research (INFLA)*, Sevilla, Spain.

[70] Gómez, E., Cañadas-Quesada, F. J., Salamon, J., Bonada, J., Vera-Candeas, P., and Molero, P. C. (2012b). Predominant fundamental frequency estimation vs singing voice separation for the automatic transcription of accompanied flamenco singing. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 601–606.

[71] Gómez, E., Herrera, P., and Gómez-Martin, F. (2013). Computational ethnomusicology: perspectives and challenges. *Journal of New Music Research*, 42(2):111–112.

[72] Gómez, F., Dıaz-Bánez, J., Gómez, E., and Mora, J. (2014). Flamenco music and its computational study. In *Proceedings of BRIDGES: Mathematical Connections in Art, Music, and Science*, pages 119–126.

[73] Gómez, F., Pikrakis, A., Mora, J., Dıaz-Bánez, J. M., Gómez, E., and Escobar, F. (2011). Automatic detection of ornamentation in flamenco. In *Proceedings of the 4th International Workshop on Machine Learning and Music (MML)*, Granada, Spain.

[74] Gomez-Martin, F., Taslakian, P., and Toussaint, G. (2009). Structural properties of euclidean rhythms. *Journal of Mathematics and Music*, 3(4):1–14.

[75] Granados, M. (2010). *Estilos y análisis musical del flamenco Vol. 1. Aplicado a la guitarra flamenca*. Maestro Fleamenco.

[76] Guastavino, C., Gomez, F., Toussaint, G., Marandola, F., and Gomez, E. (2009). Measuring similarity between flamenco rhythmic patterns. *Journal of New Music Research*, 38(2):129–138.

[77] Gulati, S., Bellur, A., Salamon, J., Ishwar, V., Murthy, H. A., and Serra, X. (2014a). Automatic tonic identification in indian art music: approaches and evaluation. *Journal of New Music Research*, 43(1):53–71.

[78] Gulati, S., Serra, J., Ishwar, V., and Serra, X. (2014b). Mining melodic patterns in large audio collections of indian art music. In *Proceedings of the 10th International Conference on Signal-Image Technology and Internet-Based Systems (SITIS)*, pages 264–271.

[79] Han, Y., Kim, J., Lee, K., Han, Y., Kim, J., and Lee, K. (2017). Deep convolutional neural networks for predominant instrument recognition in polyphonic music. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 25(1):208–221.

[80] Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.

[81] Hasdorff, L. (1976). *Gradient optimization and nonlinear control*. NASA Technical Reports.

[82] Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., et al. (2017). CNN architectures for large-scale audio classification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135, New Orleans, USA.

[83] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

[84] Hobby, J. D. (1993). Polygonal approximations that minimize the number of inpections. In *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, pages 93–102.

[85] Holzapfel, A. and Stylianou, Y. (2007). Singer identification in rembetiko music. In *Proceedings of the Sound and Music Computing Conference (SMC)*, pages 23–26.

[86] Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1):193–218.

[87] Imai, H. and Iri, M. (1986). Computational-geometric methods for polygonal approximations of a curve. *Computer Vision, Graphics, and Image Processing*, 36(1):31–41.

[88] Jacomy, M., Venturini, T., Heymann, S., and Bastian, M. (2014). ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PLoS ONE*, 9(6).

[89] Jafri, R. and Arabnia, H. R. (2009). A survey of face recognition techniques. *Journal of Information Processing Systems*, 5(2):41–68.

[90] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: A review. *ACM Computer Surveys*, 31(3).

[91] Janssen, B., De Haas, W. B., Volk, A., and van Kranenburg, P. (2013). Finding repeated patterns in music: State of knowledge, challenges, perspectives. In *Proceedings of the International Symposium on Computer Music Modeling and Retrieval*, pages 277–297. Springer.

[92] Kaufman, L. and Rousseeuw, P. J. (1987). *Statistical Data Analysis Based on the L1-Norm and Related Methods*, chapter Clustering by means of Medoids, pages 405–416. North-Holland.

[93] Kazemi, V. and Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

[94] Kim, Y. E. and Whitman, B. (2002). Singer identification in popular music recordings using voice coding features. In *Proceedings of the 3rd International Conference on Music Information Retrieval*, pages 13–17.

[95] King, D. E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758.

[96] Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR).*

[97] Kotsifakos, A., Papapetrou, P., Hollmén, J., Gunopulos, D., and Athitsos, V. (2012). A survey of query-by-humming similarity methods. In *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA).*

[98] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NIPS)*, pages 1097–1105, Lake Tahoe, USA.

