Università
degli Studi
di Padova

DEI
Dipartimento di
Ingegneria dell'Informazione

Scuola Di Dottorato In Ingegneria Dell'Informazione
Indirizzo In Scienza E Tecnologia Dell'Informazione, Ciclo: XXIV

ALIGNMENT AND IDENTIFICATION OF MULTIMEDIA DATA: APPLICATION TO MUSIC AND GESTURE PROCESSING

Direttore della Scuola
Chiar.^{mo} Prof. Matteo Bertocco

Supervisore
Chiar.^{mo} Prof. Nicola Orio

Dottorando
NICOLA MONTECCHIO

31 Gennaio 2012

ABSTRACT

The overwhelming availability of large multimedia collections poses increasingly challenging research problems regarding the organization of, and access to data. A general consensus has been reached in the Information Retrieval community, asserting the need for tools that move past metadata-based techniques and exploit directly the information contained in the media. At the same time, interaction with content has evolved beyond the traditional passive enjoyment paradigm, bringing forth the demand for advanced control and manipulation options.

The aim of this thesis is to investigate techniques for multimedia data alignment and identification. In particular, music audio streams and gesture-capture time series are considered. Special attention is given to the efficiency of the proposed approaches, namely the realtime applicability of alignment algorithms and the scalability of identification strategies.

The concept of alignment refers to the identification and matching of corresponding substructures in related entities. The focus of this thesis is directed towards alignment of sequences with respect to a single dimension, aiming at the identification and matching of significant events in related time series.

The alignment of audio recordings of music to their symbolic representations serves as a starting point to explore different methodologies based on statistical models. A unified model for the real time alignment of music audio streams to both symbolic scores and audio references is proposed. Its advantages are twofold: unlike most state-of-the-art systems, tempo is an explicit parameter within the stochastic framework; moreover, both alignment problems can be formulated within a common framework by exploiting a continuous representation of the reference content. A novel application of audio alignment techniques was found in the domain of studio recording productions, reducing the human effort spent in manual repetitive tasks.

Gesture alignment is closely related to the domain of music alignment, as the artistic aims and engineering solutions of both areas largely overlap. Expressivity in gesture performance can be characterized by both the choice of a particular gesture and the way the gesture is executed. The former aspect involves a gesture recognition task, while the latter is addressed considering the time-evolution of features and the way these differ from pre-recorded templates. A model, closely related to the mentioned music alignment strategy, is proposed, capable of simultaneously recognizing a gesture among many templates and aligning it against the correct reference in realtime, while jointly estimating signal feature such as rotation, scaling, velocity.

Due to the increasingly large volume of music collections, the organization of media items according to their perceptual characteristics has become of fundamental importance. In particular, content-based identification technologies provide the tools to retrieve and organize music documents. Music identification techniques should ideally be able to identify a recording – by comparing it against a set of known recordings – independently from the particular performance, even in case of significantly different arrangements and interpretations.

Even though alignment techniques play a central role in many works of the music identification literature, the proposed methodology addresses the task using techniques that are usually associated to textual IR. Similarity computation is based on hashing, attempting at creating collisions between vectors that are close in the feature space. The resulting compactness of the representation of audio content allows index-based retrieval strategies to be exploited for maximizing computational efficiency.

A particular application is considered, regarding Cultural Heritage preservation institutions. A methodology is proposed to automatically identify recordings in collections of digitized tapes and vinyl discs. This scenario differs significantly from that of a typical identification task, as a query most often contains more than one relevant result (distinct music work). The audio alignment methodology mentioned above is finally exploited to carry out a precise segmentation of recordings into their individual tracks.

SOMMARIO

La crescente disponibilità di grandi collezioni multimediali porta all'attenzione problemi di ricerca sempre più complessi in materia di organizzazione e accesso ai dati. Nell'ambito della comunità dell'Information Retrieval è stato raggiunto un consenso generale nel ritenere indispensabili nuovi strumenti di reperimento in grado di superare i limiti delle metodologie basate su meta-dati, sfruttando direttamente l'informazione che risiede nel contenuto multimediale.

Lo scopo di questa tesi è lo sviluppo di tecniche per l'allineamento e l'identificazione di contenuti multimediali; la trattazione si focalizza su flussi audio musicali e sequenze numeriche registrate tramite dispositivi di cattura del movimento. Una speciale attenzione è dedicata all'efficienza degli approcci proposti, in particolare per quanto riguarda l'applicabilità in tempo reale degli algoritmi di allineamento e la scalabilità delle metodologie di identificazione.

L'allineamento di entità comparabili si riferisce al processo di aggiustamento di caratteristiche strutturali allo scopo di permettere una comparazione diretta tra elementi costitutivi corrispondenti. Questa tesi si concentra sull'allineamento di sequenze rispettivamente ad una sola dimensione, con l'obiettivo di identificare e confrontare eventi significativi in sequenze temporali collegate.

L'allineamento di registrazioni musicali alla loro rappresentazione simbolica è il punto di partenza adottato per esplorare differenti metodologie basate su modelli statistici. Si propone un modello unificato per l'allineamento in tempo reale di flussi musicali a partiture simboliche e registrazioni audio. I principali vantaggi sono collegati alla trattazione esplicita del tempo (velocità di esecuzione musicale) nell'architettura del modello statistico; inoltre, ambedue i problemi di allineamento sono formulati sfruttando una rappresentazione continua della dimensione temporale. Un'innovativa applicazione delle tecnologie di allineamento audio è proposta nel contesto della produzione di registrazioni musicali, dove l'intervento umano in attività ripetitive è drasticamente ridotto.

L'allineamento di movimenti gestuali è strettamente correlato al contesto dell'allineamento musicale, in quanto gli obiettivi artistici e le soluzioni ingegneristiche delle due aree sono largamente coincidenti. L'espressività di un'esecuzione gestuale è caratterizzata simultaneamente dalla scelta del particolare gesto e dal modo di eseguirlo. Il primo aspetto è collegato ad un problema di riconoscimento, mentre il secondo è affrontato considerando l'evoluzione temporale delle caratteristiche del segnale ed il modo in cui queste differiscono da template pre-registrati. Si propone un modello, strettamente legato alla controparte musicale sopra citata, capace di riconoscere un gesto in tempo reale tra una libreria di templates, simultaneamente allineandolo mentre caratteristiche del segnale come rotazione, dimensionamento e velocità sono congiuntamente stimate.

Il drastico incremento delle dimensioni delle collezioni musicali ha portato all'attenzione il problema dell'organizzazione di contenuti multimediali secondo caratteristiche percettive. In particolare, le tecnologie di identificazione basate sul contenuto forniscono strumenti appropriati per reperire e organizzare documenti musicali. Queste tecnologie dovrebbero idealmente essere in grado di identificare una registrazione – attraverso il confronto con un insieme di registrazioni conosciute – indipendentemente dalla particolare esecuzione, anche in caso di arrangiamenti o interpretazioni significativamente differenti.

Sebbene le tecniche di allineamento assumano un ruolo centrale in letteratura, la metodologia proposta sfrutta strategie solitamente associate al reperimento di informazione testuale. Il calcolo della similarità musicale è basato su tecniche di hashing per creare collisioni fra vettori prossimi nello spazio. La compattezza della risultante rappresentazione del contenuto acustico permette l'utilizzo di tecniche di reperimento basate su indicizzazione, allo scopo di massimizzare l'efficienza computazionale.

Un'applicazione in particolare è considerata nell'ambito della preservazione dei Beni Culturali, per l'identificazione automatica di collezioni di nastri e dischi in vinile digitalizzati. In questo contesto un supporto generalmente contiene più di un'opera rilevante. La metodologia di allineamento audio citata sopra è infine utilizzata per segmentare registrazioni in tracce individuali.

CONTENTS

1 Introduction 1

2 Alignment 5

2.1 Properties of Music Signals . . . 7
2.2 Dynamic Programming-based Systems . . . 10
2.3 Probabilistic Discrete State Systems . . . 12
2.4 Probabilistic Continuous State Systems . . . 21

3 Identification 29

3.1 Near-Duplicate Recording Detection . . . 30
3.2 Identification of Versions of Music Works . . . 30
3.3 An Efficient Music Identification Engine . . . 32

4 Applications and Experimental Evaluation 41

4.1 Real-Time Musical Interactions . . . 42
4.2 Offline Alignment for Studio Recording Productions . . . 48
4.3 Real-Time Gesture Recognition and Tracking . . . 54
4.4 Improved Access to Music Heritage Collections . . . 62

5 Conclusions and Future Developments 71

LIST OF FIGURES

2.1 Alignment of music data. 6
2.2 Alignment of gesture data. 6
2.3 Attack-Decay-Sustain-Release model. 7
2.4 Spectral modeling of music signals. 9
2.5 Chroma vector descriptors. 9
2.6 Model of symbolic scores in Hidden Markov Model-based systems. 16
2.7 Refined score graph topology with explicit error modeling. 18
2.8 Frequency dependent bandwidth/delay trade off for digital filters. 19
2.9 Discrete filter bank with delay compensation. 19
2.10 Hidden Markov Model for alignment of one-dimensional gesture signals. 20
2.11 Discrete and continuous representations of symbolic scores. 24
2.12 Translation of gestures captured in different coordinate systems. 28

3.1 Architecture of the proposed music identification methodology. 33
3.2 Methodology for the estimation of the reference tuning frequency. 35
3.3 Segmentation and indexing of hashed chroma descriptors. 38
3.4 Identification of multiple works contained within a long query. 39

4.1 Alignment of a synthetic music track. 43
4.2 Uncertainty in tempo estimation. 44
4.3 Handling optional repetitions within the Particle Filtering scheme. 45
4.4 Alignment of a music performance with optional repetitions. 46
4.5 Example of an open-form score. 46
4.6 Patches in the Max/MSP and Pure Data visual programming environments. 47
4.7 A mixing session in a Digital Audio Workstation. 48
4.8 Alignment methodology in the case of unknown starting position. 50
4.9 Evolution of the position distribution, in case of uniform prior. 51
4.10 Identification of the interest region. 51
4.11 Viviani's curve. 55
4.12 Alignment error on synthetic gesture data. 56
4.13 Alignment error for a synthetic gesture with dynamic rotation and noise. 57
4.14 Dynamic estimation of gesture rotation. 57
4.15 Transformed and reconstructed Viviani's curves. 58
4.16 Two-dimensional gesture vocabulary used for identification. 59
4.17 Three-dimensional gesture vocabulary used for identification. 59
4.18 Different orientations of the capture interface. 60
4.19 A three-dimensional gesture captured with different device orientations. 60

4.20 Segmentation of gesture sequences. 61
4.21 Inconsistent metadata associated to an LP. 63
4.22 Overview of an efficient two-step identification methodology. 64
4.23 Alignment classification strategy. 65
4.24 Distribution of cover set cardinalities for the MusiCLEF collection. 66
4.25 Estimated distribution of the tuning frequencies for the MusiCLEF collection. . . 67
4.26 Effect of the alignment-based reranking phase on accuracy. 70

LIST OF TABLES

1.1 Popular music services, as of Dec. 2011. 3

4.1 Music alignment accuracy – synthetic data set. 43
4.2 Music alignment accuracy – Chopin's Mazurkas. 44
4.3 Collections used for evaluating the alignment of takes to a run-through. 53
4.4 Recognition rates for two-dimensional gestures. 58
4.5 Recognition rates for three-dimensional gestures. 61
4.6 Effect of different parametrizations on music identification accuracy. 69
4.7 Accuracy of the proposed music identification system w.r.t. MIREX campaigns. . 69
4.8 Effect of different parametrizations on LP identification accuracy. 70

CHAPTER ONE

INTRODUCTION

Music as an art form involves several facets. The most universally accessible aspect of music is arguably that of acoustic rendition: a *performance* can be directly enjoyed by listeners, even those not possessing formal music training, and can be easily recorded and distributed. Such physical aspect is however just the last step of a complex creative task: in particular, in the context of western music, composition is traditionally perceived as a process where the artist conceives a music work as an organized structure of (mostly pitched) sounds that can be expressed through symbolic representation (*notation*). Moreover, even though music can be analyzed in terms of pitch, melody, rhythm, harmony and timbre – all aspects that can be directly related to an abstract model or inferred from the analysis of an aural rendition – *physical expressions* that are related to the musical stimulus can also be measured and studied.

This thesis is concerned with the formulation and the application of techniques for processing multimedia content, in particular as related to music; the aspects delineated above, in their *audio, score* and *gesture* realizations, are the focus of the analysis.

This chapter discusses the applicative contexts envisaged by current research on multimedia content processing, and the advantages over traditional metadata based information access.

Accessing Music Collections

In recent years the increased computational resources and bandwidth at the disposal of users have brought about a radical transformation of the Internet. The Web, originally a set of static, hyper-linked pages made up for the most part of text and still images, has evolved into a dynamic, media-rich environment. As Content Management Systems enabled users to express themselves using tools such as *wikis* and *blogs*, absolving people from learning the technical skills needed to compose content and make it available, in a similar way file sharing communities allowed people to exchange data without the inconvenience of managing physical carriers or dealing explicitly with file transfer protocols. The initial hype surrounding peer-to-peer file sharing – a network paradigm very popular in the early 2000s, in which users host local copies of the shared data on their own machines – was gradually superseded by the proliferation of third-part hosted services. Along these lines, the latest trends are represented by the phenomena of media (especially video) sharing websites: data collections are constantly growing in number and size, and it is now possible to assemble amounts of media that were previously inimaginable.

The sheer amount of available data is by itself a valid reason for pushing forward research aimed at analyzing the latent information residing in the content: as Table 1.1 shows, many popular subscription services either offer unlimited streaming access to vast catalogues of music for a monthly fee or allow the user to download individually purchased tracks/albums; other services are based on user sharing of data, and some of them allow artists to sell their music. In addition to those, a notable case is that of popular video sharing websites which, even if not strictly music-related, contain large quantities of music videos or user-uploaded content containing music soundtracks.

Automatic *identification* of music documents is becoming an essential component of many such services. For instance, the video sharing platform YouTube[1] automatically identifies the background music in user-contributed videos, in order to either remove the video (because of copyright infringement reasons) or suggest where the original music can be purchased; similarly, identification techniques are used to remove duplicates inside digital music archives when metadata information is not sufficient (as in the case of multiple copies of a work catalogued in different languages).

Most commercial technologies aim at identifying a particular *recording* rather than a given music *work*: alternative takes of a composition are not recognized as different instances of the same music work. Such techniques are designed to be particularly *efficient*, exploiting several assumptions on the possible differences between copies: these differences are usually due to audio processing techniques such as lossy compression, background noise, or transfer issues from analog carriers that can result in time stretching and pitch shifting artifacts.

Identification of different performances of the same work cannot take advantage from such assumptions. Alternative recordings of a work by different artists, as well as live versions of a studio recording, may vary greatly in several aspects, such as tempo, key, instrumentation; they may also be characterized by a radically different musical structure.

The additional freedom of interpretation of the underlying music ideas is reflected in a significantly more complex identification task. The notion of music *similarity* must be explicitly taken into account, even though it intrinsically involves human judgements that often disagree and are consequently difficult to formalize. Besides the inherent difficulties in a theoretical modeling of the issues described above, an unfortunate consequence of complex theoretical approaches is often the heavy computational requirements of their software implementations. A primary objective of this thesis is the investigation of the tradeoff between effectiveness and efficiency in music identification systems.

[1] `http://www.youtube.com`

	launch	size	C	U	S	P	F	url
Pandora	2000/01	800k	♪		♪			http://www.pandora.com
Rhapsody	2001/12	13M	♪		♪	♪		http://www.rhapsody.com
iTunes	2003/04	18M	♪			♪		http://www.apple.com/itunes
Jamendo	2005/01	350k		♪			♪	http://www.jamendo.com
Deezer	2007/08	13M	♪		♪	♪		http://www.deezer.com
Amazon MP3	2007/09	17M	♪			♪		http://www.amazon.com
Spotify	2008/10	17M	♪		♪	♪		http://www.spotify.com
SoundCloud	2008/10			♪			♪	http://www.soundcloud.com
MOG	2009/12	11M	♪		♪	♪		http://www.mog.com
Rdio	2010/08	12M	♪		♪	♪		http://www.rdio.com

Table 1.1: Popular music services, as of Dec. 2011. Legend:
C "commercial content", U "user-generated content";
S "paid subscription (streaming)", P "track purchase", F "free (streaming)".

Multimedia Content Processing

Even though the access to media of various types is nowadays almost immediate, it is still unclear how to optimally exploit the retrieved content: a traditional, *passive* form of enjoyment is still the primary form of media consumption, however there is an increasing demand for *interaction* with the content. This is particularly true in the domain of experimental arts, characterized by a strong focus in the *involvement* of users, as opposed to a more traditional context in which the audience is often relegated to the role of passive witness. The exploitation of media content is also a central aspect of other domains, such as musicological analysis, where a strong interest lies not only in the retrieval of content, but also on its analysis and comparison with other related material.

The pursuit of complex interaction with content has given rise to a demand for advanced control and manipulation options. In this context, a central concept is that of *alignment*, that is of correspondence between two alternative realizations of a same idea. Even though the identification and alignment problems are in this sense closely related, they aim at answering different questions, namely *whether* two signals are representations of the same idea and *how* these representations are related: an alignment between two signals provides a match between each point in the sequences.

Research on software systems capable of performing sequence alignment in real time is of significant importance in many domains. In the context of music, the interaction of computers with live (human) performers has been the subject of research aimed at enabling responsive automatic accompaniment. Moreover, performance of electronic music can be limited because of the intrinsic difficulty in manually controlling computer-generated sounds in real time; automatic synchronization is then a viable solution. Gesture tracking systems exploit alignment to provide interactive scenarios not only in art installations, but also in entertainment systems and video games; many of the recent succesful commercial game consoles make use of physical movement as primary control medium, such as Nintendo Wii[2] and Kinect for Microsoft Xbox[3].

This thesis investigates techniques for the alignment of media; the focus is directed towards music audio streams and gesture-capture time series, both cases of low dimensional signals evolving in time. A particular interest lies in the development of a unified framework for the alignment of different media representations, such as music notation and recorded performances, and in the robustness to local mismatches.

[2] www.nintendo.com/wii

[3] www.xbox.com/kinect

Structure and Contributions of the Thesis

In a first part, a theoretical exposition of alignment (Chapter 2) and identification (Chapter 3) techniques is presented. Original contributions are juxtaposed to a review of state of the art approaches, in order to present an analysis of the theoretical differences and highlight possible improvement. A second part (Chapter 4) presents several real-world application scenarios of the mentioned technologies and an experimental validation of the proposed contributions. Finally, Chapter 5 proposes a number of aspects that deserve to be investigated by future research.

Original Contributions

The original contributions of this thesis are summarized as follows:

- A methodology for (real-time) symbolic music alignment, with an explicit model of performance errors. [Section 2.3.3]
- A unified framework for the (real-time) alignment of an audio stream to both symbolic music representations and recorded references, eventually intermixed. [Section 2.4.2]
- A system for the (real-time) tracking and recognition of gesture profiles, capable of estimating additional dynamic signal features such as rotation and scaling. [Section 2.4.3]
- A scalable music identification system, aimed primarily at efficient query processing in large collections. [Section 3.3]
- A novel application of audio alignment techniques, for the acceleration of the mixing phase in studio recording productions. [Section 4.2]

A collection of classical music recordings for the evaluation of music identification task was also assembled and made available through a public evaluation campaign. [Section 4.4.2]

CHAPTER

TWO

ALIGNMENT

The concept of alignment refers to the identification and matching of corresponding substructures in related entities. In the domain of time-series analysis, it is common to refer to the alignment of an *input* sequence to a *reference* sequence.

This chapter presents a review of alignment techniques. In the context of *music*, sequences of radically different nature are taken into account: audio signals (digitized acoustic waveforms) and symbolic music notation (Figure 2.1). *Gesture-capture* time series of acceleration and absolute position values are related to physical movement (Figure 2.2).

The above mentioned domains exploit a common underlying set of techniques. Because of the heterogeneity of methods and application areas that have been proposed in the literature, it would be impracticable to present a review which classifies them into independent categories. It is however possible to observe an evolution of research trends, suggesting an organization according to the underlying general architecture.

An introduction to commonly used tools for modeling acoustic music content is provided in Section 2.1. Section 2.2 focuses on *deterministic* systems, with particular attention towards *Dynamic Time Warping* (DTW) techniques. Methodologies that make use of *probabilistic techniques* are the focus of subsequent sections. These systems typically adopt tracking approaches, modeling the input signal as a stochastic process whose state space (or subspace) ranges over the reference domain; they are categorized according to their state representation: *discrete state* probabilistic systems, such as *Hidden Markov Models* (HMMs) and their variants, are treated in Section 2.3, while Section 2.4 deals with *continuous state* systems, in particular those based on *Montecarlo Inference.*

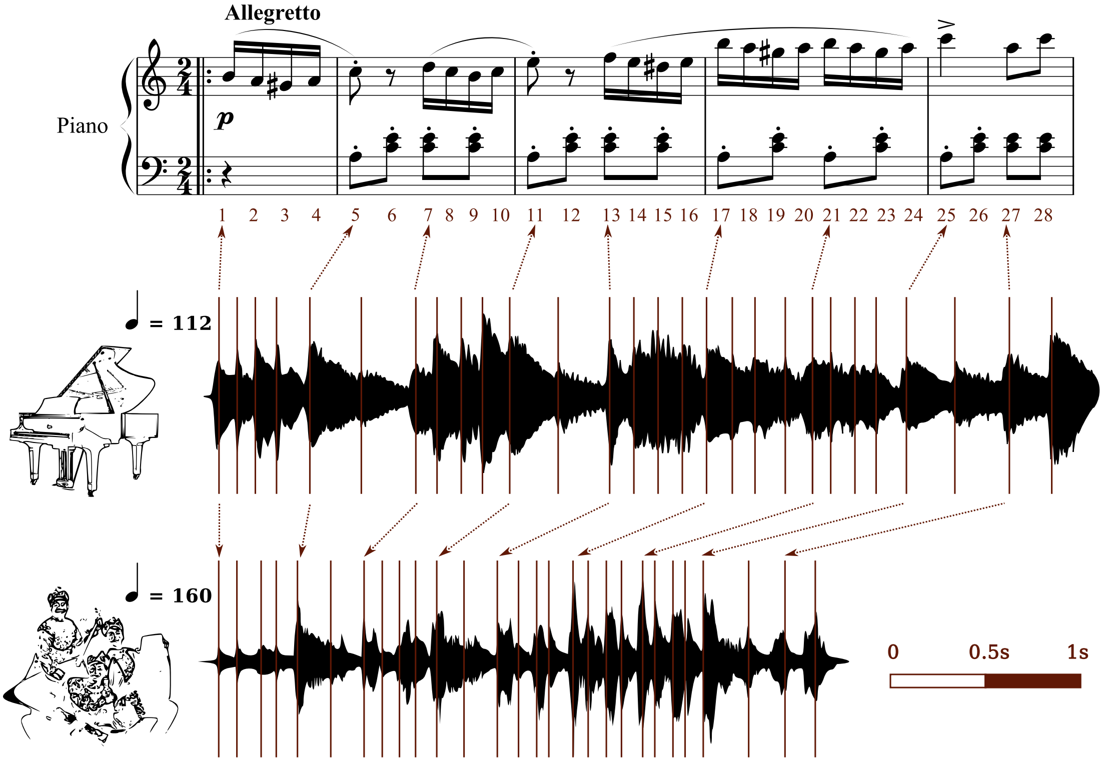


Figure 2.1: Alignment of music data: the third movement of Mozart's piano sonata K331, whose score is reproduced at the top, is interpreted by a pianist (center waveform) and an ensemble of folk musicians (bottom waveform) at significantly different tempos. Instances of both audio-to-score and audio-to-audio alignment are shown.