[99] Kroher, N., Chaachoo, A., Díaz-Báñez, J., Gómez, E., Gómez-Martín, F., Mora, J., and Sordo, M. (2017). *Springer Handbook of Systematic Musicology*, chapter Computational ethnomusicology: A study of flamenco and Arab-Andalusian vocal music. Springer-Verlag, Berlin Heidelberg.

[100] Kroher, N. and Díaz-Báñez, J. M. (2016). Towards flamenco style recognition: The challenge of modelling the aficionado. In *Proceedings of the 6th International Workshop on Folk Music Analysis*, pages 14–17.

[101] Kroher, N. and Díaz-Báñez, J. M. (Submitted). Modelling melodic variation and extracting melodic templates from flamenco singing performances. *Journal of Mathematics and Music.*

[102] Kroher, N., Díaz-Báñez, J.-M., Mora, J., and Gómez, E. (2016). Corpus COFLA: a research corpus for the computational study of flamenco music. *Journal on Computing and Cultural Heritage (JOCCH)*, 9(2):10.

[103] Kroher, N. and Gómez, E. (2014). Automatic singer identification for improvisational styles based on vibrato, timbre and statistical performance descriptors. In *Sound and Music Computing Conference (SMC)*.

[104] Kroher, N. and Gómez, E. (2016). Automatic transcription of flamenco singing from polyphonic music recordings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(5):901–913.

[105] Kroher, N., Gómez, E., Guastavino, C., Gómez, F., and Bonada, J. (2014). Computational models for perceived melodic similarity in a cappella flamenco singing. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 65–70.

[106] Kroher, N., Pikrakis, A., Moreno, J., and Díaz-Báñez, J.-M. (2015). Discovery of repeated vocal patterns in polyphonic audio: A case study on flamenco music. In *Proceedings of the 23rd European Signal Processing Conference (EUSIPCO)*, pages 41–45.

[107] Krumhansl, C. L. and Shepard, R. N. (1979). Quantification of the hierarchy of tonal functions within a diatonic context. *Journal of experimental psychology: Human Perception and Performance*, 5(4):579–594.

[108] Kruskall, J. B. and Liberman, M. (1983). *Time warps, string edits and macromolecules*, chapter The symmetric time warping algorithm: From continuous to discrete. Addison.

[109] Kung, S. Y., Mak, M.-W., and Lin, S.-H. (2005). *Biometric authentication: a machine learning approach*. Prentice Hall Professional Technical Reference Upper Saddle River.

[110] Lagrange, M., Ozerov, A., and Vincent, E. (2012). Robust singer identification in polyphonic music using melody enhancement and uncertainty-based learning. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*.

[111] Lamont, A. and Dibben, N. (2001). Motivic structure and the perception of similarity. *Music Perception: An Interdisciplinary Journal*, 18(3):245–274.

[112] Laroche, J. (2003). Efficient tempo and beat tracking in audio recordings. *Journal of the Audio Engineering Society*, 51(4):226–233.

[113] Lawson, A., Vabishchevich, P., Huggins, M., Ardis, P., Battles, B., and Stauffer, A. (2011). Survey and evaluation of acoustic features for speaker recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5444–5447. IEEE.

[114] Learned-Miller, E., Huang, G. B., RoyChowdhury, A., Li, H., and Hua, G. (2016). *Advances in Face Detection and Facial Image Analysis*, chapter Labeled faces in the wild: A survey, pages 189–248. Springer.

[115] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

[116] LeCun, Y., Huang, F. J., and Bottou, L. (2004). Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, volume 2, pages 97–104. IEEE.

[117] Lerdahl, F. and Jackendoff, R. (1985). *A generative theory of tonal music*. MIT press.

[118] Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

[119] Li, H., Lin, Z., Shen, X., Brandt, J., and Hua, G. (2015). A convolutional neural network cascade for face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5325–5334, Boston, USA.

[120] Liu, C.-C. (2013). Towards automatic music performance comparison with the multiple sequence alignment technique. In *Proceedings of the International Conference on Multimedia Modeling*, pages 391–402. Springer.

[121] Lomax, A. (1968). *Folk song style and culture*. Routledge.

[122] Löytynoja, A. and Goldman, N. (2005). An algorithm for progressive multiple alignment of sequences with insertions. *Proceedings of the National Academy of Sciences of the United States of America*, 102(30):10557–10562.

[123] Manuel, P. (1986). Evolution and structure in flamenco harmony. *Current Musicology*, 42:46–57.

[124] Margulis, E. H. (2012). Musical repetition detection across multiple exposures. *Music Perception: An Interdisciplinary Journal*, 29(4):377–385.

[125] Marqués, I. (2017). *Flamenco as a vehicle of popular religiosity.* PhD thesis, University of Seville.

[126] Marqués, I., Kroher, N., Mora, J., and Díaz-Báñez, J. M. (2017). Extraction and classification of ornamentation in flamenco singing: An evolution-based approach. In *Proceedings of the 7th International Workshop on Folk Music Analysis*.

[127] Mauch, M., Cannam, C., Bittner, R., Fazekas, G., Salamon, J., Dai, J., Bello, J., and Dixon, S. (2015). Computer-aided melody note transcription using the tony software: Accuracy and efficiency. In *Proceedings of the First International Conference on Technologies for Music Notation and Representation*.

[128] Mauch, M. and Dixon, S. (2014). pYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[129] McCullough, L. E. (1977). Style in traditional irish music. *Ethnomusicology*, 21(1):85–97.

[130] McNab, R. J., Smith, L. A., Witten, I. H., Henderson, C. L., and Cunningham, S. J. (1996). Towards the digital music library: Tune retrieval from acoustic input. In *Proceedings of the First ACM International Conference on Digital Libraries*, pages 11–18.

[131] Mesaros, A., Virtanen, T., and 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)*, pages 375–378.

[132] Mongeau, M. and Sankoff, D. (1990). Comparison of musical sequences. *Computers and the Humanities*, 24(3):161–175.

[133] Moon, H. (2004). Biometrics person authentication using projection-based face recognition system in verification scenario. In *Biometric Authentication*, pages 207–213. Springer.

[134] Mora, J., Gómez, F., Escobar-Borrego, F., and Díaz-Báñez, J. M. (2010a). Melodic characterization and similarity in a cappella flamenco cantes. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*.

[135] Mora, J., Gómez, F., Gómez, E., and Díaz-Báñez, J. M. (2016). Melodic contour and mid-level global features applied to the analysis of flamenco cantes. *Journal of New Music Research*, 45(2):145–159.

[136] Mora, J., Gomez Martin, F., Gómez, E., Escobar-Borrego, F. J., and Díaz-Báñez, J. M. (2010b). Characterization and melodic similarity of a cappella flamenco cantes. In *International Society for Music Information Retrieval Conference (ISMIR)*.

[137] Müllensiefen, D. and Frieler, K. (2006). *Data Science and Classification*, chapter Evaluating different approaches to measuring the similarity of melodies, pages 299–306. Springer.

[138] Müller, M. (2015). *Fundamentals of Music Processing*, chapter Music Structure Analysis. Springer, Cham.

[139] Müller, M. and Grosche, P. (2012). Automated segmentation of folk song field recordings. In *Proceedings of the 10th ITG Speech Communication Symposium*.

[140] Müller, M., Grosche, P., and Wiering, F. (2009). Robust segmentation and annotation of folk song recordings. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pages 735–740.

[141] Müller, M., Jiang, N., and Grosche, P. (2013). A robust fitness measure for capturing repetitions in music recordings with applications to audio thumbnailing. *IEEE Transactions on Audo, Speech, and Language Processing*, 21(3):531–543.

[142] Nasrabadi, N. M. (2007). Pattern recognition and machine learning. *Journal of electronic imaging*, 16(4):049901.

[143] Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.

[144] Ng, A., Jordan, M., and Weiss, Y. (2002). On spectral clustering: analysis and an algorithm. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 849–856.

[145] Ng, H.-W. and Winkler, S. (2014). A data-driven approach to cleaning large face datasets. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*.

[146] Nieto, O. and Farbood, M. M. (2014). Identifying polyphonic patterns from audio recordings using music segmentation techniques. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*.

[147] Novello, A., McKinney, M. F., and Kohlrausch, A. (2006). Perceptual evaluation of music similarity. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 246–249.

[148] Nwe, T. L. and Li, H. (2008). On fusion of timbre-motivated features for singing voice detection and singer identification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2225–2228.

[149] Oramas, S., Gómez, F., Gómez, E., and Mora, J. (2015). Flabase: Towards the creation of a flamenco music knowledge base. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 378–384.

[150] Orio, N. et al. (2006). Music retrieval: A tutorial and review. *Foundations and Trends in Information Retrieval*, 1(1):1–90.