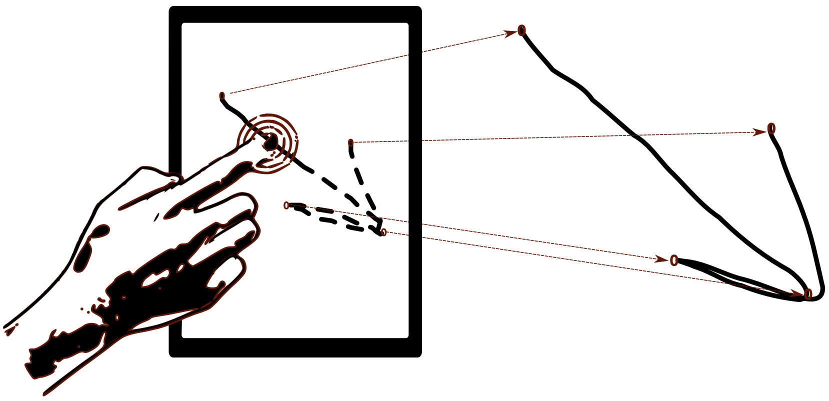


Figure 2.2: Alignment of gesture data: a two-dimensional gesture captured on a touchscreen device is aligned with a pre-recorded template, while it is still being executed.

2.1 Properties of Music Signals

A *signal* is a mathematical expression of a varying phenomenon. The digital recording of an acoustic event consists of a sequence of numeric values that measure, at regular time intervals, the acoustic intensity perceived by a listener. The notion of *frequency* is related to the periodic *repetition* of a signal over time. The phrase *frequency domain* is used to describe the domain for analysis of mathematical functions or signals with respect to frequency.

Digital Signal Processing (DSP) techniques are fundamental tools in many areas of engineering, and a crucial component in the context of music signal modeling and analysis. A rigorous treatment of the mathematical relationship between the time and frequency domains is beyond the scope of this thesis; [Oppenheim et al., 1983] provides a thorough explanation of the concepts involved, while [Steiglitz, 1997] presents an introduction to DSP with an explicit focus on audio content.

2.1.1 Modeling of Music Events in the Time Domain

Attack-Decay-Sustain-Release (ADSR) modeling is a generic model for the contours (*envelope*) of an acoustic waveform in the time domain. A popular technique in the world of electronic synthesizers, it is also a useful abstraction in the analysis of physical sounds. Figure 2.3 pictures a summarization of the four stages.

While the analysis of periodic waveforms – characteristic of the sustain and, to a lesser extent, release stages of sound – is usually performed in the frequency domain (Section 2.1.2), the attack and decay phases present strongly aperiodic development in time. The attack stage is of particular interest, as its correct recognition leads to the detection of event *onsets* (the beginning of a musical note or sound); an accurate localization of onsets in time can be successfully exploited in alignment approaches.

The presence of sudden energy bursts, along with variation in the spectral content of subsequent audio windows, are fundamental signal features exploited in onset detection approaches; a comprehensive review of such techniques is presented in [Bello et al., 2005].

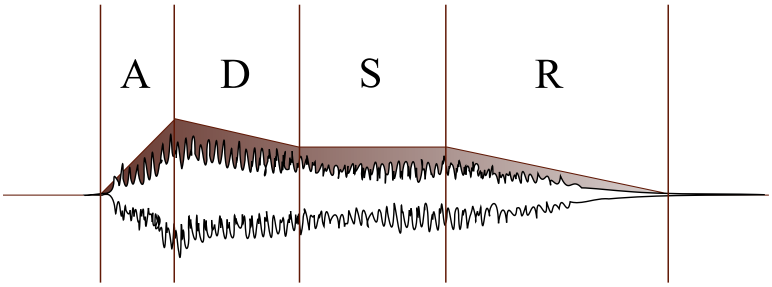


Figure 2.3: ADSR modeling of a note played on a double bass (bow).

2.1.2 Modeling of Music Events in the Frequency Domain

The theory of Fourier analysis provides tools to decompose a continuous function $s(t)$ into its constituent complex frequencies $S(f)$. Formally:

$$S(f) = \int_{-\infty}^{+\infty} s(t)e^{-i2\pi ft}dt \tag{2.1}$$

$$s(t) = \int_{-\infty}^{+\infty} S(f)e^{i2\pi ft}df \tag{2.2}$$

Under appropriate conditions, a periodic continuous function can be represented by a discrete sequence of values. Assuming that $f > f_{max} \implies S(f) = 0$ it is possible to sample the original continuous function at regular intervals $\Delta t \leq \frac{1}{2f_{max}}$ (Nyquist conditions) without any loss of information: the resulting sequence

$$s[n] = \delta(t - n\Delta t)s(t) \tag{2.3}$$

can be used to reconstruct the original signal; $\delta(\cdot)$ denotes the Dirac delta function:

$$\delta(x) = 0 \qquad x \in \mathbb{R} \setminus \{0\} \tag{2.4}$$

$$\int_{-\infty}^{+\infty} \delta(x)dx = 1 \tag{2.5}$$

The choice of the *sampling frequency* $F_S = \frac{1}{\Delta t}$ is crucial. The frequency range of human hearing is commonly assumed to be the range between 20 Hz and 20 kHz; low-end accelerometer devices operate with sampling intervals in the order of tens of milliseconds.

Periodic continuous signals are characterized by a discrete set of Fourier coefficients. This is exploited by the Discrete Fourier Transform (DFT) of a discrete signal $s[n]$ of length N

$$S[k] = \sum_{n=0}^{N-1} s[n]e^{-i2\pi \frac{k}{N} n} \tag{2.6}$$

$$s[n] = \frac{1}{N} \sum_{k=0}^{N-1} S[k]e^{i2\pi \frac{k}{N} n} \tag{2.7}$$

that can be computed efficiently by the Fast Fourier Transform (FFT) algorithm.

The frequency representation of real signals (both continuous and discrete) has Hermitian symmetry; as a corollary, those signals can be expressed by a weighted superposition of sinusoidal signals (the Discrete Cosine Transform (DCT) provides such decomposition). The modulus of the frequency representation of a signal is often referred to as *spectrum*.

In order to analyze a signal with respect to both its time and frequency aspects, it is common practice to subdivide it into short, overlapping windows at regular intervals (hop size). The representation of the spectral content of these window is referred to as *spectrogram*.

Spectral Modeling of Music Signals

The prominence of *pitched* sounds is typical of the western music tradition. Characterized by a regular frequency content, their spectrum is composed of a series of *harmonics*, spectral peaks whose frequencies form integer ratios with the *fundamental* (lowest) frequency.

The term *note* denotes a *pitch class* which encompasses a set of frequencies whose ratio is an integer power of two. A *scale* is a sequence of musical notes in ascending and descending order; scales are often associated to a *key*, a tone which listeners identify as predominant. The exact values of fundamental frequencies are determined by a *reference tuning frequency*, assigned to a particular note of the scale, and *temperament*: in the case of *equal temperament* the frequency of each note is $\sqrt[12]{2}$ the preceding one.

Such structure can be exploited in order to infer information about an acoustic signal. *Fundamental frequency estimation* approaches, such as the popular YIN algorithm [De Cheveigné and Kawahara, 2002], are often used to detect the notes being played. Another approach is based on *comparison* of the input signal with a series of templates: a popular strategy is to model a set of notes with fundamental frequencies $f_1 \ldots f_N$ as a mixture of Gaussian functions:

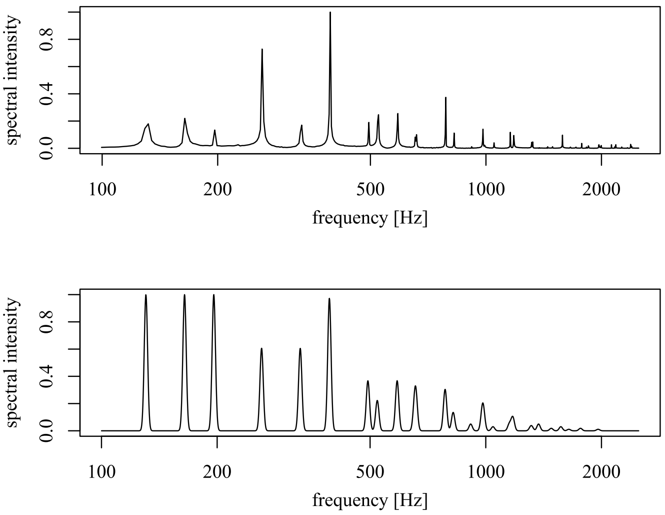


Figure 2.4: Spectral modeling of music signals. The spectrum of C major chord (C3, E3, G3), played on a piano (above) is compared to a template version constructed according to Equation 2.8 (below).

$$\text{profile}(\log(f)) = \sum_{n=1}^{N} \sum_{h=1}^{H} e^{-\lambda h} \mathcal{N}(\log(f) | \log(h f_n), \sigma^2) \tag{2.8}$$

where each note contribution is represented by a superposition of H harmonics with decreasing weights. Figure 2.4 compares a spectrum extracted from a recording with its synthetic model.

Chroma Vector Descriptors represent the energy associated to each pitch class (typically the 12 classes A, Bb, B ..., Ab) in a short time window; significant properties are the low-dimensionality and invariance to octave transpositions and timbre variations. A common formulation for the computation of the chroma descriptor $[c_1 \dots c_{12}]^T$ corresponding to an audio frame $s(t)$ is:

$$c_i = \sum_f B_i(f) \cdot S(f) \tag{2.9}$$

where $S(f)$ is a representation of $s(t)$ in the frequency domain and $B_i(f)$ is a masking template, centered on the semitones belonging to pitch class i. The actual computation varies depending on a number of parameters: $S(f)$ is usually the discrete Fourier transform of the signal, but can also be based on the instantaneous frequency of the signal, representing the main peaks of the spectrum. The shape of the $B_i(f)$s and their support in the audible range are expected to influence the ability of chroma descriptors to discriminate different music signals. Introduced in [Fujishima, 1999], many alternative computation strategies have been devised: [Ellis and Poliner, 2007], [Gómez, 2006], [Müller and Ewert, 2011].

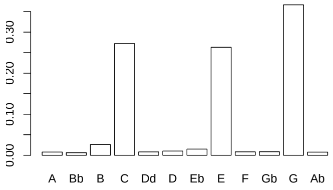


Figure 2.5: A Chroma vector descriptor, corresponding to the audio frame of Figure 2.4.

2.2 Dynamic Programming-based Systems

Dynamic programming approaches, often inspired by techniques for speech processing and biological sequence analysis, provide an intuitive way to compute a "warping" between two time series. Systems based on *pattern matching* algorithms, applied to the alignment of symbol sequences characterized by a finite alphabet are reviewed in Section 2.2.1. The technique of *Dynamic Time Warping* is introduced in Section 2.2.2; its applications are subsequently discussed in the contexts of music (Section 2.2.3) and gesture (Section 2.2.4).

2.2.1 Pattern Matching-based Music Systems

The application of alignment algorithms to music data was introduced in the context of automatic accompaniment of a human player by a computer. The first works attempted to "detect what the soloist is doing", "match the detected input against a score" [Dannenberg, 1984], and "understand the dynamics of live ensemble performance" [Vercoe, 1984] so that a computer system might be used to provide a responsive accompaniment to the player.

Alignment is performed in real time between untimed sequences of pitch symbols belonging to a finite alphabet of tempered scale pitches, such as the one expressible by the "pitch class, octave" notation: C0, Db0, ... B7, C8. Electronic musical instruments standards such as MIDI[1] can be used to obtain such sequences in a straightforward manner. Due to the limited computational power available at the time, real time pitch detection from acoustic signals was unfeasible without considering additional data sources: in [Vercoe, 1984] flute fingerings are detected using optical sensors installed in the instrument body, allowing an independent Digital Signal Processing (DSP) system to perform filtering, and consequently note detection, on a reduced set of pitch hypotheses.

The problem of alignment is transformed into a search for the *longest common subsequence* of the note sequences: using Dynamic Programming (DP) techniques, efficient real-time algorithms can be formulated. Subsequent works [Bloch and Dannenberg, 1985] extended these strategies in order to handle compound (polyphonic) events, such as chords in the case of a keyboard performance, where the order of the simultaneous events is undefined. Another improvement was introduced through the use of multiple concurrent instances of the alignment algorithm in [Dannenberg and Mukaino, 1988]. Most recent works dealing with symbolic note sequences adopt a partial [Pardo and Birmingham, 2002] or fully [Schwarz et al., 2004] probabilistic approach.

2.2.2 Review of Dynamic Time Warping

Dynamic Time Warping (DTW) is a popular generic technique for time series alignment. A simple formulation of the algorithm is reviewed below; [Rabiner and Juang, 1993] and [Müller, 2007] give a comprehensive review of its many variants in the context of acoustic data processing.

Definition

Let $u_1 \dots u_{L_u}$ and $v_1 \dots v_{L_v}$ denote respectively the input and reference time series. A *warping path* between these sequences is defined as a sequence of *pairs*; each pair (i, j) indicates that the points u_i and v_j are aligned.

The *local cost* of a pair $d(i, j)$ denotes a *penalty score* related to the distance between the points u_i and v_j; the aim of the DTW algorithm is to find the *minimum cost path* between the sequences, where the path cost is defined as the sum of the costs of the pairs it contains.

[1] `http://www.midi.org` – The original MIDI standard specifications were originally introduced in 1982.

To provide a meaningful result, several constraints are placed on the form of the path. In a popular formulation, the path is *bounded at the ends* of both sequences, *continuous* and *monotonic*. The first two conditions fix the length of path to $L_{path} = \max\{L_u, L_v\}$; the last two imply that the difference of two subsequent pairs in a path must be one of $\{(0,1),(1,0),(1,1)\}$.

Computation of the Minimum Cost Path

A recursive dynamic programming algorithm can be formulated according to the above mentioned constraints. In its simplest form, the cost of a partial minimum cost path ending with the pair (i,j) is given by:

$$D(1,1) = d(1,1) \tag{2.10}$$

$$D(i,j) = d(i,j) + \min\left\{w_a D(i,j-1), w_b D(i-1,j), w_c D(i-1,j-1)\right\} \qquad (i,j) \neq (1,1) \tag{2.11}$$

A backtracking step can extract the desired global minimum cost path, tracing the recursion step backwards from the entry $D(L_u, L_v)$. Computation of the warping path is $O(L_u L_v)$.

2.2.3 DTW-based Music Alignment Systems

An application of DTW to the alignment of an audio signal with a symbolic score was proposed in [Orio and Schwarz, 2001]. The authors avoid resorting to signal processing approaches for pitch extraction, due to their insufficient accuracy (especially in the case of polyphonic signals); they resort instead to the *comparison* of the input audio spectrum with programmatically constructed *templates*. The *peak spectral distance* (PSD) measure of similarity works by summing the energy of the (normalized) audio spectrum that falls into the frequency bands associated to the harmonic sinusoidal partials of the expected notes. A similar approach is presented in [Dannenberg and Hu, 2003], where the reference score (in the form of a MIDI file) is synthesized, and alignment is performing using chroma features.

[Müller et al., 2004] deals with polyphonic piano music. Instead of using frequency-based audio representations, a bank of 88 time-domain elliptic filters, each centered on the frequency associated to a piano key, is used to perform an ad-hoc subband decomposition.

In [Dixon, 2005], [Dixon and Widmer, 2005] the authors present an audio-to-audio alignment approach that uses euclidean distance for computing audio similarity between spectral representation. Optimizations on the ranges of the recursive computation of the warping path allow the system to run in real time. The work was subsequently extended in [Arzt et al., 2008].

In [Müller et al., 2006], the authors introduce the idea of *multiscale* alignment: an alignment path, computed at a coarse resolution level, is recursively projected onto the next level and refined; the approach makes use of chroma features.

A two-pass approach is described in [Niedermayer, 2009], where an initial alignment obtained through DTW using chroma features is refined using Non-Negative Matrix Factorization techniques (discussed in Section 2.3.5): note onsets are then corrected according to the pitch activation patterns yielded by the matrix factorization.

2.2.4 DTW-based Gesture Alignment Systems

The algorithm *uWave* [Liu et al., 2009] is based on the assumption that human gestures can be recognized using time series of forces applied to an accelerometer device. It uses a library that stores one or more instances of every vocabulary gesture, in the form of acceleration time series provided by a three-axis accelerometer. Dynamic Time Warping is used in conjunction with a simple similarity measure between acceleration vectors based on Euclidean distance. Gesture recognition is performed by selecting the alignment with minimum final cost. An issue identified by the authors is related to the orientation of control devices: since a controller can be tilted around three axes, the reading of the accelerometer might not reflect consistently the external force applied by the user.

2.3 Probabilistic Discrete State Systems

The process of performing a music work can be regarded as stochastic because of the freedom of interpretation, yet the knowledge about the work that can be obtained from a score or previous performances can be exploited to model an interpretation. Similarly, a physical gesture can be modeled stochastically, to account for (often unintentional) differences in repeated executions. An introduction to probability theory is provided in [Ross, 2002].

A *stochastic process* is a collection of random variables. A discrete time stochastic process $x_0, \ldots, x_t$ forms a *Markov chain* if:

$$P(x_t = a_t | x_{t-1} = a_{t-1}, x_{t-2} = a_{t-2}, \ldots, x_0 = a_0) = P(x_t = a_t | x_{t-1} = a_{t-1}) \tag{2.12}$$

that is, state x_t depends on the previous state x_{t-1} but is independent of the particular history of how the process arrived at state x_{t-1} (*Markov* or *memoryless property*).

2.3.1 Review of Hidden Markov Models

A *Markov chain* is a memoryless random process, in which the state of the system undergoes transitions among a finite (or, more generally, countable) set of states. In a straightforward interpretation, each state corresponds to an observable physical event, and Hidden Markov Models extend this concept to include the case in which the observation is a *probabilistic function* of the state: the resulting models is a *doubly embedded stochastic process*, where an underlying, non-observable stochastic process (hence the term *hidden*) can only be observed through another set of stochastic processes that produce a sequence of observations.

Below are reviewed the most important aspects in relation to the content of this thesis. The classic introduction in [Rabiner, 1989] presents a more rigorous mathematical treatment, and [Bishop, 2006] the same topics are approached from an alternative mathematical perspective.

It should also be noted that from a purely mathematical point of view, HMM and DTW are interchangeable [Durbin et al., 1998]. Advantages of HMMs are their *contextual interpretation* and the possibility of performing supervised training in a statistically meaningful fashion.

Definition

A Hidden Markov Model is characterized by:

- the sets of states in the model and observation symbols, respectively N, V;
- the state transition probability matrix $A = \{a_{i,j} = P(x_t = j \mid x_{t-1} = i)\}, \quad i, j \in N$.
- the observation probability distribution: $B = \{b_j(z) = P(z \mid x_t = j)\}, \quad j \in N, z \in V$.
- the initial distribution on states $\pi = \{\pi_i\}, \quad \pi_i \in \mathbb{R}, \quad i \in N, \quad \sum_{i \in N} \pi_i = 1$

A Hidden Markov Model can then be specified with the compact notation: $\lambda = (A, B, \pi)$.

Incremental Inference: the Forward Decoding Algorithm

The forward decoding algorithm makes used of *forward variables*, which represent the probability of a partial observation sequence up to time t ending in state i:

$$\alpha_t(i) = P(z_1 \dots z_t, x_t = i \mid \lambda) \qquad i \in N \tag{2.13}$$

The definition suggests the following strategy for recursive computation:

$$\alpha_1(j) = \pi_j b_j(z_1) \qquad j \in N \tag{2.14}$$

$$\alpha_{t+1}(j) = \left[\sum_{i \in N} a_{i,j}\ \alpha_t(i) \right] b_j(z_{t+1}) \qquad 1 \leq t < T, j \in N \tag{2.15}$$

Each step of the forward variables computation can be carried out in a time proportional to the square of the cardinality of the set of states. The values of α_T also provide a way to evaluate the *likelihood of the input sequence*:

$$P(z_1 \dots z_T | \lambda) = \sum_{i \in N} \alpha_T(i) \tag{2.16}$$

Assuming that the values of the forward variables are proportional to the probability distribution over states, it is straightforward to make use of normalization to decode:

$$P(x_t = i | z_1 \dots z_t, \lambda) = \frac{\alpha_t(i)}{\sum_{j \in N} \alpha_t(j)} \tag{2.17}$$

and consider the most probable state at time t as:

$$\hat{x}_t = \arg\max_{i \in N} \alpha_t(i) \tag{2.18}$$

Smooth Inference: the Forward-Backward and Viterbi Algorithms

Contrarily to the forward decoding algorithm, which considers at each instant only *past* observations, smooth inference algorithms make use of the whole observation sequence.