[151] Panteli, M., Bittner, R., Bello, J. P., and Dixon, S. (2017). Towards the characterization of singing styles in world music. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 636–640, New Orleans, USA.

[152] Papapavlou, M. (2003). The city as a stage: Flamenco in andalusian culture. *Journal of the Society for the Anthropology of Europe*, 3(2):14–24.

[153] Pazzani, M. J. and Billsus, D. (2007). *The adaptive web*, chapter Content-based recommendation systems, pages 325–341. Springer.

[154] Peeters, G. (2006). Chroma-based estimation of musical key from audio-signal analysis. In *International Society for Music Information Retrieval Conference (ISMIR)*, pages 115–120.

[155] Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, 4(2):1883.

[156] Petrick, J. F. (2002). Development of a multi-dimensional scale for measuring the perceived value of a service. *Journal of leisure research*, 34(2):119–134.

[157] Pikrakis, A., Gómez, F., Oramas, S., Díaz-Báñez, J. M., Mora, J., Escobar-Borrego, F., Gómez, E., and Salamon, J. (2012). Tracking melodic patterns in flamenco singing by analyzing polyphonic music recordings. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 421–426.

[158] Pikrakis, A., Kopsinis, Y., Kroher, N., and Díaz-Báñez, J. M. (2016). Unsupervised singing voice detection using dictionary learning. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pages 1212–1216, Budapest, Hungary.

[159] Pikrakis, A., Theodoridis, S., and Kamarotos, D. (2003). Recognition of isolated musical patterns using context dependent dynamic time warping. *IEEE Transactions on Speech and Audio Processing*, 11(3):175–183.

[160] Poliner, G. E., Ellis, D. P. W., Ehrmann, A. F., Gómez, E., Streich, S., Pliner, G. E., Ellis, D. P. W., Ehrmann, A. F., Gómez, E., Streich, S., and Ong, B. (2007). Melody transcription from music audio: Approaches and evaluation. *IEEE Transactions on Audio, Speech and Language Processing*, 15(4):1247–1256.

[161] Rabiner, L. R. and Juang, B.-H. (1993). *Fundamentals of speech recognition*. PTR Prentice Hall.

[162] Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

[163] Ranjani, H., Paramashivan, D., and Sreenivas, T. V. (2017). Quantized melodic contours in indian art music perception: Application to transcription. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pages 174–180, Suzhou, China.

[164] Rao, P., Ross, J. C., Ganguli, K. K., Pandit, V., Ishwar, V., Bellur, A., and Murthy, H. A. (2014). Classification of melodic motifs in raga music with time-series matching. *Journal of New Music Research*, 43(1):115–131.

[165] Ratanamahatana, C. A. and Keogh, E. (2004). Everything you know about dynamic time warping is wrong. In *Third Workshop on Mining Temporal and Sequential Data*, pages 22–25, Seattle, USA.

[166] Rodríguez-López, M. and Volk, A. (2013). Symbolic segmentation: a corpus-based analysis of melodic phrases. In *Proceedings of the International Symposium on Computer Music Modeling and Retrieval*, pages 548–557. Springer.

[167] Rodríguez-López, M. E. and Volk, A. (2015). Location constraints for repetition-based segmentation of melodies. In *Proceedings of the International Conference on Mathematics and Computation in Music*, pages 73–84. Springer International Publishing.

[168] Rogers, G. L. (2004). Interdisciplinary lessons in musical acoustics: The science-math-music connection. *Music Educators Journal*, 91(1):25–30.

[169] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Computational and Applied Mathematics*, 20:53–65.

[170] Rubner, Y., Tomasi, C., and Guibas, L. J. (1998). A metric for distributions with applications to image databases. In *Proceedings of the 6th International Conference on Computer Vision*, pages 59–66.

[171] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323:533–536.

[172] Ryynanen, M. and Klapuri, A. (2008). Query by humming of midi and audio using locality sensitive hashing. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2249–2252. IEEE.

[173] Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processingransactions on acoustics, speech, and signal processing*, 26(1):43–49.

[174] Salamon, J. and Gómez, E. (2014). Melody extraction from polyphonic music signals: Approaches, applications and challenges. *IEEE Signal Processing Magazine*, 31(2):118–134.

[175] Salamon, J., Gómez, E., Ellis, D. P., and Richard, G. (2012a). Melody extraction from polyphonic music using pitch contour characteristics. *IEEE Transactions on Audio, Speech and Language Processing*, 20(6):1759–1770.

[176] Salamon, J., Rocha, B., and 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)*, Kyoto, Japan.