As was the case for forward variables (Equation 2.13), *backward variables* can be defined in a similar fashion

$$\beta_t(i) = P(z_{t+1} \dots z_T, x_t = i | \lambda) \qquad i \in N \tag{2.19}$$

and a recursive computation procedure can be devised:

$$\beta_T(i) = 1 \qquad i \in N \tag{2.20}$$

$$\beta_t(i) = \sum_{j \in N} a_{i,j} b_j(z_{t+1}) \beta_{t+1}(j) \qquad t < T, i \in N \tag{2.21}$$

Using backward variables in combination with forward variables, it is possible to compute the probability distribution over states given the complete observation sequence:

$$\gamma_t(i) = P(x_t = i | z_1 \dots z_T, \lambda) \tag{2.22}$$

$$= \frac{\alpha_t(i)\beta_t(i)}{\sum_{j \in N} \alpha_t(j)\beta_t(j)} \tag{2.23}$$

The individually most likely sequence is then decoded, for each t, as:

$$\hat{x}_t = \arg\max_{i \in N} \gamma_t(i) \tag{2.24}$$

Even though the state sequence decoded using the forward-backward algorithm maximizes the expected number of correct states, such sequence does not necessarily correspond to the most likely one: it might even be unfeasible in case the transition probability between two subsequent states in the decoded path is 0.

The *Viterbi* algorithm aims at finding the most probable *global* sequence. Let

$$\delta_t(i) = \max_x P(x_1, \ldots, x_{t-1}, x_t = i, z_1, \ldots, z_t \mid \lambda) \qquad i \in N \tag{2.25}$$

be the best "score" (highest probability) along a single path at time t, given the first t observations, that ends in state i. By induction, the following algorithm can be devised:

$$\delta_1(i) = \pi_i b_i(z_1) \qquad i \in N \tag{2.26}$$

$$\delta_t(j) = \big(\max_{i \in N} \delta_{t-1}(i) a_{i,j}\big) b_j(z_t) \qquad j \in N, 1 < t \leq T \tag{2.27}$$

$$\psi_t(j) = \arg\max_{i \in N}(\delta_{t-1} a_{i,j}) \qquad j \in N, 1 < t \leq T \tag{2.28}$$

The term $\psi(\cdot)$ is used in the final backtracking step, required to recover the complete sequence:

$$\hat{x}_T = \arg\max_{i \in N} \delta_T(i) \tag{2.29}$$

$$\hat{x}_t = \psi_{t+1}(\hat{x}_{t+1}) \qquad t < T \tag{2.30}$$

Training

The training problem is by far the most complex one, as there are no algorithms that allow to analytically adjust the model parameters in order to find a *globally optimal* solution. Nonetheless, several techniques exist, of which the most popular one is the *Expectation-Maximization* (EM) method introduced in [Baum et al., 1970] and reproposed, in the context of Information Theory, in [Welch, 2003]. For a rigorous introduction to the EM method, in the context of mixture models, see [Bishop, 2006]. An alternative training technique is based on Viterbi Decoding; detailed in [Lember and Koloydenko, 2008], it is often preferred to EM parameter estimation because of its reduced computational demands, at the expense of theoretically poorer results[2].

In the models considered in this chapter however, the transition graph and observation models are created parametrically, starting from a reference medium, *before* any input sequence is observed. The main practical reason behind this lies in the intrinsic difficulty of manual data annotation, which renders large-scale supervised learning unfeasible. Even if manual annotation is available describing a correct alignment, the training algorithms reviewed above cannot be exploited unless this annotation refers to the individual states of the underlying HMM model. Moreover, the HMM is created parametrically, thus independent models should be trained for each individual reference media. For these reasons, it is customary to adjust the model parametrization by hand, or resort to more general optimization schemes.

2.3.2 HMM-based Symbolic Music Alignment Systems

The rationale behind audio to score alignment approaches that make use of Hidden Markov Models is that the most relevant acoustic features of a music performance can be modeled statistically as observations. These systems exploit a *discrete representation of the reference media*, most often in the form of a *linear chain* of events. Such structure is justified by the intuitive similarity to the linear nature of event sequences. Moreover, the resulting *left-to-right* topology of the graph allows efficient decoding strategies to be implemented, by constraining the

[2] Contrarily to the EM procedure, Viterbi decoding training does not guarantee that the likelihood of the sequence, given the updated model parameters, is non-decreasing at each iteration of the algorithm.

computation so that only states in proximity of the previous most probable state are involved. As Figures 2.6(a) and 2.6(b) show, it is straightforward to divide a symbolic score into non-overlapping events, each composed of a (possibly empty) set of notes. In monophonic scores, all the notes and explicit rests correspond to events, while events in polyphonic scores are bounded by onsets and offsets of all the notes played by the various instruments/voices.

The transition matrix of the HMM model is built parametrically, according to the music score. The incoming audio signal is divided into short, possibly overlapping frames of fixed length; a transition occurs every time a new audio frame is observed and the advancement of the performance with respect to the score is tracked by decoding. The crucial point in the design of an HMM-based score alignment system lies in the definition of the graph topology (the transition matrix) and observation modeling strategies.

Graph Topologies

It is relatively straightforward to model the topology of a graph, associated to the transition matrix of the HMM, so that it intuitively represents the score as a linear chain of events as depicted in Figure 2.6(c). Individual events are in turn modeled by sequences of states, as in Figure 2.6(d).

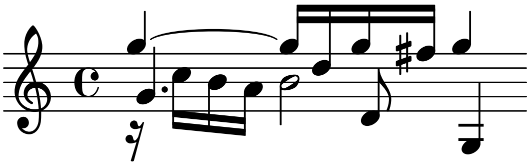
[Cano et al., 1999] was among the first systems for audio to score alignment. The authors present a model aimed at *monophonic* music, extending the work on voice alignment of [Puckette, 1995] by moving from a rule-based model to a probabilistic system. In their model, each event is modeled as a 3-state sequence *attack-sustain-release*, which allows to adopt different observation functions for each stage of the event model.

[Raphael, 1999] introduced a popular strategy for modeling sustained notes or rests, using a *chain* of states (as opposed to a single state with a self-loop probability) so that a probability distribution can be assigned to the event *duration.* In this formulation each state has a self-loop probability p as shown in Figure 2.6(d). The probability of a segment duration d using n states is then modeled by a negative binomial distribution

$$p(d) = \binom{d-1}{n-1} p^{d-n}(1-p)^n \tag{2.31}$$

with expected value $\mu = \frac{n}{1-p}$ and variance $\sigma^2 = \frac{np}{(1-p)^2}$. The duration of an event is modeled by setting the values of n and p accordingly; in particular the value of μ is proportional to the event duration in the score, and two cases can be distinguished, depending on whether n is fixed or variable. In the former case, the total number of states in the graph is proportional to the number of events in the score. Event duration is modeled by self-loop probability, however the variance of the distribution changes with events duration. An additional problem with such model arises in presence of short events, where the expected number of time steps is smaller than the length of the state chain: $d < n \Rightarrow p(d) = 0$; in that case, a different graph topology has to be adopted. Adopting a variable value for n allows for a more precise modeling of event duration. It is reasonable to compute n and p so that the variance of the duration distribution is proportional to the expected event duration. In such case p is constant for all the events, and the only parameter responsible for the event duration is the number of sustain states, the total number of which is thus proportional to the score duration.

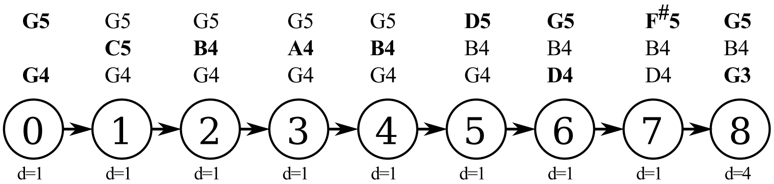
In [Orio and Déchelle, 2001] an additional type of state was introduced, aimed at modeling error situations explicitly; such states are named *ghost states*, as they are not related to particular events in the score, in contrast to *event states*, associated to chords and rests. This strategy was used also in [Schwarz et al., 2004] for a symbolic (MIDI-based) score follower. Ghost states, which also provide a convenient measure of confidence of the system, are described in detail in Section 2.3.3.



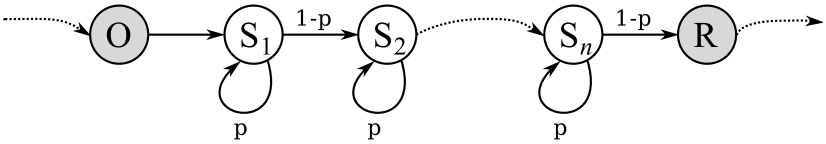
(a) Original score.

event	0	1	2	3	4	5	6	7	8
notes	**G5**	G5	G5	G5	G5	**D5**	**G5**	**F$^{\#}$5**	**G5**
		C5	**B4**	**A4**	**B4**	B4	B4	B4	B4
	G4	G4	G4	G4	G4	G4	**D4**	D4	**G3**
duration	1	1	1	1	1	1	1	1	4

(b) Event sequence (shaded entries represent onsets).



(c) Chain-of-events representation.



(d) Representation of individual events in the HMM transition graph, with optional onset (O) and rest (R) states, and variable-length sustain-states (S) chain.

Figure 2.6: Model of symbolic scores in Hidden Markov Model-based systems.

Observation Modeling

Observation modeling is the mechanism which allows a system to respond to external signals. In the case of Hidden Markov Models, this translates into an appropriate choice of a probability distribution $p(z|x)$ over all possible observations z for each state x.

Even though fundamental frequency estimation is a key component of the approach described in [Cano et al., 1999], which is targeted at monophonic music, real time pitch detection of polyphonic signals is still prone to errors that would substantially affect the quality of the alignment. To this end, most systems opt to compare the incoming signal features with the expected ones corresponding to particular graph states. Observation likelihood computation is dependent on the type of state, and is closely related to the graph topology aspects described in the previous section.

[Orio and Schwarz, 2001] introduced the Peak Spectral Distance method, mentioned in Section 2.2.3. Later versions of the system and alternative approaches in the literature make use of more refined similarity measures, as reviewed in [Orio et al., 2003]. In particular, modeling spectra as vectors in an Euclidean space presents the opportunity to exploit the *cosine similarity* measure of angle θ between two vectors u, v:

$$\cos(\theta) = \frac{u \cdot v}{||u||\ ||v||} \tag{2.32}$$

On the other hand, observation modeling of rest and onset states typically make use of simpler strategies based on signal energy thresholding.

An advantage of HMM systems over the non-probabilistic approaches presented in Section 2.2 is the possibility of training the observation model using statistically meaningful methods. Relevant contributions in this regard are [Cano et al., 1999], [Raphael, 1999], [Orio and Déchelle, 2001], [Schwarz et al., 2005].

2.3.3 A Discrete Filter Bank Approach to Audio to Score Matching

This section describes our contribution in the context of audio to score alignment, using a HMM model. [Montecchio and Orio, 2008] and [Montecchio and Orio, 2009] build upon the system described in [Orio and Déchelle, 2001], introducing relevant improvements in both the observation model and graph topology.

Explicit Modeling of Error Situations

In its simplest form, the topology of the score level graph represents an event sequence as a linear chain of states. This approach however does not present any explicit model for local differences between the score representation and the audio performance, thus the overall alignment can be severely affected by local mismatches in case of unexpected input.

In order to overcome such problems, ghost state were introduced in [Orio and Déchelle, 2001]. The basic graph topology is modified so that event states can perform a transition to an associated ghost states, which in turn can perform either a self-transition or a forward transition to subsequent event states. The final representation is made up of two parallel chains of nodes, as shown in Figure 2.7. In case of local differences between the score and the performance, the most probable path can pass through one or more ghost states and return to the lower chain when the performance matches again the score.

In their original formulation, all observation likelihood values and transition probabilities related to ghost states were fixed. While this provides a convenient “escape route” for the alignment path in case of score misalignments, such modeling leaves room for improvements. In [Montecchio and Orio, 2008], the transition probabilities from a ghost state to subsequent

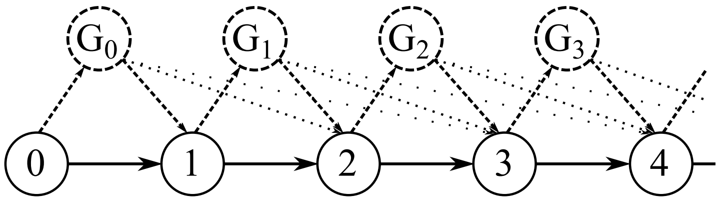


Figure 2.7: Refined score graph topology with explicit error modeling.

event states are proportional to the proximity of events with the score: this resembles the idea of *locality* of a mismatch due to an error. The same principle is applied to the observation modeling strategy for ghost states, that linearly combines the observation likelihoods of related event states with an exponentially decreasing weighting as a function of score distance.

Discrete Audio Processing Front-end

Spectrum analysis is typically done via the Fast Fourier Transform algorithm. A significant problem with this approach is related to the linear frequency resolution of the FFT, which leads to a significant loss of precision in the lower frequency range. The situation is partially compensated by upper harmonics, however different strategies can be studied to improve the performances of a system. The use of a bank of discrete filters, investigated in [Müller et al., 2004] in the context of piano music, is a possible solution. The implementation of [Montecchio and Orio, 2009] uses second order filters of the form

$$H_i(z) = \frac{(1-r_i)\sqrt{1-2r_i\cos(2\theta_i)+r_i^2}}{(1-r_ie^{-j\theta_i}z^{-1})(1-r_ie^{j\theta_i}z^{-1})} \qquad z \in \mathbb{C}, \theta_i \in [0,2\pi), r_i \in [0,1) \tag{2.33}$$

$$\theta_i = 2\pi\frac{f_i}{F_S} \qquad 0 \le f_i < \frac{F_s}{2} \tag{2.34}$$

$$r_i = \exp\big\{-\frac{\theta_i}{2}(2^{\frac{\Delta W}{24}} - 2^{-\frac{\Delta W}{24}})\big\} \tag{2.35}$$

which have unit gain at θ_i (the nominal frequency of the i-th note, normalized with respect to the sampling frequency F_S) and a -3dB bandwidth of ΔW (expressed in semitones).

The application of a digital filter has the effect of introducing a frequency-dependent *delay* in a signal processing chain. The *group delay* of a filter is equal to the negative derivative of its phase with respect to the angular frequency; in the case of a filter expressed by Equation 2.33, the group delay for the i-th filter, evaluated at its central frequency, is:

$$-\frac{d}{d\omega}\angle H_i(j\omega)\big|_{\omega=\theta_i} = \frac{r\,\left(2\,r^2-(3\cos{(2\,\theta_i)}+1)\,r+\cos{(2\,\theta_i)}+1\right)}{(r-1)\,(r^2-2\cos{(2\,\theta_i)}\,r+1)} \tag{2.36}$$

Assuming that each filter has the same bandwidth in semitones, filters corresponding to the lowest notes have a much higher group delay; to narrower filters correspond higher group delays, as shown in Figure 2.8.

The specific amount of delay is traded off with the filter bandwidth in a real-time context: the output of each filter is routed to a delay line in order to compensate for the different group delays (Figure 2.9); as a consequence, the alignment systems reaction time is equal to the group delay associated with the filter having the lowest frequency.

2.3.4 HMM-based Gesture Alignment Systems

A vast amount of works in the literature deals with gesture recognition techniques in the field of machine vision. In the context of this thesis, attention is restricted to works that approach

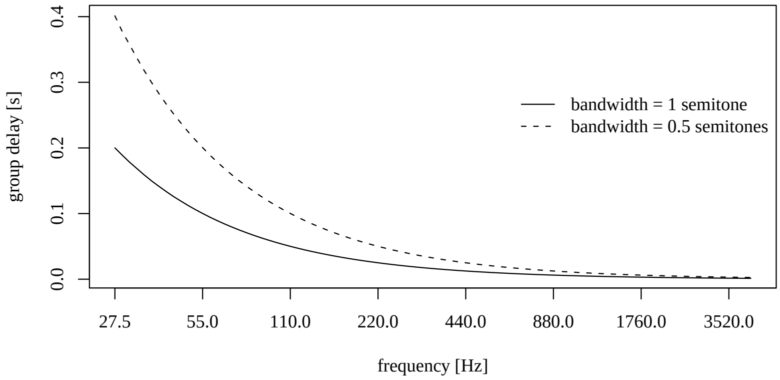


Figure 2.8: Frequency-dependent bandwidth/delay trade off for digital filters: the group delay for the filter of Eq. 2.33 is pictured, marking on the horizontal axis the fundamental frequencies of the A0 . . . A7 notes in a piano keyboard.

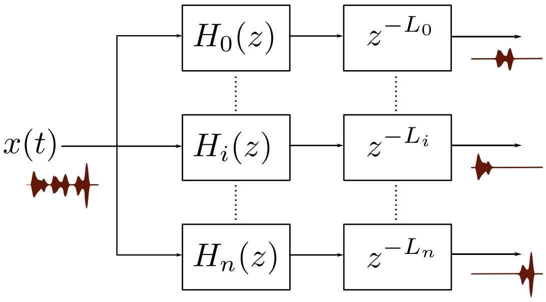


Figure 2.9: Discrete filter bank with delay compensation.

the problem of gesture recognition by matching the trajectories delineated by *low-dimensional* feature sequences, such as those captured from accelerometer devices.

In this context, the identification of a *test* gesture is carried out by selecting the best match amongst a set of pre-recorded *template* gestures. Several techniques rely on Hidden Markov Models since it is possible to *learn* the temporal structure of templates. HMMs give a compact view of a gesture trajectory by clustering its parts and building a transition model over it. In [Wilson and Bobick, 1999] the authors propose a model that takes into account parametric changes in execution, a concept further investigated in [Brand and Hertzmann, 2000].

A State of the Art HMM-based Gesture Tracking System

In this section we review the methodology presented in [Bevilacqua et al., 2010], a model for the identification and tracking through alignment which underlies the gesture follower currently in use at IRCAM[3]. It is considered the state of the art for use in artistic productions, having been used in complex situations such as tracking of gestures performed by instrumentalists or dancers for controlling digital media.

[3] Institute de Recherche et Coordination Acoustique/Musique, located in Paris (FR). `http://www.ircam.fr`

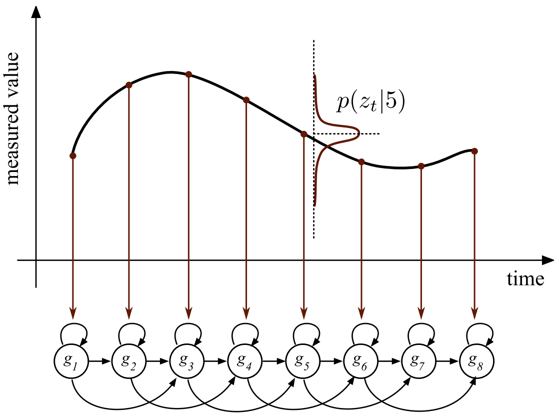


Figure 2.10: Hidden Markov Model for alignment of one-dimensional gesture signals.

Alignment Model Contrarily to models dealing with music content, where the underlying musical idea is mediated via a spectral representation that evolves over time, in the case of gesture data the relevant feature vectors are observed directly. Let $g_1 \dots g_{L_g}$ be a reference *template* gesture, and $z_1 \dots z_{L_z}$ an *input* gesture that is to be tracked. Both signals are uniformly sampled at a given sampling rate.

Each state in the HMM is associated to a sample of the template gesture; the parameter σ is a constant adjusted by the user controlling the degree of precision to which the system expects gestures to be repeated. A left-to-right HMM is built from the template, and captures the temporal evolution of the continuous signal. The topology of the graph associated to the state transition matrix is depicted in Figure 2.10.

The observation model considers a multivariate normal distribution. For an observation z_t at time t, the conditional observation likelihood for the k-th state is computed as

$$p(z_t|k) = \mathcal{N}(z_t|g_k, \sigma^2) \tag{2.37}$$

The forward decoding algorithm infers in real time both the most probable position along the reference gesture and the sequence likelihood. A major drawback of this approach is that, in order to estimate other features such as scaling, offset and rotation, an independent HMM must be constructed for each possible combination of the (discretized) invariants, rendering the approach unfeasible in many situations where the above mentioned invariants cannot be controlled.

Extension to Gesture Recognition The above model can be extended in order to be used in real-time recognition tasks. A straightforward solution to this tasks considers a series of *independent* left-to-right HMMs, one for each gesture in the template set, modeled as reported before. Each incoming observation increments the forward variables for each template. The recognized gesture is the one characterized by the highest likelihood value at the last observation.

2.3.5 Alternative Models

Several alternative statistical frameworks have been proposed in the alignment literature to overcome the shortcomings associated to HMMs. In the music domain, the most perceived limitation in such models is the lack of an explicit control over *tempo*, indeed one of the very fundamental aspects of music.

The use of Hierarchical Hidden Markov Models (HHMMs) [Fine et al., 1998], a recursive generalization of HMMs developed to model multi-scale structures, is investigated in [Cont, 2006]. The approach also makes use of Non Negative Matrix Factorization [Lee and Seung, 2001], an unsupervised learning technique that decomposes a signal into the individual contributions of a set of templates. A particular focus is directed towards the *sparseness* of the resulting representation. [Cont, 2008a] presents an evolution of the system, where the state-space generative model of the score is modeled as a Hidden Hybrid Markov/semi-Markov chain (HHMSM). In such model the duration associated to states is dynamically varying, and dependent on the decoded tempo. The work also analyzes issues related to musical time, in particular with references to events that are inherently *atemporal*, such as the succession of notes in trills or glissandos. Additional considerations on the modeling of tempo are reported in [Cont, 2010].

An architecture based on Bayesian graphical models is the subject of [Raphael, 2004], extended in [Raphael, 2006]. Position in the score and tempo are modeled as probabilistic motion equations, in a way that is similar to the approach that presented below in Section 2.4.2; the inference algorithm however is based on dynamic programming techniques.

Conditional Random Fields (CRFs) [Lafferty et al., 2001] are a *discriminative* probabilistic graphical model. Contrarily to HMMs, which model the *joint* distribution of state and observation sequences, CRFs model the conditional distribution of observations given state sequences explicitly. The possibility of computing observation functions from several analysis frames is exploited in the context of audio-to-score alignment in [Joder et al., 2010].

2.4 Probabilistic Continuous State Systems

Sequential Montecarlo Inference techniques, often referred to as Particle Filtering, were developed to perform inference on stochastic processes that do not typically admit analytic solutions. These methods deal with state-space systems characterized by *continuous* hidden variables, and are characterized by an intuitive inference procedure which resembles a *simulation.* Introduced in [Gordon et al., 1993], they subsequently enjoyed great popularity. In comparison with standard approximation methods, they do not depend on local linearisation techniques, and allow for great flexibility at the expense of computational power. For a rigorous mathematical treatment of these topics, the tutorials [Arulampalam et al., 2002] and [Doucet and Johansen, 2009] should be consulted; a thorough introduction to Particle Filtering methods in the broader context of Bayesian signal processing theory can also be found in [Candy, 2009].

Section 2.4.1 presents a compact review of inference based on Sequential Importance Sampling, which forms the basis of the alignment approaches described in Sections 2.4.2 and 2.4.3.

2.4.1 Review of Sequential Montecarlo Inference Techniques

State-space approaches to time series modeling revolve around the representation of the system state as a vector, which contains all relevant information required to describe the system under investigation. Bayesian approaches construct the posterior state probability density function using all available information, including the set of received measurements.

Recursive Bayesian Filtering Framework

Suppose that a system can be described by a state transition probability distribution function $p(x_t|x_{t-1})$ and a conditional observation probability distribution $p(z_t|x_t)$, dependent on the current state x_t. Even though these functions could in principle be time-dependent, in the remainder of this section the attention is restricted to a simpler case, for the sake of clarity.

The aim at each time step is to infer an up-to-date state probability distribution, given the complete information about previous observations, $p(x_t|z_{1\dots t})$. A recursive procedure can then be devised which operates in a prediction-update fashion.

Prediction: supposing that the pdf $p(x_{t-1}|z_{1\dots t-1})$ is available, the Chapman-Kolmogorov equation can be used to predict the new pdf without knowledge of the latest measurement z_t:

$$p(x_t|z_{1\dots t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|z_{1\dots t-1})dx_{t-1} \tag{2.38}$$

Update: as soon as a measurement z_t becomes available, Bayes' rule can be used to infer

$$p(x_t|z_{1\dots t}) = \frac{p(z_t|x_t)p(x_t|z_{1\dots t-1})}{p(z_t|z_{1\dots t-1})} \tag{2.39}$$

Unfortunately, the recursive propagation of the posterior density is only a conceptual solution that cannot be determined analytically in general cases. The advantage of Montecarlo approximation algorithms resides in the lack of restrictions in the formulation of the state transition and observation probabilities distributions.

Incremental Inference: the Sequential Importance Sampling Algorithm

The Sequential Importance Sampling (SIS) algorithm is a technique for implementing a recursive Bayesian filter via Montecarlo simulation. It is based on the principle of Importance Sampling, which, applied in order to obtain an approximation of the "true" posterior pdf, allows for a tractable inference algorithm.

Importance Sampling Let $p(x) \propto \pi(x)$ be a pdf from which it is difficult to extract samples, but which is easy to evaluate up to a multiplicative constant. Let samples $x^1 \dots x^{N_s}$ be drawn from an arbitrary pdf $q(\cdot)$. Then, an approximation of $p(x)$ can be expressed as:

$$p(x) \approx \sum_{i=1}^{N_s} w^i \delta(x - x^i) \qquad\qquad w^i \propto \frac{\pi(x^i)}{q(x^i)}, \sum_{i=1}^{N_s} w^i = 1 \tag{2.40}$$

Sampling among $x_1 \dots x_{N_s}$ according to the discrete distribution function $w_1 \dots w_{N_s}$ provides a convenient, although possibly inefficient, way to sample from the original $p(x)$.

Continuous Posterior Pdf Approximation A discrete *random measure* approximates the analytical form of the (continuous) state probability distribution, using the principle of Importance Sampling. Let $\{x_t^i, w_t^i\}_{i=1}^{N_s}$ denote a set of N_s *support points* $x_t^1 \dots x_t^{N_s}$ (often referred to as *particles*) along with their associated weights $w_t^1 \dots w_t^{N_s}$, $\sum_{i=1}^{N_s} w_t^i = 1$ at time t. It is then possible to consider the approximation:

$$p(x_t|z_{1\dots t}) \approx \sum_{i=1}^{N_s} w_t^i \delta(x_t - x_t^i) \tag{2.41}$$

Algorithm Derivation, Resampling Stage Combining the Chapman-Kolmogorov Equation 2.38 with the assumption of a Markovian system (Equation 2.12), and exploiting the principle of Importance Sampling through the posterior pdf representation given by Equation 2.41, Algorithm 1 can be derived to incrementally estimate an approximation of $p(x_t|z_{1\dots t})$. Together with an appropriate initialization of the prior state distribution (i.e., an appropriate setting of

Algorithm 1: SIS Particle Filter - Incremental step

```
for $i = 1 \dots N_s$ do
    sample $x_t^i$ according to $q(x_t^i|x_{t-1}^i, z_t)$
    $\hat{w}_t^i \leftarrow w_{t-1}^i \frac{p(z_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{t-1}^i, z_t)}$
$w_t^i \leftarrow \frac{\hat{w}_t^i}{\sum_j \hat{w}_t^j} \qquad \forall i = 1 \dots N_s$
$N_{eff} \leftarrow (\sum_{i=1}^{N_s} (w_t^i)^2)^{-1}$
if $N_{eff} <$ resampling threshold then
    resample $x_t^1 \dots x_t^{N_s}$ according to ddf $w_t^1 \dots w_t^{N_s}$
    $w_t^i \leftarrow N_s^{-1} \qquad \forall i = 1 \dots N_s$
```

$x_0^{1 \dots N_s}$) it forms the basis of the Sequential Importance Sampling (SIS) Particle Filter algorithm.

The conditional resampling stage is due to a common problem with the SIS particle filter, namely the *degeneracy* phenomenon: after a few iterations, all particles but one typically have negligible weight (it can be shown that the variance of the importance weights is non-decreasing over time). Intuitively, resampling replaces a random measure of the true distribution with an equivalent one (in the limit of $N_s \to \infty$) that is better suited for the inference algorithm. A common measure of degeneracy, commonly referred to as *effective sample size*, is estimated as N_{eff}, and provides a rough estimation of the number of "active" particles. Different resampling strategies are characterized by peculiar computational requirements and statistical properties, as argued in [Douc and Cappé, 2005, Hol et al., 2006].

One iteration of the algorithm can be computed, when using efficient resampling schemes such as Systematic Resampling, in $O(N_s)$. Computational requirements do not depend on the size of the reference process, thus allowing a sequence of T observation to be decoded in $O(N_s T)$ (as opposed HMMs, where the decoding algorithm was quadratic in the number of states).

Smooth Inference: the Forward-Backward and Viterbi Algorithms

As was the case for Hidden Markov Models, it is possible to derive a counterpart of the Forward-Backward and Viterbi Algorithms in the context of Particle Filtering methods, in order to perform inference using the whole observation sequence at each instant.

The state pdf at time t, considering the whole observation sequence $z_1 \dots z_T$, can be factored as:

$$p(x_t|z_{1\dots T}) = \int p(x_t, x_{t+1}|z_{1\dots T}) dx_{t+1} \tag{2.42}$$

$$= \int p(x_{t+1}|z_{1\dots T}) p(x_t, x_{t+1}|z_{1\dots t}) dx_{t+1} \tag{2.43}$$

$$= p(x_t|z_{1\dots t}) \int \frac{p(x_{t+1}|z_{1\dots T}) p(x_{t+1}|x_t)}{\int p(x_{t+1}|x_t) p(x_t|z_{1\dots t}) dx_t} \tag{2.44}$$

The forward-backward algorithm makes use of the incremental inference algorithm to perform a first pass on the observation sequence; the resulting random measure $\{x_{1\dots T}^i, w_{1\dots T}^i\}_{i=1}^{N_s}$ is then used to compute the *smoothed importance weights* $\{w_{1\dots T|T}^i\}_{i=1}^{N_s}$ as follows:

$$w_{T|T}^i = w_T^i \qquad i = 1 \dots N_s \tag{2.45}$$

$$w_{t|T}^i = w_t^i \left[\sum_{j=1}^{N_s} w_{t+1|T}^j \frac{p(x_{t+1}^j|x_t^i)}{\sum_{k=1} N_s w_t^k p(x_{t+1}^j|x_t^k)} \right] \qquad i = 1 \dots N_s \tag{2.46}$$

Computation of Equation 2.46 costs $O(N_s^2)$ to perform; it can however be reduced using approximations for certain classes of models, as described in [Klaas et al., 2006].

The Maximum A Posteriori path can be computed using the Viterbi algorithm on the $\{x^i_{1\ldots T}\}_{i=1}^{N_s}$ grid obtained through SIS inference. Again, its cost is quadratic in the number of particles N_s, resulting in computational requirements in the order of $O(N_s^2 T)$.

2.4.2 A Unified Approach for Music Content Alignment

This section builds upon [Montecchio and Cont, 2011b], which introduced a generic framework for music alignment. Other models for audio-to-score alignment that appeared around the same time make use of similar techniques, namely [Otsuka et al., 2011] and [Duan and Pardo, 2011]. A major advantage of the proposed approach is the definition of a unified methodology allowing the real time alignment of audio to both a symbolic score and an audio reference, and possibly a *combination* of the two.

Alignment Model

Given a music stream and a reference medium, in the form of either a symbolic score or an audio recording, the alignment problem is reformulated as a tracking problem, where the current position of the audio stream along the reference is modeled using physical motion equations. This is possible thanks to the exploitation of a *continuous* representation of the position dimension associated to a symbolic score: using a continuous variable not only provides a unified structure for both problems of symbolic and audio alignment, but also allows to deal with notated references in musically meaningful units.

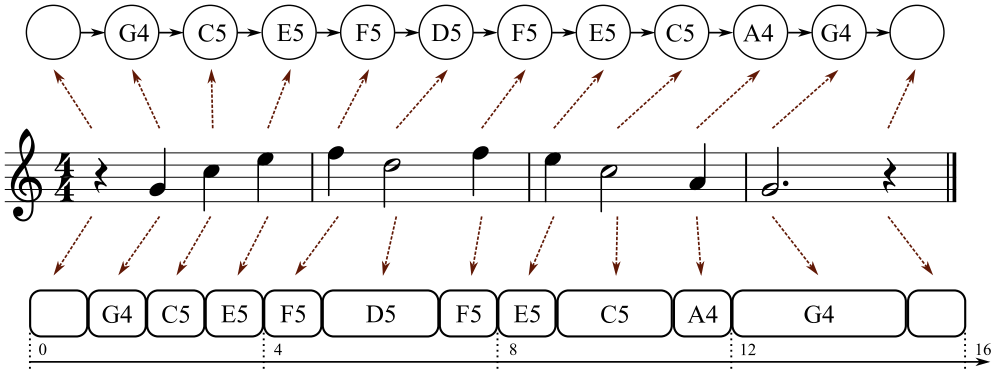


Figure 2.11: Discrete and continuous representations of symbolic scores.

The system state at time t is modeled as a two-dimensional vector x_t, representing the current position along the reference and execution speed (tempo). In the case of audio to audio alignment, position is measured in seconds and execution speed corresponds to the playback speed ratio; in the case of symbolic score references, position is measured in (fractional) quarter notes from the beginning of the score, and tempo in quarter notes per second (i.e. bpm/60), as in Figure 2.11. An additional advantage with respect to HMM-based systems, in the context of symbolic alignment, is that the continuous modeling of events with short duration does not present issues related to the number of states used.

The incoming signal processing frontend is based on spectral features extracted from the FFT analysis of an overlapping, windowed signal representation, with hop size ΔT. A discrete filterbank frontend such as the one described in Section 2.3.3 could in principle be adopted in the case of symbolic alignment.

State Transition Modeling The state transition likelihood $p(x_t|x_{t-1})$ makes use of tempo estimation in the previous frame and assumes that tempo is stationary:

$$p(x_t|x_{t-1}) = \mathcal{N}\left(x_t \;\middle|\; \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} x_{t-1}, \begin{bmatrix} \sigma_1^2\,\Delta T & 0 \\ 0 & \sigma_2^2\,\Delta T \end{bmatrix}\right) \tag{2.47}$$

Intuitively, this corresponds to a performance where it is expected that the tempo is rather steady but can fluctuate; the parameters σ_1^2 and σ_2^2 control the variability of tempo and the possibility of local mismatches that do not affect the overall tempo estimate (e.g., a single note played with some delay).

Observation Modeling The function $p(z_t|x_t)$ represents the likelihood of observing an audio frame z_t given the current position along the reference, expressed in x_t. A simple spectral similarity measure is considered, using the Kullback-Leibler (KL) divergence formula[4] between two normalized vectors u, v of length L:

$$D_{KL}(u||v) = \sum_{i=1}^{L} u_i \ln \frac{u_i}{v_i} \qquad\qquad \sum_{i=1}^{L} u_i = \sum_{i=1}^{L} v_i = 1, u_i > 0, v_i > 0 \;\forall i \tag{2.48}$$

Using this measure, the spectrum at frame t is compared to the spectrum referenced by x_t in the reference audio, or, in the symbolic case, a template spectrum associated to the score event which is active at the score position referenced by x_t. The resulting observation likelihood is an exponentially decreasing function of the KL divergence between the considered spectra.

Proposal Distribution Thanks to the availability of efficient sampling procedures for normal random variables, the proposal distribution function $q(x_t|x_{t-1}, z_t)$ can sample directly according to the state transition distribution, dispensing an implementation from explicitly evaluating Equation 2.47. Particle Filtering approaches that exploit similar techniques are called *Condensation* (CONditional DENSity propagATION) algorithms.

Decoding The decoding of position and tempo is carried out by computing the expected value of the resulting random measure as $\sum_{i=1}^{N_s} x_t^i w_t^i$. Alternatively, in the case of a predefined segmentation of the reference medium (this is obtainable easily in the case of symbolic scores) it is possible to compute a discrete probability distribution for each individual event by summing the related weights w_t depending on their position in the reference s_t.

2.4.3 Estimation of Adaptive Gesture Features using Temporal Profiles

The proposed approach was introduced in [Caramiaux et al., 2012]; it was developed as an improvement of the model proposed in [Bevilacqua et al., 2010], reviewed in Section 2.3.4.

Other models based on particle filtering techniques have been proposed in the literature, such as [Black and Jepson, 1998a] and [Visell and Cooperstock, 2007]. An explicit advantage of the presented model with respect to those and other systems based on similar techniques is that it explicitly models a series of signal features, making the system not only invariant to, but also capable of estimating features such as rotation, speed and scale of the gesture execution.

[4] In this context, the KL divergence formula is devoid of any Information Geometrical meaning, and is used because of its capacity to capture spectral similarities better than cosine similarity.

Alignment Model

The state of the system is composed of the position along the reference gesture, and a series of other features; its particular form depends on the application and on the input data representation. Let $x_t \in \mathcal{R}^D$ be the system state at instant t, and D the dimensionality of the state space. With p_t we denote the component of the state vector x_t associated to the normalized position along the reference:

$$\text{beginning of the gesture} \quad 0 \leq p_t \leq 1 \quad \text{end of the gesture} \tag{2.49}$$

State Transition Modeling The state transition probability density function (transition pdf) is the analogous to the HMM state transition matrix, and is defined as a Gaussian distribution whose mean depends on the previous system state:

$$p(x_t|x_{t-1}) = \mathcal{N}\Big(x_t\big|Ax_{t-1}, \text{diag}\Big(\sigma_1 \ldots \sigma_D\Big)\Big) \tag{2.50}$$

Considering the dimensions of the state vector p and v, respectively related to position and speed, the corresponding entries in the transition matrix are set to

$$A_{p,v} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \tag{2.51}$$

so that the behavior of the system resembles the motion equations of the music alignment model of Equation 2.47. Moreover, the simple form of the distribution again allows for direct sampling of $q(x_t|x_{t-1}, z_t)$, which is useful (although not necessary) in the implementation of the algorithm.

Observation Likelihood The observation likelihood is based on the Mahalanobis distance. Particle Filtering-based system allow for non-linear functions $f(x_t, g(p_t))$ of the inferred state value x_t and the expected template sample $g(p_t)$ given the current estimated system position p_t, computed interpolating the sampled feature points of the reference gesture. In [Black and Jepson, 1998b], a simpler probability distribution was defined, taking into account scale and velocity. Here, a distance $d(z_t, f(x_t, g(p_t)))$ is defined by:

$$d(z_t, f(x_t, g(p_t))) = \sqrt{[z_t - f(x_t, g(p_t))]^T \Sigma^{-1} [z_t - f(x_t, g(p_t))]} \tag{2.52}$$

While a Gaussian distribution was used in [Black and Jepson, 1998a], the proposed model involves a Student's T distribution that depends on three parameters: the mean μ, the covariance matrix Σ and the degree of freedom ν. For a D-dimensional input vector z_t at time k, the Student's T distribution probability density function is evaluated as:

$$St(z_t|f(x_t, g(p_t)), \Sigma, \nu) = \frac{1}{Z(\Sigma, \nu)} \big(1 + \frac{d(z_t, f(x_t, g(p_t)))^2}{\nu}\big)^{-\frac{\nu+D}{2}} \tag{2.53}$$

$$\frac{1}{Z(\Sigma, \nu)} = \frac{\Gamma(\nu/2 + D/2)}{\Gamma(\nu/2)} \frac{|\Sigma|^{-1/2}}{(\nu\pi)^{D/2}} \tag{2.54}$$

The rationale behind such choice is to allow for heavier distribution tails; it is then possible to explicitly account for samples that are generated far from the expected position in the modeled process. It should be noted that the factor in Equation 2.54 does not need to be explicitly computed, as it is constant and thus irrelevant because of the normalization step in the Particle Filtering update algorithm. The computational complexity is thus similar to that of evaluating a Gaussian distribution, and the Student's T distribution reduces indeed to a Gaussian distribution in the case $\nu \to \infty$, with mean μ and covariance Σ.

Extension to Recognition

The approach can be extended in order to perform simultaneous alignment and recognition, introducing an additional state dimension h associated to an integer variable representing an index over the gesture vocabulary. It is then sufficient to modify Equation 2.52 so that the distance function associated to each particle is computed according to the indexed template gesture: in the case of the i-th particle, $d(z_t, f(x_t^i, g_{h_t^i}(p_t^i)))$.

The index h associated to a particle does not change over time. It should be noted that this does not break the discussed probabilistic formulation, as the proposal normal sampling function can be thought of as concentrating all its mass in a single point; formally, for the i-th particle,

$$p(h_{t+1}^i | x_t = \left[\ldots, h_t^i\right]^T, z_t) = \delta(h_{t+1}^i - h_t^i) \tag{2.55}$$

Moreover, thanks to the resampling step in Algorithm 1, no computational power is wasted, as the particles corresponding to incorrect gestures, which tend to have low importance weights, are eventually “resampled into” the correct hypotheses.

Adaptation of the Model to Different Gesture Signals

We consider as a starting point the case of two-dimensional gestures, such as those obtainable using a touchscreen or a graphic tablet, and show how the proposed model can be extended to other signal types via an appropriate choice of the system state model and observation likelihood function.

Two-Dimensional Gestures In [Wobbrock et al., 2007], the authors proposed a preprocessing step that roughly consists in rotating, scaling and translating the input and template gestures before computing the Euclidean distance; either the rotation angle and the scaling coefficient are invariants in the recognition process. The proposed model on the other hand allows for taking into account these invariants explicitly, by defining them as state variables. Therefore, the gesture features estimated at each time instant k are the following: position p_t, velocity v_t, rotation angle r_t and scaling coefficient s_t:

$$x_t = \begin{bmatrix} p_t & v_t & r_t & s_t \end{bmatrix}^T \in [0, 1] \times \mathbb{R}^3 \tag{2.56}$$

The state transition matrix A conveys the idea that rotation and scaling of a gesture are roughly stationary between subsequent sampling time; it is therefore defined as:

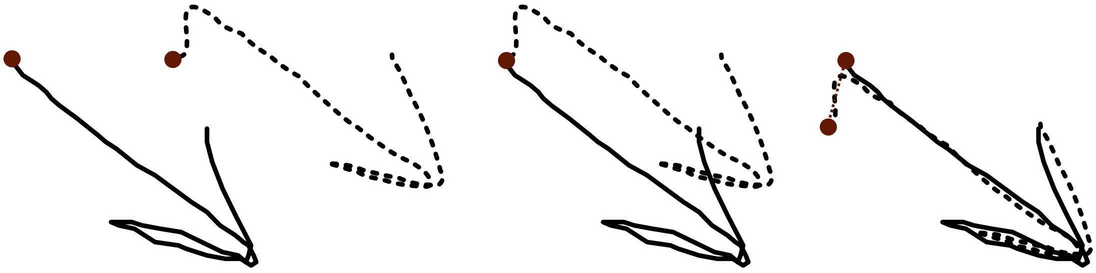
$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.57}$$

The function which expressed the expected position of the point $g(p_t)$ after rotation and scaling according to the state vector x_t is given by:

$$f(x_t, g(p_t)) = \mathcal{I}_2 s_t \begin{bmatrix} \cos(r_t) & -\sin(r_t) \\ \sin(r_t) & \cos(r_t) \end{bmatrix} g(p_t) \tag{2.58}$$

Modeling the Coordinate System of the Reference Gesture The approach presented above assumes that both the input and template gestures are captured with respect to the same coordinate system; in other words, it assumes that corresponding gestures can be superimposed without translation. In case this assumption is violated, one possible solution is to consider the alignment of the discrete *derivative* of each time sequence; however, significant information regarding absolute spatial positioning is lost.

An alternative, most commonly used technique consists in translating both the input and template gesture so that the first coordinate of each time series coincides. Even though the approach works in the majority of the cases, in some situations such simple translation fails to capture a valid reference point, as in the example pictured in Figure 2.12.



(a) Original coordinate system. (b) Naive translation. (c) Appropriate translation.

Figure 2.12: Translation of gestures captured in different coordinate systems. The highlighted point marks the beginning of the gestures.