[177] Salamon, J., Serra, J., and 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.

[178] Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review*, 1(1):27–64.

[179] Schluter, J. and Bock, S. (2014). Improved musical onset detection with convolutional neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6979–6983. IEEE.

[180] Schlüter, J. and 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)*, pages 121–126.

[181] Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

[182] Serra, X. (2011). A multicultural approach in music information research. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR).*

[183] Shen, J., Shepherd, J., Cui, B., and Tan, K.-L. (2009). A novel framework for efficient automated singer identification in large music databases. *ACM Transactions on Information Systems (TOIS)*, 27(3):18.

[184] Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations (ICLR).*

[185] Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR).*

[186] Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197.

[187] Song, J., Bae, S.-Y., and Yoon, K. (2002). Query by humming: Matching humming query to polyphonic audio. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 329–332. IEEE.

[188] Song, L., Li, M., and Yan, Y. (2013). Automatic vocal segments detection in popular music. In *Proceedings of the Ninth International Conference on Computational Intelligence and Security.*

[189] Spera, C. C. (2010). Flamenco nuevo: Tradition, evolution and innovation. Master's thesis, University of Southern California.

[190] Sra, S., Nowozin, S., and Wright, S. J. (2012). *Optimization for machine learning.* MIT Press.

[191] Sridhar, R. and Geetha, T. V. (2008). Music information retrieval of carnatic songs based on carnatic music singer identification. In *Proceedings of the International Conference on Computer and Electrical Engineering*, pages 407–411.

[192] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., et al. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

[193] Tamayo, M. C. and Cano, E. R. (1986). *La guitarra: historia, estudios y aportaciones al arte flamenco*. Consejo Superior de Investigaciones Científicas, Diputación Provincial de Teruel.

[194] Taminau, J., Hillewaere, R., Meganck, S., Conklin, D., Nowé, A., and Manderick, B. (2009). Descriptive subgroup mining of folk music. In *Proceedings of the 2nd International Workshop on Machine Learning and Music (MML)*, pages 1–6, Bled, Slovenia.

[195] Tatusova, T. A. and Madden, T. L. (1999). Blast 2 sequences, a new tool for comparing protein and nucleotide sequences. *FEMS microbiology letters*, 174(2):247–250.

[196] Temperley, D. (2004). *The cognition of basic musical structures*. MIT press.

[197] Temperley, D. and Marvin, E. W. (2008). Pitch-class distribution and the identification of key. *Music Perception: An Interdisciplinary Journal*, 25(3):193–212.

[198] Thompson, B. (1985). Flamenco: A tradition in evolution. *The World of Music*, 27(3):67–80.

[199] Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673–4680.

[200] Thrasher, A. R. (1985). The melodic structure of jiangnan sizhu. *Ethnomusicology*, 29(2):237–263.

[201] Toussaint, G. (2010). Computational geometric aspects of rhythm, melody, and voice-leading. *Computational Geometry*, 43(1):2–22.

[202] Tsai, W.-H. and Lee, H.-C. (2012). Automatic singer identification based on speech-derived models. *International Journal of Future Computer and Communication*, 1(2):94–96.

[203] Turnbull, D., Lanckriet, G., Pampalk, E., and Goto, M. (2007). A supervised approach for detecting boundaries in music using difference features and boosting. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*.

[204] Typke, R., Giannopoulos, P., Veltkamp, R. C., Wiering, F., Van Oostrum, R., et al. (2003). Using transportation distances for measuring melodic similarity. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*.

[205] Typke, R., Wiering, F., and Veltkamp, R. C. (2007). Transportation distances and human perception of melodic similarity. *Musicae Scientiae*, 11(1):153–181.

[206] Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302.

[207] Ullrich, K., Schlüter, J., and Grill, T. (2014). Boundary detection in music structure analysis using convolutional neural networks. In *Proceedings of the International Society of Music Information Retrieval Conference (ISMIR)*, pages 417–422, Taipei, Taiwan.

[208] Urbano, J., Lloréns, J., Morato, J., and Sánchez-Cuadrado, S. (2011). Melodic Similarity through Shape Similarity. In Ystad, S., Aramaki, M., Kronland-Martinet, R., and Jensen, K., editors, *Exploring Music Contents*, pages 338–355. Springer.

[209] Van den Oord, A., Dieleman, S., and Schrauwen, B. (2013). Deep content-based music recommendation. In *Proceedings of Advances in neural information processing systems (NIPS)*, pages 2643–2651, Lake Tahoe, USA.