The introduction of additional state variables representing spatial offsets is a viable solution, although its efficiency is limited because of the dimensionality of the observation space (this is especially true in the case of three-dimensional gestures). In the proposed approach, a single state variable R is introduced, limited in the range $[0, 1]$. The reference translation point is chosen according to the point expressed by $g(R)$; in order to privilege the initial points in the gesture, an appropriate prior distribution is expressed, such as a (negative) exponential distribution. This representation strategy allows to optimize the efficiency of the approach, by concentrating the computational effort on a restricted subspace of the state.

Extension to Higher-Dimensional Gestures In order to extend the approach to three-dimensional gesture signals, representing absolute spatial position, it is trivial to modify Equation 2.58 in order to use additional rotation angles (and eventually non-uniform axes scaling).

Signals captured from accelerometer devices do not suffer from issues related to coordinate systems, because of their implicitly differential nature. On the other hand, an additional problem is represented by the presence of a constant bias in the signal introduced by gravity. This issue often presents itself in the case of hand held devices, such as game controllers; in that case, a different orientation of the devices manifests itself in the signal as an offset. In the proposed model, this issue can be taken care of by the introduction of state variables representing offsets. The computational load introduced by the additional complexity can be limited by considering an appropriate prior distribution on the offsets, according to realistic grasps on the device.

CHAPTER

THREE

IDENTIFICATION

The problems of identification and alignment are closely related: it could be argued that stating whether or not two music recordings are “the same”, or refer to the same underlying “music idea”, is equivalent to deciding whether the recordings share sufficiently similar patterns of common traits. It is not surprising that a vast number of works in the music identification literature makes use of alignment techniques that were the subject of the preceding chapter.

Research on music identification has immediate applications in the industry: it is common practice for copyright supervision and Intellectual Property management agencies to monitor radio and TV stations, as well as Internet broadcasts. The problem of music identification is then of particular appeal in the context of large data sets, where the efficiency of the proposed approaches assumes a role of primary importance.

Acoustic fingerprinting is the subject of Section 3.1, a family of techniques aimed at identifying *copies of a recording*. Subsequent sections introduce identification strategies that attempt at spotting different *versions of a music work*. The relevant literature is briefly reviewed in Section 3.2, focusing on techniques that consider efficiency a primary goal. Section 3.3 proposes a system aimed at efficient retrieval of content-based music queries in large scale collections.

3.1 Near-Duplicate Recording Detection

Acoustic Fingerprinting techniques aim at identifying audio recordings using condensed digital summaries of the acoustic content; identification should be carried out regardless of compression, interference or distortion in the signals.

The key aspect exploited by fingerprinting systems is the consistency of particular acoustic features even under noise presence. It is generally believed that most important perceptual audio features live in the frequency domain [Haitsma and Kalker, 2002]; even under distortion many relevant features in the spectrogram are preserved, thus many compact digests (fingerprints) computed for small audio regions are then matched efficiently using indexing approaches. A similar approach is adopted in [Wang, 2003] where prominent spectral features are extracted in pairs, storing their frequencies along with their time stamps; two recordings are compared efficiently via a histogram measure of the locations of matching frequency pairs.

A strategy based on computer-vision is presented in [Ke et al., 2005], which transforms a music identification problem into the retrieval of corrupted sub-images. A review of fingerprinting approaches is presented in [Cano et al., 2005].

In spite of the already high accuracy rates that characterize commercial implementations, research in the domain of acoustic fingerprinting is nonetheless is still active. In [Ramona and Peeters, 2011] the K-Nearest Neighbors (KNN) algorithm is used to perform retrieval on a reduced number of fingerprints computed using larger time windows for audio analysis. A further proof of the interest for fingerprinting methods is given by the abundance of related open-source implementations: worth of note are AcoustID[1], The Echo Nest[2] and Last.fm[3].

3.2 Identification of Versions of Music Works

Audio fingerprint approaches aim at identifying a particular recording; alternate renderings of a composition are considered different artifacts and, in most of the cases, not recognized as different instances of the same music work.

The automatic identification of different performances of the same work, often referred to as *cover song identification* task by the Music Information Retrieval community, must take into account a high degree of variability between alternative performances. [Serrà, 2011] presents a thorough review of the musical facets that can be affected in the process. The investigation of what can represent a meaningful musical variation, the result of which is still perceived as closely related to the original content, is the object of open debate; moreover, the factors that determine the outcome of music similarity judgments are largely dependent on the cultural background of the listener. Without going into detail, it is sufficient to consider the common example of live performances of studio recordings: even when interpreted by the same musicians, a work may be played with a different tempo, in a different key, with different instruments, and introducing significant changes in the chorus-verse structure; in addition to that, additional music material might appear such as solos, intros, and codas.

Using an analogy with passage-level Information Retrieval techniques, [Casey and Slaney, 2006] argues that temporal sequence information cannot be discarded when searching musically similar passages within a narrow range of musical styles or within a single musical piece. The search for similar sequences in different recordings is often carried out with alignment techniques, in particular those that consider the alignment of *partial subsequences*. The use of Dynamic Programming techniques for computing the similarity between two recordings based on the quality of alignment of tonal descriptors (chroma features in particular, discussed in Section 2.1)

[1] `http://acoustid.org/chromaprint`

[2] `http://echoprint.me/`

[3] `https://github.com/lastfm/Fingerprinter`

recur often in the identification literature: [Hu et al., 2003] (using MIDI files as reference), [Gómez and Herrera, 2006], [Serra et al., 2008].

A significant advantage of alignment techniques is their robustness to tempo differences. A more efficient alternative is the use of beat-synchronous descriptors in conjunction with simpler similarity measures, as in the case of [Ellis and Poliner, 2007], [Ellis et al., 2008].

The application of a post-processing step to a chroma sequence, inspired by techniques for the extraction of rhythm-related descriptors [Lidy and Rauber, 2005], is a key aspect of [Jensen et al., 2008], where a Fourier transform is applied to the rows – each corresponding to a pitch class – of the chromagram representation of a recording in order to obtain a tempo invariant descriptor by discarding phase information.

Instead of comparing sequences by computing their alignment, [Serrà et al., 2011] proposes to fit a time series *model* to the music descriptors of one recording, and measure its capability to *predict* the other recording; the authors investigate issues related to predictability also for relatively long intervals.

An indirect mechanism for detection of repeated sequences is proposed in [Ahonen, 2010], where similarity is a function of the Normalized compression distance (NCD) [Cilibrasi and Vitányi, 2005] metric. The idea is to measure the information in a sequence using Kolmogorov complexity (the length in bits of the shortest binary program that produces the sequence as an output), that can be approximated using standard lossless data compression algorithms.

Efficient Music Identification Techniques

Several works in the literature have dealt explicitly with the issue of efficiency in content-based music identification outside the domain of audio fingerprinting. A crucial component of most of these works is *vector quantization*, the process of mapping a (continuous) space of vectors into a *codebook*, a finite set of elements (*codewords*).

In [Kurth and Müller, 2008] the codebook selection, associated to the space of chroma vectors with unitary euclidean norm (the 11-dimensional sphere), is performed using an adaptation of the LBF algorithm [Linde et al., 1980]. An *index* structure is subsequently used to store the quantized versions of the original feature sequences, which are in turn used to perform similarity computation using fuzzy matching techniques.

In [Riley et al., 2008] a codebook is selected using the popular K-Means algorithm, used to turn chroma feature sequences into a *histogram* representation. Locality Sensitive Hashing (LSH) techniques, reviewed below, are used to perform efficient search in the new space. LSH is also used in [Marolt, 2008], which proposes an efficient algorithm for performing retrieval according to the similarity of beat-synchronous *melodic* representations extracted from the audio signal.

A methodology for measuring content-based music similarity, inspired by LSH approaches, is the subject of [Casey et al., 2008]. The related AudioDB[4] system [Rhodes et al., 2010] provides a framework for managing large collections of multimedia data exploiting content-based retrieval methods.

Locality Sensitive Hashing is a framework that aims at probabilistic dimension reduction of high-dimensional data. The fundamental idea is to hash the input elements so that similar items are mapped to the same buckets with high probability (the number of buckets being much smaller than the universe of possible inputs). Originally introduced in [Gionis et al., 1999] for performing similarity search in high dimensional spaces, it was extended [Datar et al., 2004] to make use of more sophisticated probabilistic techniques. A tutorial on the application of LSH techniques in the multimedia domain is presented [Slaney and Casey, 2008].

[4] `http://www.omras2.org/audioDB`

3.3 An Efficient Music Identification Engine

This section describes a music identification engine initially introduced in [Di Buccio et al., 2010b], [Di Buccio et al., 2010c] and subsequently refined in [Montecchio et al., 2012]. The identification task is addressed by way of a comprehensive methodology that includes an audio content analysis procedure to improve the extraction of standard audio content descriptors and a computation of similarity among audio documents inspired by classic text Information Retrieval methods [Manning et al., 2008].

3.3.1 Overview of the Methodology

The proposed methodology is based on a two-level representation. At the first level, a recording is subdivided into a sequence of shorter segments; this aims at loosely capturing a possible structure in a music work, due for example to repetitions of a theme or modulations. Since automatic identification of the constituting elements of a song is still a subject of open research, a simplified segmentation approach is exploited, dividing a recording into overlapping excerpts of fixed length.

Content-based descriptors are subsequently extracted from each of the resulting segments, obtaining the second representation level. Ideally, a representation of the audio signal contained in a segment should make use of musically meaningful units, e.g. bars or beats. Since the extraction of such units is still an open issue, the proposed approach consists in computing content-based descriptors at equally spaced time intervals. In line with most of the approaches to music identification, the adopted descriptors are *chroma features*, reviewed in Section 2.1.2. Chroma extraction is preceded by *tuning frequency adjustment* in order to be robust to adoption of different reference frequencies, and is followed by a *key finding* procedure, to deal with alternative versions of the same music work performed in different keys.

The resulting chroma vectors are not directly exploited as music descriptors to perform identification, being instead transformed into an integer value by means of a *hashing* function. The integer value corresponding to a chroma vector depends on the energy distribution among the pitch classes. This simple hashing function has the same aim of more complex LSH approaches, and is formulated using explicit prior knowledge on music signal modeling.

The representation steps map a recording into a sequence of segments, each represented as a sequence of integer hashes. Both the recording to be identified and the recordings in the collection – hereafter named respectively *query* and *documents* in accordance with the terminology adopted by the Information Retrieval community – undergo the same three step representation processing. A similarity function is subsequently applied to each query-document pair, where documents are ranked in decreasing order of the similarity score to them assigned by the function. The basic intuition behind the similarity function is that similar recording present a similar energy distribution among pitch classes, and therefore have many hashes in common. Similarity computation exploits the two-level representation of audio in order to combine the information of *local* resemblance between segments.

An overview of the methodology is depicted in Figure 3.1. Each of the following sections provides an in-depth description of the blocks constituting the complete approach.

3.3.2 Audio Analysis

Audio analysis is the process of converting the audio content into a sequence of chroma features; in order to improve the accuracy of subsequent identification steps, particular attention is directed to the estimation of the tuning frequency and identification of the key of the performance.

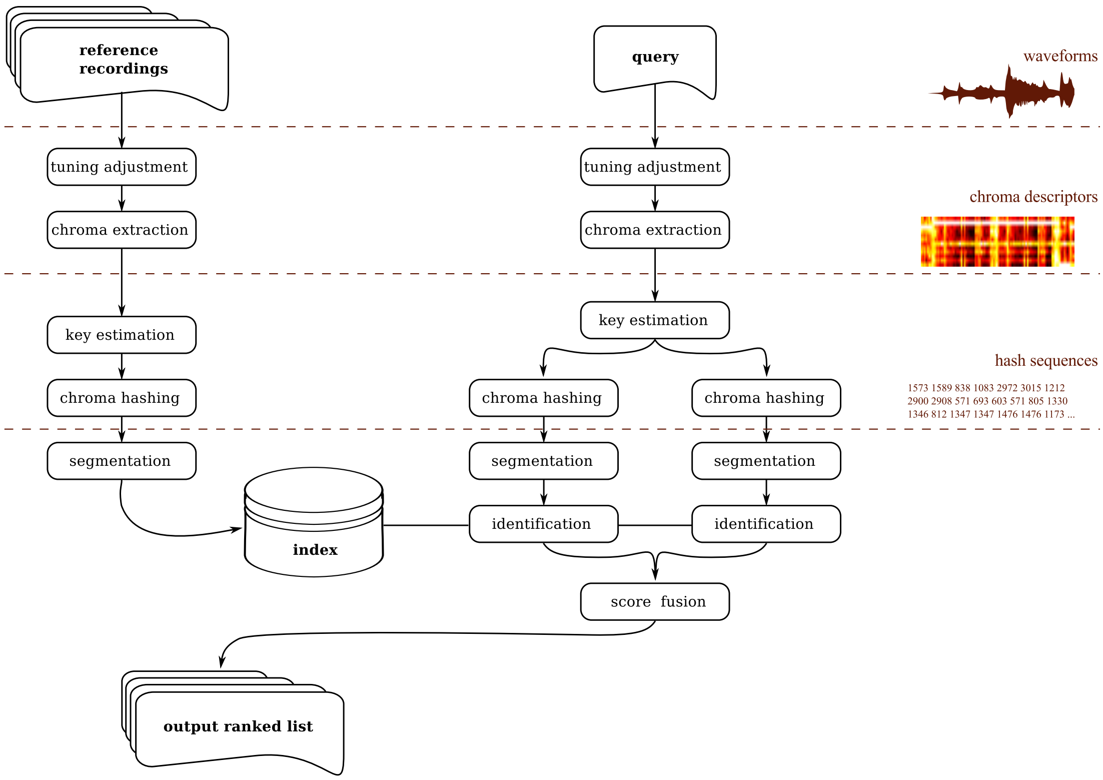


Figure 3.1: Architecture of the proposed music identification methodology.

Tuning Frequency Estimation

As discussed in Secion 2.1.2, the exact pitch of notes is a function of a reference tuning frequency. Setting the A above middle C to 440 Hz is the common practice nowadays – the practice of setting the reference tuning frequency to 440 Hz being standardized in 1939 by the International Organization for Standardization (ISO) – however historically many different reference frequencies were used, with extremes ranging from 376.3 Hz to 570.7 Hz [Mendel and Ellis, 1969]; today most professional orchestras tune their instruments to a slightly higher pitch than 440 Hz.

Even though a limited offset of the tuning frequency from the standard 440 Hz does not lead to significant problems in typical MIR systems – a perfect tuning of each note is unrealistic for the human voice or any acoustic instrument – problems arise when the reference frequency is distant enough from the standard that a listener cannot decide whether notes should be classified according to a certain pitch class or to the one immediately above/below. In the literature the problem has not been dealt with extensively, with the exception of [Dressler and Streich, 2007], [Lerch, 2006].

The proposed tuning frequency detection algorithm works by first filtering the input signal according to different reference frequencies, which are then interpolated in order to produce a final estimation. An even number m of reference tuning frequencies $f_1 \dots f_m$, equally spaced on a logarithmic scale, is chosen (see Figure 3.2(a)) from a semitone-wide interval centered on 440 Hz. With respect to each of these frequencies, the input audio content is filtered according to a bank of narrow, semitone-spaced bandpass filters of the form of Equation 2.33, and the total energies of the resulting filtered signals are summed in order to get, for each candidate tuning frequency f_i, a measure of “fitness” to the input audio content e_i (Figure 3.2(b)). The underlying motivation is that in a realistic performance, in which most notes are played in tune, the output energies of the filter banks that are centered nearest to the actual tuning frequency should exhibit the highest values.

The estimation of the reference tuning frequency makes use of circular statistics, and in particular of the von Mises distribution (circular normal distribution):

$$p(\theta|\mu,\kappa) = \frac{e^{\kappa\cos(\theta-\mu)}}{2\pi I_0(\kappa)} \qquad \theta,\mu \in [0,2\pi), \quad \kappa \geq 0 \tag{3.1}$$

where μ and κ are analogous to the center and the variance of the normal distribution, and $I_0(\cdot)$ is the Bessel function of order 0. The choice of a periodic distribution is motivated by the assumption that an offset of an integer number of semitones in the initial m frequencies should not alter the resulting estimation (Figure 3.2(c)). In order to fit into this framework, the frequencies $f_1 \dots f_m$ are mapped to the angles $\theta_1 \dots \theta_m$ in the $[-\pi,\pi]$ range. Considering an integer quantization $\hat{e}_i$ for the energy e_i of a filtered signal as the number of realizations of θ_i, a maximum likelihood estimation can be performed to get the most probable tuning frequency (the mean μ of the distribution, which is also the mode) and a measure of confidence (a higher κ parameter corresponds to a sharper distribution, thus to a higher confidence). In particular, let $\hat{N} = \sum_{i=1}^{m} \hat{e}_i$ (i.e. $\hat{N}$ corresponds to the total number of samples), then

$$\hat{\mu} = \mathrm{Arg}\big(\sum_{i=1}^{m} \hat{e}_i e^{j\theta_i}\big) \tag{3.2}$$

$$\frac{I_1(\hat{\kappa})}{I_0(\hat{\kappa})} = \frac{1}{\hat{N}} \sum_{i=1}^{m} \hat{e}_i \cos(\theta_i - \hat{\mu}) \tag{3.3}$$

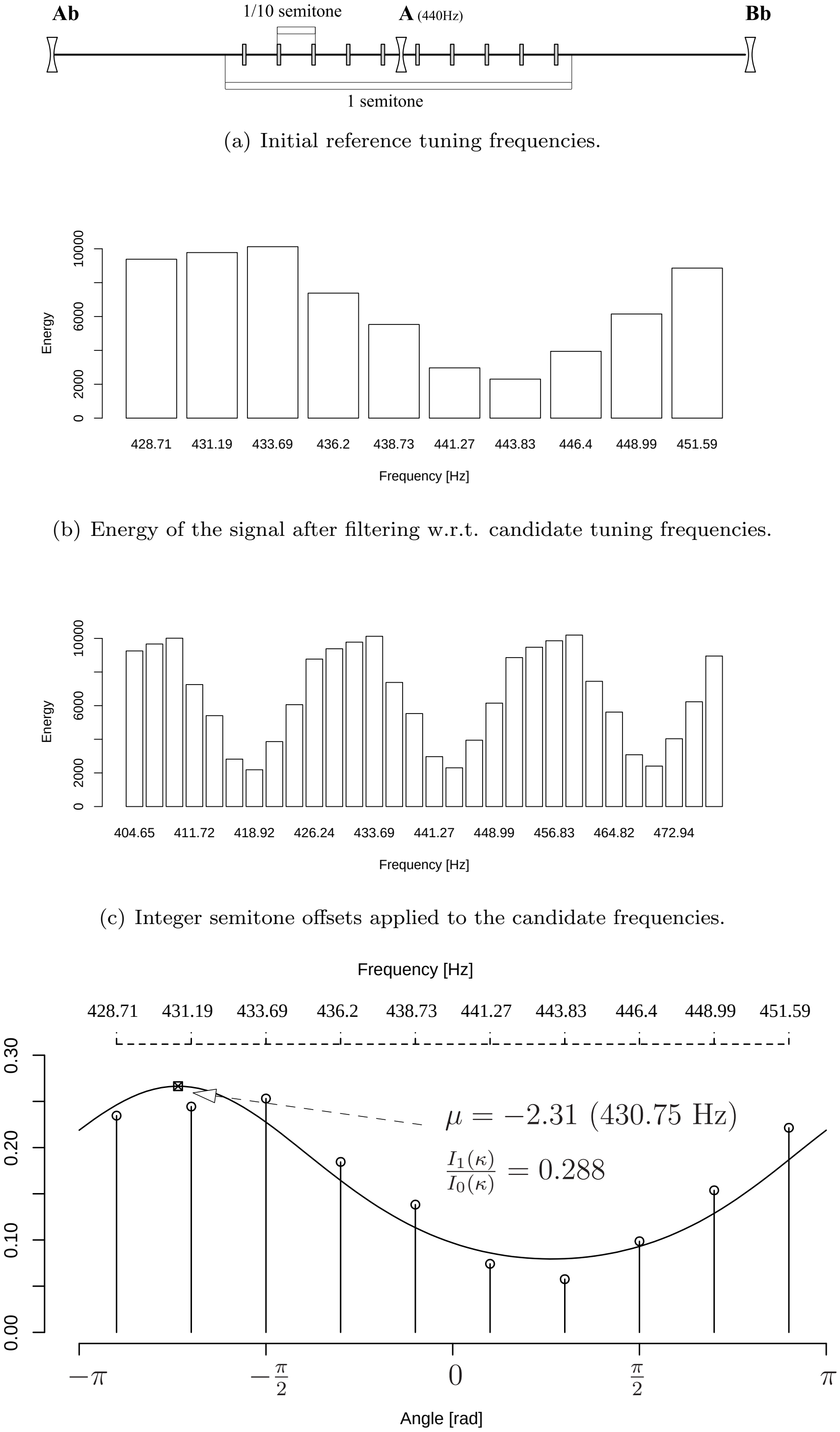
Taking the limit of the quantization step to 0 and normalizing the energy values with respect to their sum, the number of samples $\hat{e}_1 \dots \hat{e}_m$ can be replaced by the real energy values $e_1 \dots e_m$, yielding exact computations (although determining the κ parameter still requires numerical approximation because of the terms involving Bessel functions). The resulting fitted distribution is shown in Figure 3.2(d).

Key Estimation

A common problem when dealing with music identification is that different versions of the same music work can be performed in different keys. Performing the same query once for each possible transposition, and selecting the best match for each document in the collection, is both demanding computationally and less accurate than performing the query just once (in the correct key), because the possibility of identification errors is replicated.

A more refined approach makes use of the audio content in order to detect the key of the recording. It is important to note that, for the aims of music identification, it is only relevant to detect the *difference* of the keys for two versions of the same same music work. For instance, suppose that two recordings that are to be compared are, respectively, in C major and in G major: the only information that is needed is that the first recording is 7 semitones lower than the second. For this reason, it is important that a key-finding algorithm estimates keys consistently, even in case it assigns the same wrong key to two different recordings.

The first step towards key estimation is to produce a single chroma vector, which represents a harmonic *profile* for the whole recording. In order to detect the most probable key, an inner product is carried out between a suitably chosen vector of weights and all the possible transpositions (circular rotations) of this profile. The most probable keys for the recordings are finally extracted by sorting the transposition indexes according to the results of the inner products.



(a) Initial reference tuning frequencies.

(b) Energy of the signal after filtering w.r.t. candidate tuning frequencies.

(c) Integer semitone offsets applied to the candidate frequencies.

(d) Maximum Likelihood estimation of the distribution parameters.

Figure 3.2: Methodology for the estimation of the reference tuning frequency.

The vector of weights is learned in a supervised fashion from the audio content, using a randomized hill climbing approach: starting from random initial values, at each step the algorithm tries to maximize the number of recording pairs for which the relative key difference matches manual annotation by repeatedly sampling a weight vector from nearby points in the space; since the algorithm is unable to match all the key differences, a match is also tolerated for the second and third most probable hypotheses, at the cost of a higher penalty in the training function. This relaxation corresponds to a real scenario in which the recordings in the collection are transposed to their most probable key, and a query is executed by transposing it at most three times (instead of 12).

3.3.3 Efficient Identification

Audio descriptors are computed from the music signal in the form of chroma vectors that undergo further processing in order to obtain a compact, discrete representation. Such representation is the basis for an efficient computation of document similarity.

Chroma Feature Hashing

The quantized version q of a chroma vector c is obtained by taking into account the ranks of the chroma pitch classes, sorted by their values. Let r_i be the position in which the i-th component c_i would be ranked after a descending sort (starting from 0); a k-level rank-representation of c is constructed by considering a base 12 number computed as:

$$q = \sum_{i:r_i<k} i \cdot 12^{r_i} \tag{3.4}$$

For example, the 3-level quantization of the chroma vector depicted in Figure 2.5 yields the value $\tilde{q} = 10 \cdot 12^0 + 3 \cdot 12^1 + 5 \cdot 12^2 = 766$. This approach has been already applied to develop a clustering component for a statistical approach to classical music identification, as reported in [Miotto and Orio, 2008]. As was pointed out in [Miotto, 2011], the advantage of hashing music descriptors with an explicitly defined function, with respect to general LSH approaches, is that the resulting words do not depend on the statistical properties of a particular collection and therefore can be shared between different collections.

Similarity Computation

As described in Section 3.3.1, each recording is segmented into short excerpts of fixed length; the feature extraction and hashing steps turn the original acoustic representation into a sequence of segments, where each segment is represented by a sequence of hashes. Identification is carried out computing a similarity score among the query and the documents in the collection, and ranking the documents accordingly.

Let Q (D) denote a recording associated to a query (audio document in the database), composed of a sequence $\{q_1 \dots q_{|Q|}\}$ ($\{d_1 \dots d_{|D|}\}$) of hash sequences, each of length $|q|$ ($|d|$); the similarity between segments is computed as:

$$S(Q,D) = \underbrace{\sqrt[|Q|]{\prod_{q \in Q} \max_{d \in D} \Big\{ \underbrace{\sum_{t \in q \cap d} \min\Big(\frac{\mathrm{tf}(t,d)}{|d|}, \frac{\mathrm{tf}(t,q)}{|q|}\Big)}_{\text{Similarity at segment level}} \Big\}}}_{\text{Similarity at recording level}} \tag{3.5}$$

where $\mathrm{tf}(t,d)$ denotes the count (*term frequency*) of hashes with value t in segment d.

It is possible to conceptually distinguish two phases in the similarity computation procedure. The first step implies the computation of local similarity between segments d and q as the (normalized) number of terms they have in common

$$S_L(d,q) = \sum_{t \in q \cap d} \min\Big(\frac{\mathrm{tf}(t,d)}{|d|}, \frac{\mathrm{tf}(t,q)}{|q|}\Big) \tag{3.6}$$

while the second step aggregates the contributions of all the query segments, by computing the geometric mean of the best local similarity values for each query segment:

$$S(Q,D) = \sqrt[|Q|]{\prod_{q \in Q} \max_{d \in D} S_L(d,q)} \tag{3.7}$$

Implementation

An efficient implementation of the similarity computation procedure is possible because any information regarding the *ordering* of segments, and of the hashes contained therein, is discarded. This is reflected in the notation of Equation 3.5 by the exclusive use of *set operations.* A hash value in a segment is interpreted as a *term* in a *textual document.*

The remainder of this section describes how this parallel allows to exploit data structures and ranking algorithms commonly adopted in text retrieval, and adapt them to the context music identification. The proposed methodology has been implemented in FALCON[5], a software architecture built on top of the popular open source text search engine Lucene[6] and released under an open source license.

Data Representation The computation of the similarity score for a query-document pair requires information on the hash frequency in each query and document segment. Information on the frequency of occurrence of an hash in a specific query segment can be extracted at query time and efficiently accessed by means of data structure maintained in memory. The current implementation of the architecture exploits a list of maps, where each list entry (each map) corresponds to a segment and retains *[hash,frequency]* pairs.

Indexing Information on the frequency of an hash in a document can be efficiently accessed by means of an inverted index. An inverted index shares the same intuition of a book index. Let *vocabulary* denote the set of distinct terms appearing in the book. For a given term, the book index provides information on the pages where the term is used, allowing the reader to efficiently access the passages where the term is mentioned. In an inverted index data structure an *inverted list* is associated to each distinct term appearing in at least one of the documents in the collection; the entries of the inverted list are the (identifiers of the) documents where the term occurs. Moreover, additional information necessary by the ranking function can be stored in the inverted list entries, e.g. the frequency of occurrence of the terms in the documents or the positions where the terms occur.

In the adopted methodology, chroma hashes are the vocabulary entries, and each segment of a recording is interpreted as a textual document. More specifically, the proposed approach can be related to a segment-to-passage mapping, where a textual document is subdivided into passages, and inverted list entries maintain term information at a passage level. Figure 3.3 provides an example of the indexing process when applied to a recording, represented as a sequence of hashes. Such sequence undergoes a segmentation process where possibly overlapping subsequences are extracted from the recording sequence. After segmentation, a recording is

[5] `http://ims.dei.unipd.it/falcon`

[6] `http://lucene.apache.org`

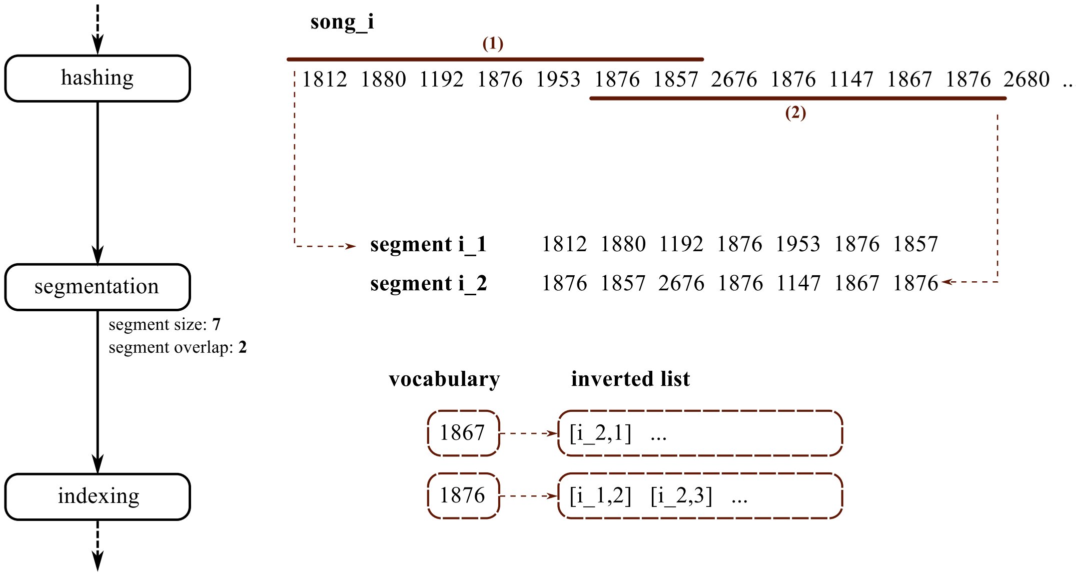


Figure 3.3: Segmentation and indexing of hashed chroma descriptors.

represented by a *set* of hash sequences. In accordance with the *bag of features* paradigm, it is hypothesized that information on hash occurrences at a segment level allows to effectively identify a recording; positional information is therefore ignored, thus each segment is effectively treated as a *set* of hashes. An inverted index can efficiently store hash occurrence information: the vocabulary is the set of distinct hashes in all the recordings in the collection, while the entries in the inverted list for a hash are the *[segment,frequency]* pairs for that hash.

Document At A Time Processing The adoption of an inverted index allows to compute efficiently the segment similarity score in Equation 3.6. When a query Q is submitted to the system it undergoes the same steps as the documents in the collection. The key finding algorithm described in Section 3.3.2 is applied to the query, thus obtaining diverse transpositions that the system treats as distinct queries, processed in parallel. After the computation of the hash values representing the audio content the query is split into a number of *segment-queries*. Each segment-query is processed by a Document At A Time (DAAT) strategy. DAAT strategies evaluate the contributions of every query term with respect to a single document before considering the next document. One advantage of the DAAT strategy is that it does not require intermediate document scores to be maintained during the entire ranking process, thus limiting the run-time memory usage.

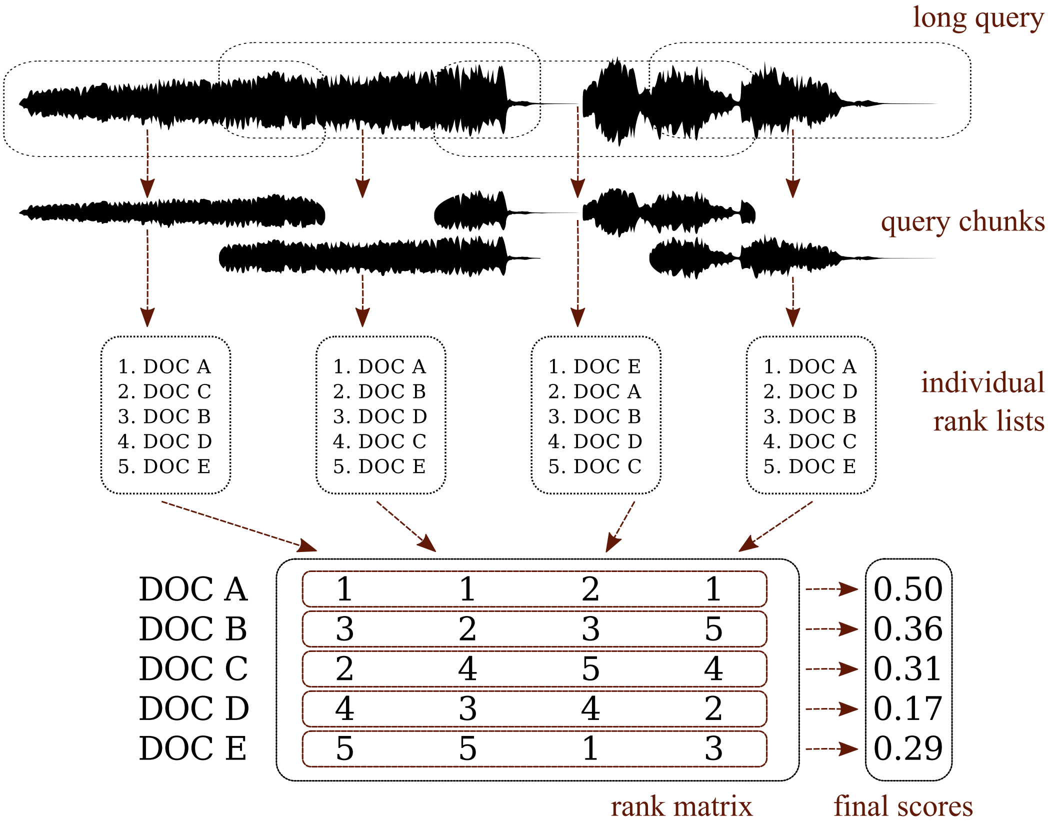
Score Fusion For each segment-query the system returns a ranked list whose entries are the document segments of all the documents in the collection ranked in decreasing order of similarity score at the segment level. The maximum among all the document segments is then computed by going through the returned ranked list. The next step consists in the aggregation of the results returned by the diverse segment-queries, according to Equation 3.7. Finally, the results obtained from the diverse considered transpositions are merged to obtained a final results list. An advantage of this retrieval strategy is that the diverse query (transpositions) and the diverse segment-queries can be processed in parallel.

3.3.4 Identification of Multiple Works Within a Single Query

The algorithm detailed above assumes that a query can be matched with a recording of the same music material. This is the typical scenario for cover song identification systems, however in order to handle the case of a query containing multiple distinct works – such as digitizations of tapes or vinyl discs containing several tracks – the methodology must be modified.

To this end, an additional time-resolution level was added to the segmentation hierarchy. The recording of a long query is divided in shorter elements – referred to as "chunks", in order to prevent confusion with the segments in which a single query or document is divided – each one in turn used as an individual query. The procedure returns a number of rank lists, that are merged into a structure called *rank matrix*.

The rationale behind this choice is related to the way the similarity between a query and the documents is computed. As it can be seen from Equation 3.7, it involves a maximization over indexed segments of the local similarities for each query segment. As a consequence, the system is designed to support short queries containing a portion of a corresponding recording; it is however ineffective in the case of long queries containing segments of different recordings, because the geometric mean is computed also from local scores with very low similarity values. It should be noticed that, while the maximization and averaging indexes of Equation 3.7 could be in principle reversed (there is no theoretical reason for the asymmetry of the global similarity function), the implementation described in Section 3.3.3 depends on this particular choice for performance optimization reasons.



DOC A	1	1	2	1	0.50
DOC B	3	2	3	5	0.36
DOC C	2	4	5	4	0.31
DOC D	4	3	4	2	0.17
DOC E	5	5	1	3	0.29

Figure 3.4: Identification of multiple works contained within a long query.

The similarity S_i, associated to the recording corresponding to the result indexed in the i-th row of the rank matrix, is computed with a simple data fusion approach:

$$S_i = \max_{w=1\ldots L-W} \sum_{k=w}^{w+W-1} \frac{1}{(\max(r_{i,j}, C))^2} \tag{3.8}$$

where L denotes the number of query chunks (columns of the rank matrix), and $r_{i,j}$ represents the position of recording i in the rank list produced from chunk j. The underlying idea is to consider each row of the rank matrix, and to analyze small windows of length W: an indexed recording spanning multiple chunks is supposed to consistently rank in high positions, and the windowing (along with the saturation constant C) acts as a filter for documents whose behavior in the ranking sequence is noisy.

The choice of using rank values instead of similarity scores is motivated by the consideration that averaging similarity scores works only if all segments belong to the same recording, whereas different recordings induce radically different similarity values. As for many data fusion approaches, the choice of rank values reduce the effect of large differences in the similarity scores. Moreover, the choice of ranks permits to be independent of the particular parametrization adopted for the lower-level local similarity computation.

CHAPTER FOUR

APPLICATIONS AND EXPERIMENTAL EVALUATION

The technologies reviewed in the preceding chapters have applications both in the context of multimedia computing and in more general scenarios. Below, four use cases were chosen in order to evaluate quantitatively the accuracy of the proposed methodologies as well as their efficiency and applicability in real-world situations.

Section 4.1 discusses real-time alignment of music from the perspective of artistic performances. Evaluation is carried out regarding both audio and symbolic facets, and issues related to the performance of contemporary music are taken into account. In Section 4.2 a new applicative context is introduced, aiming at accelerating the mixing phase in studio recording productions by automatization of repetitive tasks through the exploitation of alignment techniques. Gesture data is the subject of Section 4.3, which evaluates the accuracy of realtime alignment for two and three-dimensional data, as well as the simultaneous detection of the gestures from a set of templates. Finally, music is again the subject of interest in Section 4.4, which investigates the applicability of music identification strategies in the context of Cultural Heritage preservation in sound archive institutions. Alignment strategies are used in combination with efficient identification techniques to refine the accuracy of the approach while maintaining the ability to scale to large data sets.

4.1 Real-Time Musical Interactions

The use of electronics in the creation of music dates back to the early 20th century; the *Theremin* is widely considered to be the first electronic instrument (1920), and as soon as 1907 Busoni was envisaging the use of electrical sound sources in music for making use of microtonal intervals [Busoni, 1911]. During the past century *electronic music* evolved along with the technological progress, and both art and commercial music are nowadays strictly connected with the use of computers in the aspects of performance and composition.

A review of the history of electronic music is well beyond the scope of this thesis. It is however in the context of this art form that *musical interaction* between human performers and machines became a relevant subject of research. A role of primary importance is assumed by *score following* software systems, aimed at tracking the performance of a human player along a pre-determined score in real time.

Among the applications of score following systems, the most popular one is arguably that of automatic accompaniment. The first systems were developed with the precise aim *"to replace any member of the group by a synthetic performer so that the remaining live members cannot tell the difference [...] as an intelligent and musically informed collaborator"* [Vercoe, 1984]. Even assuming a perfectly accurate real-time tracking of a human performer, the synthesis of musically meaningful accompaniment is in itself a challenging problem. As musician *learn* and *rehearse* a piece, the *training* of a score following system using previous rehearsal is described in [Vercoe and Puckette, 1985], [Raphael, 2001], [Raphael, 2010]. In those works, a fundamental aspect is that of *timing*, which is learned according to previous performances. Another point of view is discussed in [Cont, 2008b], which investigates the *anticipation* of musical events.

Besides the use of real-time alignment techniques as "substitute" for a human player, several works investigate the exploitation of responsive computer software as an instrument that is part of the instrumentation ascribed in a music work. This approach is of particular interest for contemporary art music facilities: [Puckette and Lippe, 1992] provides an early report on the practical usage of score following systems in concert productions at IRCAM, where such systems were used as early as 1987 for synchronizing *live electronics* with different acoustic instruments (flute, clarinet, percussions, piano). An interesting discussion contained therein regards compromises that composers are often forced to make, to ensure that their music can be followed correctly: although these are seen as limitations that technological progress is supposed to overcome, it could be argued that works that make explicit use of a score following system simply involve an additional "instrument", subject to constraints that a composer should take into account much in the same way as any other acoustic instrument. The creative use of score following technologies and the new scientific questions that emerge out of their use in the context of artistic productions is the subject of [Cont, 2011].

4.1.1 Experimental Results

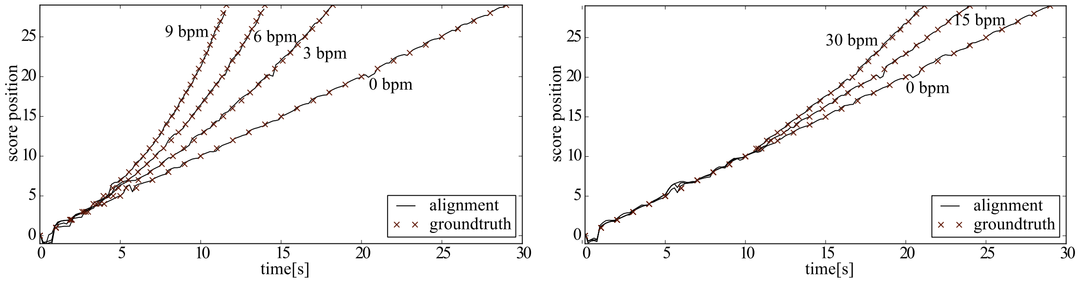
This section evaluates the accuracy of the proposed music alignment framework based on Sequential Montecarlo Inference, described in Section 2.4.2. Such system was found to be consistently superior to the HMM-based system described in Section 2.3.3, in particular regarding robustness to divergence in the parametrization from the optimal settings.

The first set of experiments investigates the ability to adapt to tempo changes using a synthesized note sequence, while the remaining ones make use of real music recordings.

Alignment to Synthetic Data

A random monophonic note sequence was synthesized using a sampled clarinet patch selected among the Garritan Personal Orchestra[1] library. As the alignments plotted in Fig. 4.1 show, tempo increases linearly at every quarter note (Fig. 4.1(a)) or suddenly (Fig. 4.1(b)), starting from 60 bpm. The average and maximum alignment error measures for the score events – defined as the delay or anticipation of the first detection of the event with respect to the nominal onset time – are reported in Table 4.1.

In all the tests a local misalignment can be seen around score position 5; this is due to the fact that two subsequent notes are repeated and no form of onset detection is used. Increasing the number of particles leads to a more robust tempo estimation and consequently avoids this problem. The average error is comparable to the analysis window hop size (12 ms).



(a) Incremental tempo increase

(b) Sudden tempo change

Figure 4.1: Alignment of a synthetic music track.

tempo variation	avg. error (s)	max. error (s)
steady	0.12	0.71
3 bpm linear	0.13	0.63
6 bpm linear	0.09	0.34
9 bpm linear	0.09	0.23
15 bpm sudden	0.12	0.71
30 bpm sudden	0.11	0.52

Table 4.1: Music alignment accuracy – synthetic data set.

Alignment to Symbolic and Audio References

The core of experimental collection is formed by recordings of Chopin's mazurkas annotated with the onset times for the events in the score; in particular, a subset of the collection created by C. Sapp for the Mazurka Project[2] was considered, consisting of 8 mazurkas for which 4 different performances were annotated.

The symbolic reference scores were extracted from MIDI files downloaded from the Web, thus increasing the difficulty of the alignment task because of the many imperfections that are often present, such as skipped notes and incorrect notation. Another significant issue is related to the intrinsic difficulty in formalizing music aspects like embellishments – which are extensively used in Chopin's music and realized differently by performers.

[1] `http://www.garritan.com`

[2] `http://www.mazurka.org.uk/`

	audio to score		audio to audio	
	avg. error [s]	max. error [s]	avg. error [s]	max. error [s]
Op. 6 n. 4	0.11	1.24	0.18	0.59
Op. 7 n. 2	0.16	2.65	0.18	1.06
Op. 17 n. 4	0.19	2.01	0.17	2.14
Op. 24 n. 2	0.19	3.89	0.29	9.41
Op. 30 n. 2	0.13	2.32	0.13	0.58
Op. 63 n. 3	0.18	2.89	0.15	1.23
Op. 67 n. 1	0.21	2.31	0.16	1.36
Op. 68 n. 3	0.34	4.42	0.19	2.12

Table 4.2: Music alignment accuracy – Chopin's Mazurkas.