[210] van Kranenburg, P. (2010). *A computational approach to content-based retrieval of folk song melodies*. PhD thesis, Utrecht University.

[211] van Kranenburg, P., Biró, D. P., Ness, S. R., and Tzanetakis, G. (2011a). A computational investigation of melodic contour stability in jewish torah trope performance traditions. In *Proceedings of the International Society of Music Information Retrieval Conference (ISMIR)*, pages 163–168, Miami, USA.

[212] van Kranenburg, P., de Bruin, M., Grijp, L. P., and Wiering, F. (2014). The meertens tune collections. Technical Report 2014-1, Meertens Online Reports.

[213] van Kranenburg, P., Garbers, J., Volk, A., Wiering, F., Grijp, L. P., and Veltkamp, R. C. (2010). Collaboration perspectives for folk song research and music information retrieval: The indispensable role of computational musicology. *Journal of Interdisciplinary Music Studies*, 4(1):17–43.

[214] van Kranenburg, P. and Tzanetakis, G. (2010). A computational approach to the modeling and employment of cognitive units of folk song melodies using audio recordings. In *Proceedings of the 11th International Conference on Music Perception and Cognition*, pages 794–797.

[215] van Kranenburg, P., Volk, A., and Wiering, F. (2011b). On operationalizing the musicological concept of tune family for computational modeling. In *Proceedings of Supporting Digital Humanities: Answering the unaskable*, Copenhagen, Denmark.

[216] van Kranenburg, P., Volk, A., and Wiering, F. (2013). A comparison between global and local features for computational classification of folk song melodies. *Journal of New Music Research*, 42(1):1–18.

[217] Vaughn, K. (2000). Music and mathematics: Modest support for the oft-claimed relationship. *Journal of aesthetic education*, 34(3/4):149–166.

[218] Velardo, V., Vallati, M., and Jan, S. (2016). Symbolic melodic similarity: State of the art and future challenges. *Computer Music Journal*, 40(2):70–83.

[219] Volk, A. and de Haas, W. B. (2013). A corpus-based study on ragtime syncopation. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 163–168, Curitiba, Brazil.

[220] Volk, A. and Honingh, A. (2012). Mathematical and computational approaches to music: challenges in an interdisciplinary enterprise. *Journal of Mathematics and Music*, 6(2):73–81.

[221] Volk, A. and van Kranenburg, P. (2012). Melodic similarity among folk songs: An annotation study on similarity-based categorization in music. *Musicae Scientiae*, 16(3):317–339.

[222] Walton, C. W. (2005). *Basic forms in music*. Alfred Music.

[223] Wang, C. and Dubnov, S. (2015). Pattern discovery from audio recordings by variable markov oracle: A music information dynamics approach. In *Proceedings of the IEEE Conference on Audio, Speech and Signal Processing (ICASSP)*.

[224] Wang, S., Ewert, S., and Dixon, S. (2016). Robust and efficient joint alignment of multiple musical performances. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(11):2132–2145.

[225] Washabaugh, W. (1995). Ironies in the history of flamenco. *Theory, Culture and Society*, 12(1):133–155.

[226] Weiss, R. J. and Bello, J. P. (2010). Identifying repeated patterns in music using sparse convolutive non-negative matrix factorization. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*.

[227] White, J. D. (1994). *Comprehensive musical analysis*. Scarecrow Press.

[228] Widmer, G. and Goebl, W. (2004). Computational models of expressive music performance: The state of the art. *Journal of New Music Research*, 33(3):203–216.

[229] Worms, C. (2010). *Desde la guitarra, armonía del flamenco vol 1-3*. Acordes concert.

[230] Wright, D. (2009). *Mathematics and music*, volume 28. American Mathematical Soc.

[231] Xenakis, I. (1971). *Formalized Music: Thought and Mathematics in Composition*. Indiana University Press.

[232] Yi, D., Lei, Z., Liao, S., and Li, S. Z. (2014). Learning face representation from scratch. *arXiv preprint:1411.7923*.

[233] Zeiler, M. D. (2012). ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

[234] Zhang, C. and Zhang, Z. (2010). A survey of recent advances in face detection. Technical Report MSR-TR-2010-66, Microsoft Research.

[235] Zhang, T. (2003). Automatic singer identification. In *Proceedings of the International Conference on Multimedia and Expo (ICME)*, Baltimore, MD, USA. IEEE.

[236] Zhao, W., Chellappa, R., Phillips, P. J., and Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM computing surveys (CSUR)*, 35(4):399–458.

[237] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929. IEEE.