Chopin mazurkas form a suitable test set because they are complex polyphonic pieces, challenging for state of the art score following softwares, and their executions are characterized by substantial tempo oscillations not explicitly notated in the score [Grosche and Sapp, 2010].

Results are provided in Table 4.2, using the same evaluation methodology used for the synthetic dataset; in the case of audio alignment, error is defined to be the euclidean distance of the ground truth label from the closest alignment point. A close analysis of the results, obtained inspecting the alignment plots, reveals that most of the times the maximum alignment error occurs on the very last beats of the pieces, characterized by a significant *rallentando*.

Tempo Change Issues

The analysis of the alignments of mazurka Op. 68 n. 3 pointed out a situation which is at the moment problematic for the score following system: between bars 31 and 32 there is a tempo change indicated by the composer, followed by 12 repetitions of the same chord that render an estimation of the new tempo not possible without some form of onset detection. As Figure 4.2 shows, the alignment is correct again when the melody starts, because of the added diversity to the harmonic content.

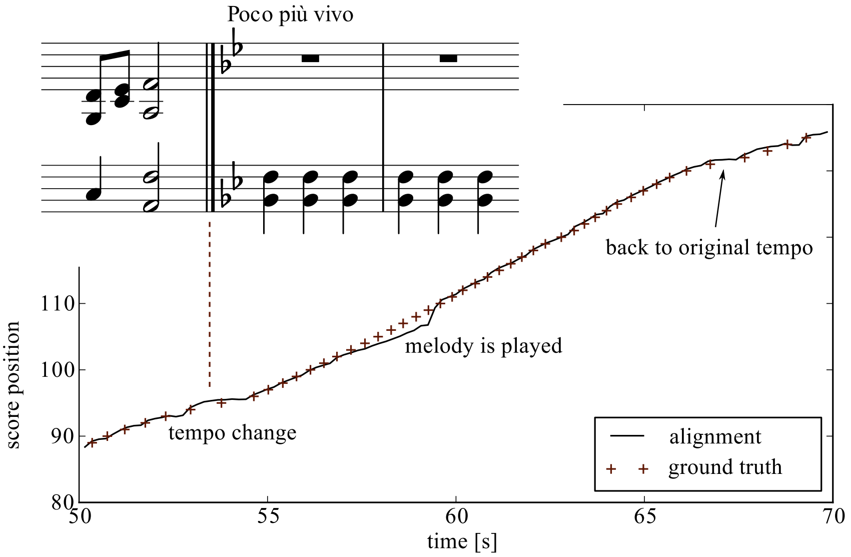


Figure 4.2: Uncertainty in tempo estimation.

Model Parametrization

The experimental results reported above make use of parametrization settings that are shared for all the experiments. The number of particles N_s is intentionally low (500 for the audio-to-score alignment evaluation, 200 for the audio-to-audio case) to present results that can be obtained on devices with low computational power; a fairly optimized version of the approach, running on a single core of an Intel 2.4 GHz i5 CPU, can support around 150 000 particles in real time, providing more accurate estimations.

4.1.2 Handling of Optional Repetitions

Even though recorded interpretations of classical music usually respect the composer's instructions regarding repetitions, performers often choose to skip some of those repetitions, as is the case for one of the recordings of mazurka Op. 7 n. 2. The particle filtering scheme is easily adapted to this situation, by allowing the particles to "jump" via a simple modification of the sampling step and transition pdf, as in Figure 4.3. Figure 4.4 compares two performances of the mentioned mazurka, where the skipped repetition is clearly visible.

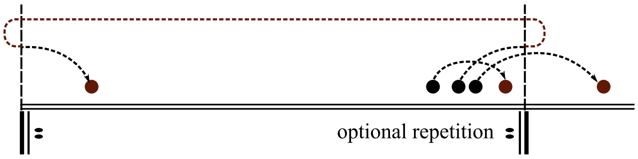


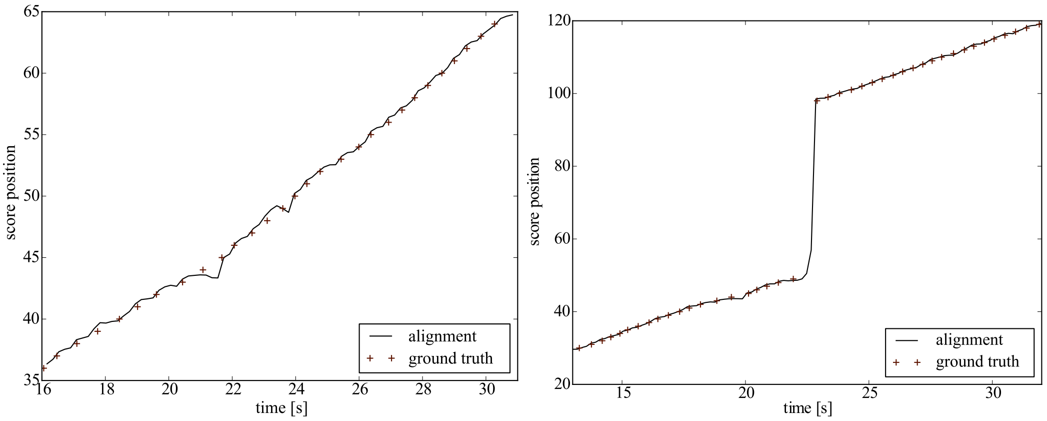
Figure 4.3: Handling optional repetitions within the Particle Filtering scheme.

The ability to follow optional repetitions can in principle be extended to the case where the performance can, at certain pre-defined points in the score, skip ahead (or go back) to other sections of the music, thus creating an *open form* interactive score, a common practice in contemporary music: this is the case of *Tensio*, for string quartet and electronics, by P. Manoury, an excerpt of which is depicted in Figure 4.5. In the final section of the work, the violin player has the option, after playing each bar, to choose among 12 possibilities (only 6 are pictured), each associated to different electronic processing, thus creating a different sonic experience at each performance.

4.1.3 Implementation of the Alignment Methodology

The proposed audio alignment methodology has been implemented in the form of two plug-in modules for the Pure Data[3] and Max/MSP[4] *dataflow programming* software environments. Both systems allow users to specify a program as a directed graph of the data flowing between operations (an approach introduced in [Puckette, 1988]); they are among the preferred software environments for music productions thanks to their intuitiveness. Figure 4.6 depicts the implementations of the score and audio following components in action.

[3] http://puredata.info
[4] http://cycling74.com



(a) Repetitions as in the score

(b) Skipped repetitions

Figure 4.4: Alignment of a music performance with optional repetitions.

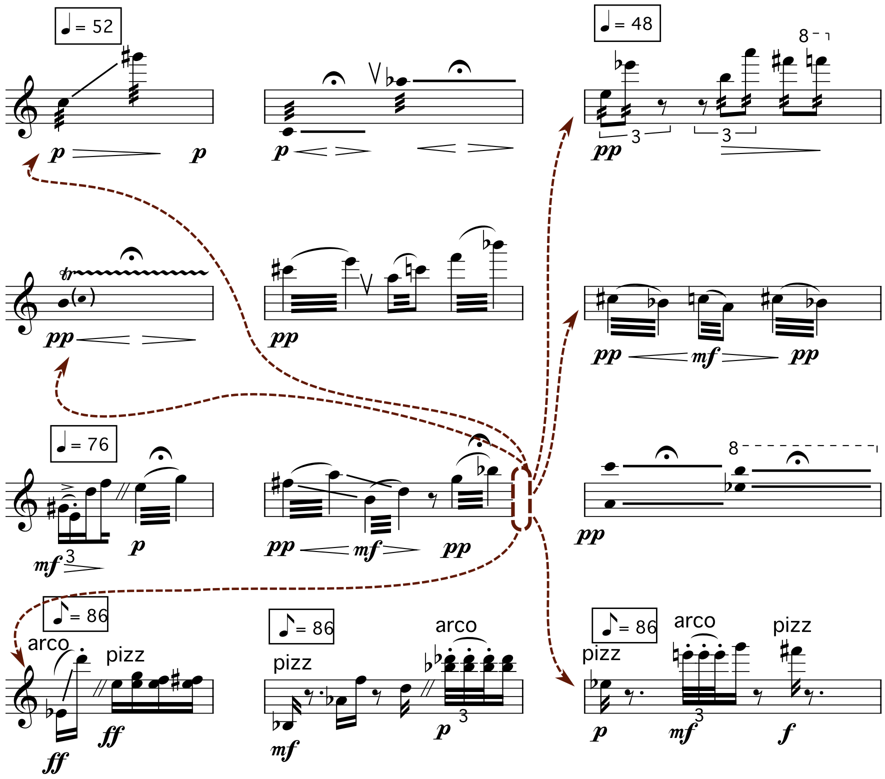
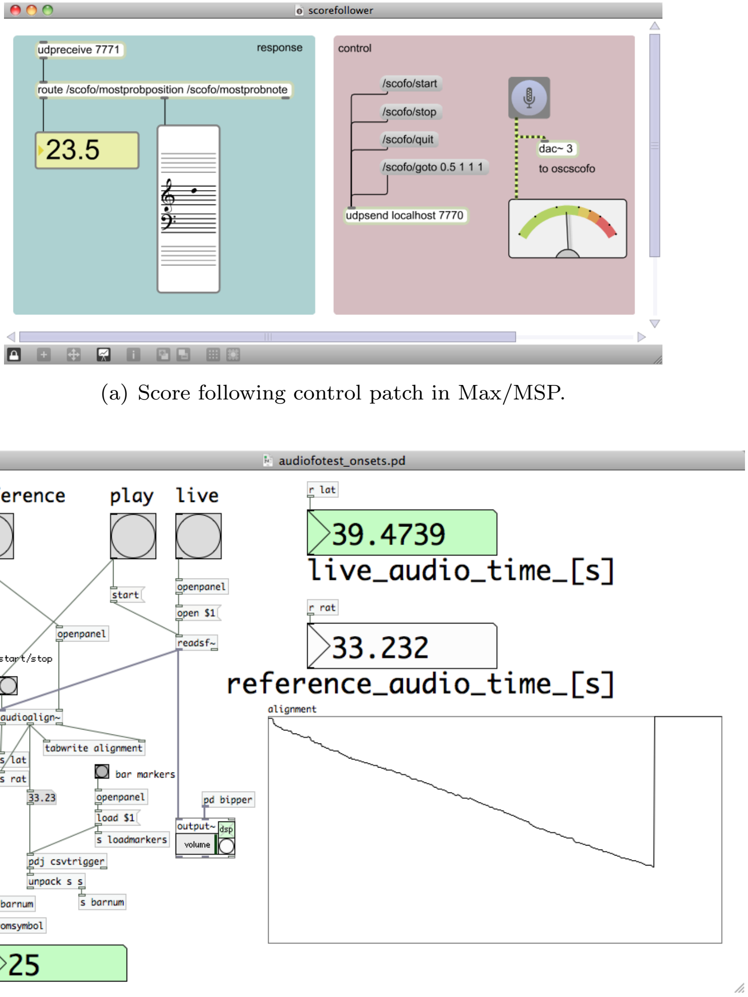


Figure 4.5: Detail of the open-form score for the final movement of P. Manoury's *Tensio*. Arrows denote possible choices for the performer at the end of a particular bar.



(a) Score following control patch in Max/MSP.

(b) Audio following in Pure Data using the [audioalign~] object.

Figure 4.6: Patches in the Max/MSP and Pure Data visual programming environments.

4.2 Offline Alignment for Studio Recording Productions

The common practice in productions of studio recordings consists of several phases. At first the raw audio material is captured and stored on a support; this material is subsequently combined and edited in order to produce a *mix*, which is finalized in the *mastering* phase for commercial release. Nowadays, the whole process revolves around a computer Digital Audio Workstation (DAW). In the case of acoustic recordings, the initial task involves grabbing a complete reference run-through of the entire piece, after which additional takes of specific sessions are recorded so that the mixing engineer can mask performance mistakes and reduce eventual environmental noises. The role of a mixing engineer is to integrate these takes within the global reference in order to achieve a seamless final mix [Bartlett and Bartlett, 2008].

The first step in preparing a mix session consists in *arranging* the takes with regards to the global reference. Figure 4.7 shows a typical DAW session prepared out of a reference run-through (the top track), where additional takes are aligned appropriately. Those takes usually require further cleanup as they commonly include noise or conversation that are not useful for the final mix. This means that, in addition to alignment, the mixing engineer identifies cut-points for each take that correspond to *relevant regions* in the reference. The additional takes are finally blended with the reference by cross-fading short overlapping audio regions to avoid perceptual discontinuities.

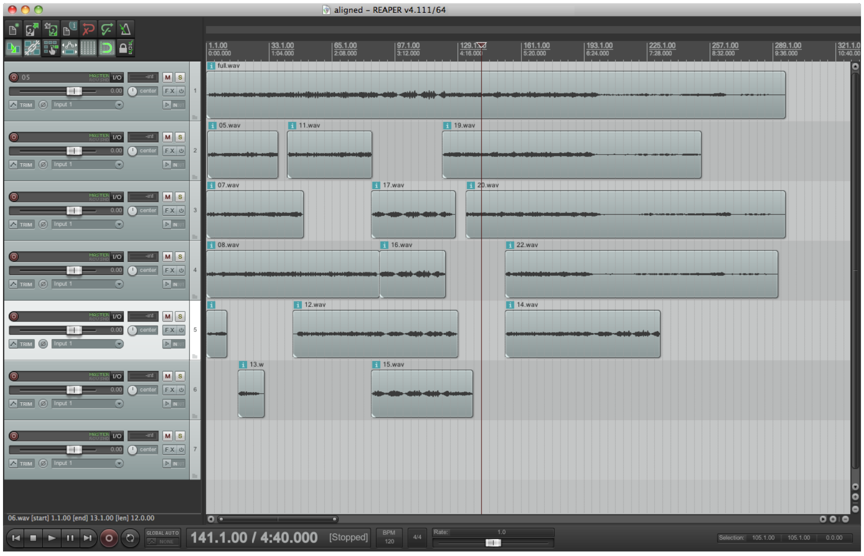


Figure 4.7: A mixing session in a Digital Audio Workstation.

The aim of the proposed methodology, introduced in [Montecchio and Cont, 2011a], is to facilitate the process of mixing by integrating automatic (audio to audio) alignment techniques into the production chain. Special care is taken to consider existing practices within the workflow, such as automatic identification of interest points. In contrast to most literature on audio alignment, the main concern is related to two aspects: the ability to identify a *partial alignment with an unknown starting position* and the detection of *regions of interest* inside the alignment. To do this, the approach of Section 2.4.2 is extended. Even though alternative methods such as HMM or DTW could have been used for this aim, sequential Montecarlo inference presents significant advantages thanks to its straightforward applicability to the present context, its flexibility regarding the degree of accuracy given by the availability of smoothing algorithms and the possibility to trade accuracy for computational efficiency in a direct way.

4.2.1 Related Work

At the application level, alignment techniques were introduced in a similar context in [Dannenberg and Hu, 2003]. A subsequent work [Dannenberg, 2007] describes an "Intelligent Audio Editor" where alignment of audio to a machine-readable, symbolic representation of a piece is integrated into the workflow, improving the automation of several editing processes through operations such as pitch and timing corrections. The application of these approaches is precluded in the present context by the requirement of accessing a symbolic representation of the music. Nonetheless, despite this limitation, these works provide important insights about the integration within a DAW setup.

At the technological level, audio alignment has often been the subject of extensive research. In contrast to traditional methods, an important aspect of the proposed approach is the consideration of *partial results* and the detection of *interest regions*. An audio alignment method with similar aims was introduced in [Müller and Appelt, 2008], that explicitly deals with the synchronization of recordings that have different structural forms. Identification of repetitive structures in audio recordings is the object of [Müller et al., 2011].

4.2.2 Methodology

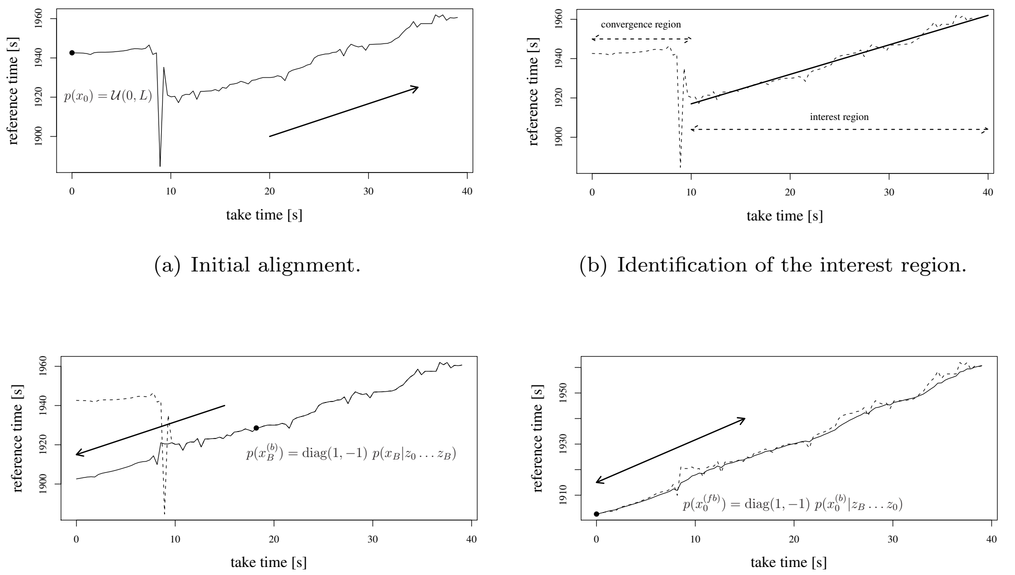
In the first phase a rough alignment is produced as in Figure 4.8(a); the initial uncertainty in the alignment is due to the fact that the initial position is not known a priori. The second phase identifies a sufficiently long region of the alignment that can be reasonably approximated by a straight line, as in Figure 4.8(b); this region intuitively corresponds to the "correct" section of the alignment. These two phases solve the task of placing the takes along the reference as in Figure 4.7. The remaining steps address the tasks in which a more accurate alignment is required, for sensible operations like automatic cross-fading. In the third phase, the initial portion of the alignment is corrected, starting from a position inside the region found in the previous phase and using a reversed variant of the alignment algorithm (Figure 4.8(c)). Finally, a refined alignment is produced by exploiting a smoothing algorithm for sequential Montecarlo inference, as shown in Figure 4.8(d).

Initialization

Initialization plays a central role in the performance of the alignment algorithm; in a probabilistic context this corresponds to an appropriate choice of the prior distribution $p(x_0)$. In a real-time setup, such as the one described in Section 4.1, the player is expected to start the performance at a well known point of the reference; this fact is exploited in the design of the algorithm by setting an appropriately shaped prior distribution, typically one characterized by a low variance centered around the beginning. In the presented situation however the initial point is not known (it represents indeed the subject of interest). To cope with this, a uniform prior distribution $p(x_0)$ is set with respect to the whole duration L of the reference performance; the algorithm is expected to "converge" to the correct position after a few iterations. Figure 4.9 shows the evolution of the probability distribution for the position of the input at different moments of the alignment.

Degeneracy Issues w.r.t. Realtime Alignment

A relevant parameter of Algorithm 1 (page 23) is the resampling threshold. In a degenerate situation most particles have close-to-zero weight, resulting in most of the computation being spent in updating particles which are subject to numerical approximation errors. Resampling is applied to obviate this issue, introducing however other problems (in particular, *sample*



(a) Initial alignment.

(b) Identification of the interest region.

(c) Correction of the beginning of the alignment.

(d) Final alignment, using smoothed inference.

Figure 4.8: Alignment methodology in the case of unknown starting position.

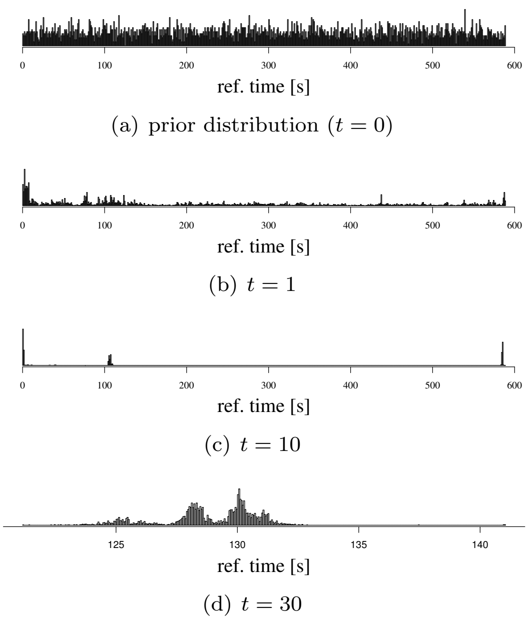
impoverishment, due to a small number of particles being selected multiple times) thus its usage should be limited. In the real-time following case, described in Section 4.1, the mass of the distribution is always concentrated around a small region of the domain thus allowing the resampling threshold to be relatively low. In contrast, in a situation such as the one depicted in Figure 4.9, the sparsity of the distribution in the initial phases of the alignment imposes a much higher resampling threshold, otherwise many relevant hypotheses are soon lost in the resampling phase and cannot be recovered.

Identification of the Interest Region

This phase aims at identifying a region of the alignment which is certainly "correct". As depicted in Figure 4.8(b), a typical alignment can be subdivided into two regions, the first one being characterized by irregular oscillations (not enough data has been observed yet to select the most probable hypothesis with enough confidence) and the second one resembling a straight line; they will be referred to as *convergence region* and *interest region*, respectively.

As can be inferred by observing the plot in Figure 4.8(b), the most important characteristic of the interest region is its slope. From a technical point of view, the slope should be as constant as possible for the alignment region to be significant, while from a musical perspective it should be roughly unitary, implying that the tempos of the take and the reference are approximately the same. In addition to that, the duration of the interest region should be long enough to discard noisy sections of the alignment.

The interest region is identified in the following manner: each of many initial candidate regions $w_1 \dots w_W$ is iteratively expanded as long as it meets the criteria exposed above; the longest of the resulting intervals is elected as the interest region, unless none of them matches the requirements, in which case the alignment is identified as incorrect. The process described above is depicted in Figure 4.10 (dashed horizontal lines represent the regions progressively examined by the algorithm) and formalized in Algorithm 2.



(a) prior distribution ($t = 0$)

(b) $t = 1$

(c) $t = 10$

(d) $t = 30$

Figure 4.9: Evolution of the position distribution, in case of uniform prior.

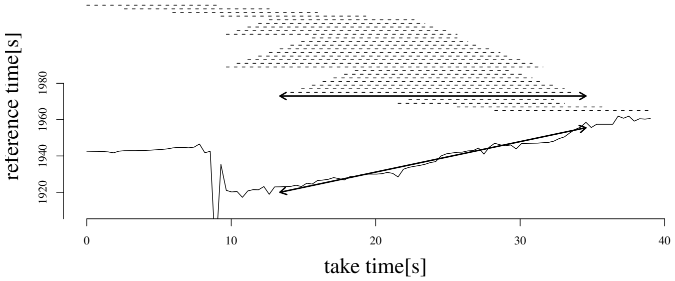


Figure 4.10: Identification of the interest region.

Correction of the Convergence Region

In order to fix the convergence region of the alignment path, the alignment algorithm is run "backwards" on time-reversed audio streams, starting from a point that is known to be correct. The starting point B is chosen inside the region of interest. The prior distribution for the backward alignment is equal to that of the forward alignment at B, however with the value of the velocity for each particle inverted:

$$p(x_B^{(b)}) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} p(x_B | z_0 \dots z_B) \tag{4.1}$$

The audio stream of the take is then reversed and processed by Algorithm 1, as in Figure 4.8(c). Experimentation shows that a narrow uniform or Gaussian prior centered in $(B, -1)^T$ are for practical purposes equivalent to the form of $p(x_B^{(b)})$ mentioned above.

Algorithm 2: Identification of interest region

```
$w_1, \dots, w_W \leftarrow$ regularly spaced intervals in $[0, L]$
candidates $\leftarrow \emptyset$
for $i = 1 \dots W$ do
    while $|w_i| < L$ do
        $w_i \leftarrow \max(0, w_i^{start} - \Delta T), \min(L, w_i^{end} + \Delta T)$
        $a_i \leftarrow$ slope of LS-fit line for points in $w_i$
        $e_i \leftarrow$ mean difference with LS-fit line in $w_i$
        if $a_i \in [1 - \Delta A, 1 + \Delta A] \cap e_i < \Delta E$ then
            candidates $\leftarrow$ candidates $\cup$ $i$
        else
            break
if $|candidates| > 0$ then
    interest region $\leftarrow \max_{i \in \text{candidates}} w_i$
else
    alignment is incorrect
```

Smoothing Inference

Sequential Montecarlo inference algorithms are typically used for online estimation; this implies that at each instant only the information about the past is exploited, instead of the whole observation sequence. In the context of an offline application however these real-time constraints can be dropped. Both the Forward/Backward and Viterbi inference algorithms can be deduced, as reported in Section 2.4.1. Even though the running time of both algorithms is quadratic in the number of particles, the issue can be mitigated by an appropriate choice of the prior distribution $p(x_0^{(fb)})$ as a resampling of diag$(1, -1)$ $p(x_0^{(b)})$, using a smaller number of samples.

4.2.3 Experimental Results

The efficacy of the proposed approach is evaluated regarding the initial phase of laying out the takes as in Figure 4.7. The accuracy of the alignment in terms of latency and average error was not be performed in this case, due to the lack of a (manually annotated) reference linking the timings of each musical event for all takes to the reference recording. However in this situation the aim is rather to position correctly the highest number of takes against the reference, rather than to align them with the highest possible precision. We select the center point of the interest region identified in the second phase as the alignment reference for the whole take.

In all the tests, the number of particles N_s is proportional to the duration of the reference (60 particles per second). A minute of audio can be aligned in 2.29s for $N_s = 100\ 000$ on a laptop computer with a 2.4 Ghz Intel i5 processor (using a single core).

Dataset Description

Recordings produced in two real-life sessions, by different groups of sound engineers, were collected, totaling approximately 3 hours of audio data. The first collection is a recording session of the second movement of J. Brahms' sextet op. 18; the second collection was produced shortly after the premiere of P. Manoury's "Tensio", for string quartet and live electronics, in December 2010. Table 4.3 summarizes their characteristics.

dataset	n. of rec.	duration [s]		
		ref.	takes (avg,std)	total
Brahms	20 + ref.	588.8	112.8, 92.0	2844.0
Manoury	49 + ref.	2339.4	113.5, 94.0	7900.4

Table 4.3: Collections used for evaluating the alignment of takes to a run-through.

Alignment Accuracy - Brahms Collection

For this collection, a manual placement of all the takes with respect to the reference recording was performed using a musical score, in order to evaluate the correctness of the automatic procedure. Aural inspection of the data showed that none of the recordings but one presented undesired noises. All the takes but one were correctly aligned. In the unsuccessful case, the length of the recording itself was one second shorter than the minimum length for an interest region (15s); using last alignment point as a reference, the placement of this take also results to be correct.

Alignment Accuracy - Manoury Collection

The dataset contains a complete run-through and 49 separate takes. The particularity of this dataset is the presence of undesired material for the final mix in many of the individual takes (such as speech, practice sessions, volume and calibration tests). Out of 49 takes, 14 contain exclusively noise and 21 partially. In the former case the alignment is considered correct if the file is discarded, in the latter the aim is to align correctly the interesting portion of the take. This is in sharp contrast with the "cleanness" of the Brahms set and presents difficulties that were not foreseen when formulating the alignment procedure.

Contrarily to the Brahms dataset, the evaluation of the alignment precision was done a posteriori: instead of performing a manual alignment in advance, the results of the automatic alignment were checked. The reason for this lies behind the length (approximately 40 minutes) and complexity of the piece: even with the score at disposal, it was immediately evident that a manual alignment would have taken a very long time. It is precisely this difficulty that sound engineers had to face.

A first experiment aligning this dataset yielded rather poor results on the 21 files containing noise regions of significant length (in some cases up to more than one minute); since in almost all cases the noisy portion was at the beginning, it was decided to directly align the reversed audio streams in the first phase. With this simple adaptation, of the 35 files containing interesting regions, 26 were correctly aligned; all of the 14 takes that contained exclusively noises were correctly discarded by the algorithm.

The absence of false positives (no noise-only takes were mistakenly aligned) and the correct positioning of all the aligned files suggest that the simple algorithm for identification of the interest region is robust enough to be applied to rather short audio segments, yielding the possibility of repeating the alignment algorithm multiple times on different subregions of the audio in order to avoid noisy sections and consequently increasing accuracy.

4.2.4 Integration within the Existing Workflow

The audio industry has established over the years common procedures for mixing that are adopted in most professional studio records worldwide. Integration of new technologies within an existing workflow therefore requires special attention to common practices. To this end, several interviews with sound engineers were conducted.

From an R&D standpoint, the ideal integration would be obtained via a direct implementation of the technology into the graphical user interface of common DAW softwares. Such integration would allow novel possibilities, such as linking two tracks by means of their alignment and defining the placement of transition points between them for crossfading, avoiding any destructive editing regarding the discarded audio regions. Such integration however requires direct collaboration with software houses which are mostly close to public domain development.

An alternative solution is represented by standalone alignment tools, whose outputs should be directly importable into a commercial DAW using an interchange format. Virtually all the major DAWs and video post production systems support the Open Media Framework (OMF) and the Advanced Authoring Format (AAF), respectively owned by Avid Technology, Inc.[5] and by the Advanced Media Workflow Association (AMWA)[6].

4.3 Real-Time Gesture Recognition and Tracking

The study of gestural interaction in the context of music has been the subject of several research works. A suggestive similitude is that of a conductor, who controls an ensemble of musicians through gestures that are not aimed at a direct generation of sound through physical contact. Although this particular case has been investigated in many works – such as [Churnside et al., 2011], which describes an installation that allows users to intuitively control tempo and dynamics of a pre-recorded performance – the research domain is much broader, encompassing more general studies on the relationship between gesture and sound: a recent review of the relevant literature is proposed in [Caramiaux, 2011].

The design of musical instruments that exploit more abstract forms of control than traditional acoustic instruments is a field that emerges consequently from research in gesture interaction; the development of digital instruments which move beyond the traditional piano-keyboard control paradigm is the subject of [Miranda and Wanderley, 2006]. As a confirmation of this trend, it is interesting to note that the latest edition (2011) of the Margaret Guthman Musical Instrument Competition[7] – an annual event to "find the world's best new ideas in musicality, design and engineering" – was won by *MO*[8] (Musical Objects), modular tangible interfaces that enable musical expression with everyday objects exploiting gesture tracking technologies.

In this section the focus is on the evaluation of the accuracy of gesture alignment and recognition using the approach proposed in Section 2.4.3.

4.3.1 Alignment of Synthetic Gestures

A controlled reference is synthesized using an arbitrary gesture that is considered as a movement in the 3-dimensional Cartesian space. The *Viviani curve* (pictured in Figure 4.11) was selected for this task:

$$\begin{array}{rcl} x(t) & = & a(1+\cos(t)) \\ y(t) & = & a\sin(t) \\ z(t) & = & 2a\sin(t/2) \end{array} \tag{4.2}$$

[5] http://www.avid.com

[6] http://www.amwa.tv

[7] http://www.gtcmt.gatech.edu

[8] http://interlude.ircam.fr/wordpress/?p=229

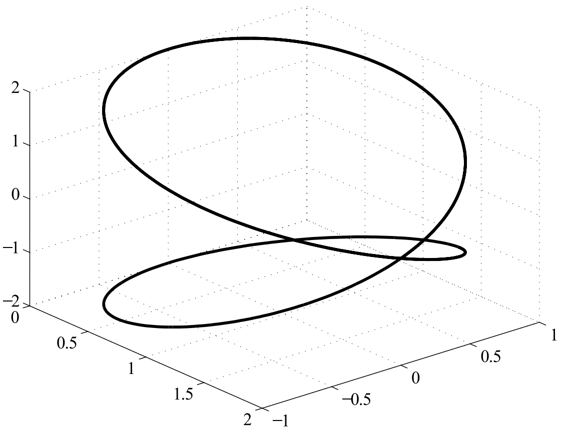


Figure 4.11: Viviani's curve (Equation 4.2), for $a = 1$.

Temporal alignment

Starting from the Viviani curve of Equation 4.2, a transformation is applied in order to create a new shape to be used as input gesture. The curve is sampled according a non-linear function of its index, namely the cubic polynomial x^3.

Experimentation is aimed at evaluating the accuracy of the proposed model, using the HMM system described in Section 2.3.4 as a reference, according to different values of the observation likelihood standard deviation (σ) and the signal-to-noise ratio (SNR). In the particular form of the proposed model evaluated here, the state space is defined as a two-dimensional vector that consists of just position and speed; this choice is motivated by the aim of replicating as closely as possible the modeling power of the HMM system. The experiment was performed using $N_s = 200$ particles.

The time series of the estimated position corresponding to the alignment should ideally match the cubic function used for resampling. The accuracy of the estimation, depicted in Figure 4.12, is evaluated computing the Euclidean distance between the estimated position time series and the ground truth.

Figure 4.12 also shows that for both models the signal-to-noise ratio value defines a threshold for the standard deviation of the likelihood pdf, under which the method is unable to accurately estimate the position (a smaller standard deviation means a narrower Gaussian distribution, resulting in a less "tolerant" system). The figure clearly reveals that the Particle Filtering-based model is more tolerant to noise. The minimum error values achieved by both models are equivalent (occurring in the best possible experimental conditions, i.e., high SNR and narrow observation pdf); the proposed model however achieves consistently better results in all the other cases.

Rotation matrix adaptation

This section analyzes how a three-dimensional dynamic rotation matrix can be estimated by the proposed model. Three time series are estimated, each defining the evolution of a rotation angle: the gesture is in fact executed while changing orientation over time. The alignment algorithm is evaluated at different noise levels. The three time series are arbitrarily defined as

$$\begin{aligned}\phi(t) &= t^2\\ \theta(t) &= t\\ \psi(t) &= \sqrt[3]{t}\end{aligned} \tag{4.3}$$

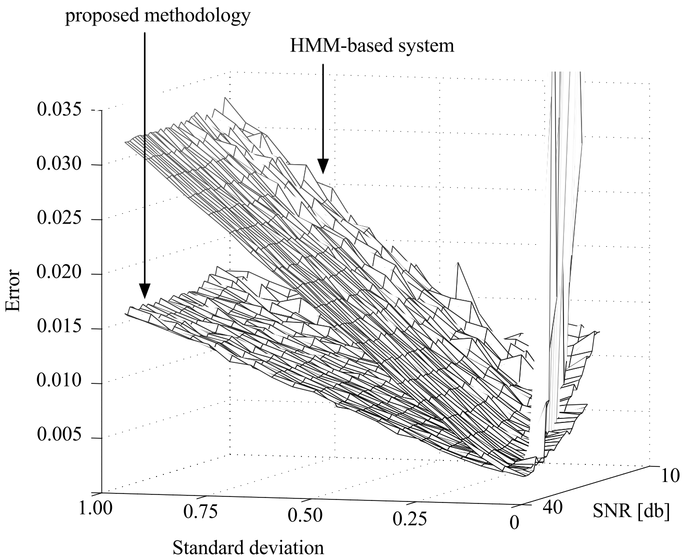


Figure 4.12: Alignment error on synthetic gesture data.

From the three time series the dynamic rotation matrix is computed at each time step and the template curve of Equation 4.2 is rotated accordingly; a white Gaussian noise is added to the resulting curve (Figure 4.15(a)).

For each test, the model returns the estimated angles ϕ, θ, ψ that are compared to the ground truth. The errors obtained for each value of σ and SNR are illustrated by Figure 4.13. As expected, the global minimum is attained at the lowest SNR value.

Figure 4.14 depicts the dynamic estimation of the three rotation angles, comparing their evolution to the ground truth. The estimation is noisy since the Gaussian variances $\sigma_\phi, \sigma_\theta, \sigma_\psi$ are high, compared to the SNR level. From these estimated angles together with the estimated temporal alignment, the curve can be resynthesized according to the estimated features, as shown in Figure 4.15(b).

In the example, initial rotation was null: $\phi(0) = \rho(0) = \psi(0) = 0$; the prior distribution of the rotation random variable in the model was set accordingly. In the case of non-zero initial values, the prior distribution should be more tolerant and the model should require a certain latency before converging.

4.3.2 Identification of Real Gestures

In this section the accuracy of the proposed model is evaluated using real-world datasets; results are compared with other models proposed in the literature.

Two-Dimensional Gestures

In this experiment, the case of two-dimensional pen gestures is considered making use of the dataset presented in [Wobbrock et al., 2007]; assessments are performed replicating the experimental conditions proposed therein.

Dataset The dataset was introduced in [Wobbrock et al., 2007]. Ten participants have been recruited. For each one of the 16 gestures in the vocabulary (figure 4.16), "subjects entered one practice gesture before beginning three sets of 10 entries at slow, medium, and fast speeds" [Wobbrock et al., 2007]. Hence, the whole dataset contains 4800 gesture examples.

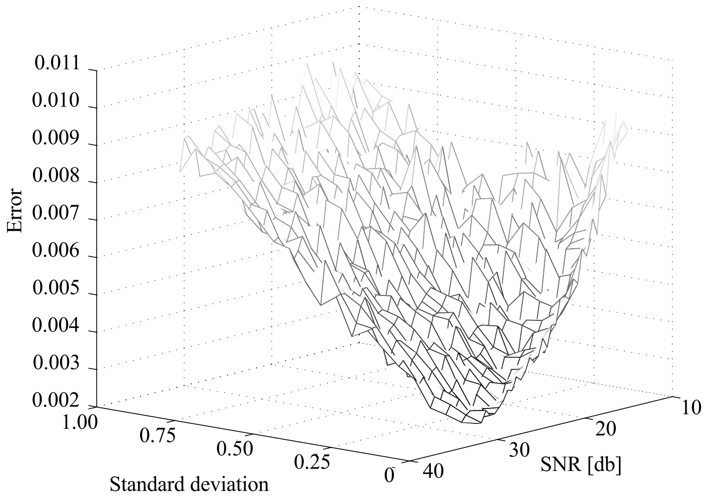


Figure 4.13: Alignment error for a synthetic gesture with dynamic rotation and Gaussian noise.

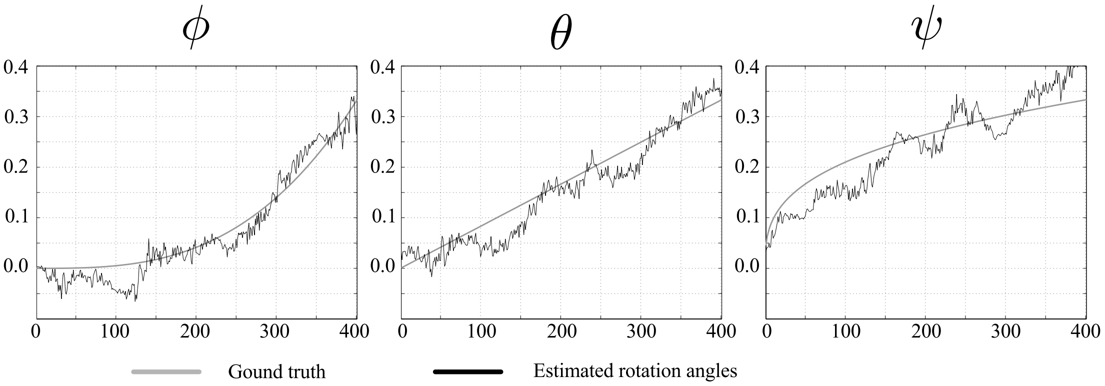
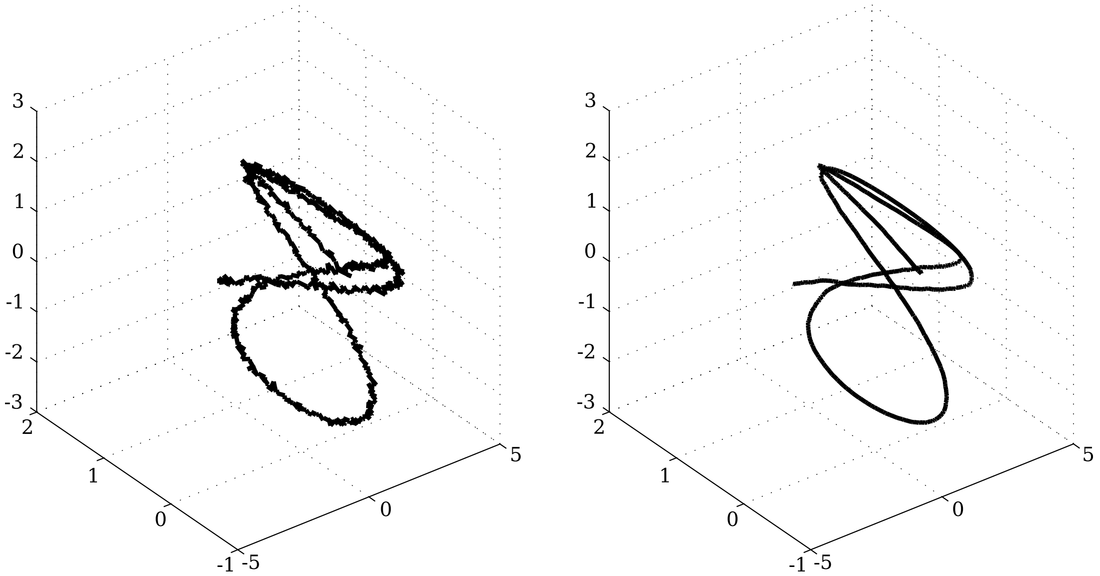


Figure 4.14: Dynamic estimation of gesture rotation.

Model Configuration In [Wobbrock et al., 2007], the authors proposed a pre-processing step that roughly consists in rotating, scaling and translating gestures appropriately before computing Euclidean distance. Both the rotation angle and the scaling coefficient are invariants in the recognition process. The proposed model allows for taking into account these invariants by defining them as state variables; therefore, the gesture features estimated are the following: position, velocity, rotation angle and scaling coefficient.

Experiment Setup The test procedure is as follows: of a given participant's gestures at a given speed, one example per gesture type is considered. The test procedure selects randomly one template example from the 10 available trials per gesture, and a random test example from the remaining trials. This process is repeated 100 times.

Results The recognition rates together with their standard deviations are reported in Table 4.4. Results show that the proposed system and a DTW-based method based on [Liu et al., 2009] achieve the best recognition rates. The proposed Particle Filtering methodology does however perform *causal* inference, thus can be exploited for joint realtime gesture recognition and tracking.



(a) Application of dynamic rotation and noise.

(b) Resynthesized curve according to the features estimated by the alignment.

Figure 4.15: Transformed and reconstructed Viviani's curves.

In order to exhibit one particular advantage of the proposed model over other systems (the invariance to selected features) an additional experiment was carried out to show how the methodology responds to rotation in the 2D space. Test gestures were repeatedly rotated randomly according to different angle ranges, and matched against non-rotated templates. To allow for a fair evaluation, a subset of the gestures pictured in Figure 4.16 was selected, as some shapes occur more than once in rotated variants (in particular, gestures 11, 12, 15, 16 were removed as they resemble respectively gestures 6, 7, 2, 5). The parametrization used to obtain the results presented before was chosen, and the prior distribution of the rotation random variable was set to be inclusive of the possible gesture rotations. An average accuracy of 97.4 % was attained over 21 trials in which the allowed rotation angle (and prior distribution accordingly) was set at regular intervals in $[0, 2\pi)$. The accuracy values in the various intervals present no significant differences, highlighting the independence of the proposed system to input rotation.

	\$1	DTW	HMM	proposed
mean	97.27	97.86	95.78	97.81
std. dev.	2.38	1.76	2.06	2.18

Table 4.4: Recognition rates (in percentage) for two-dimensional gestures. The proposed model is compared with: $1 [Wobbrock et al., 2007], DTW [Liu et al., 2009], HMM (Section 2.3.4).

Three-Dimensional Gestures

In this section, recognition accuracy is evaluated on three-dimensional gestures captured by accelerometers devices. The considered gestures are reported in Figure 4.17.

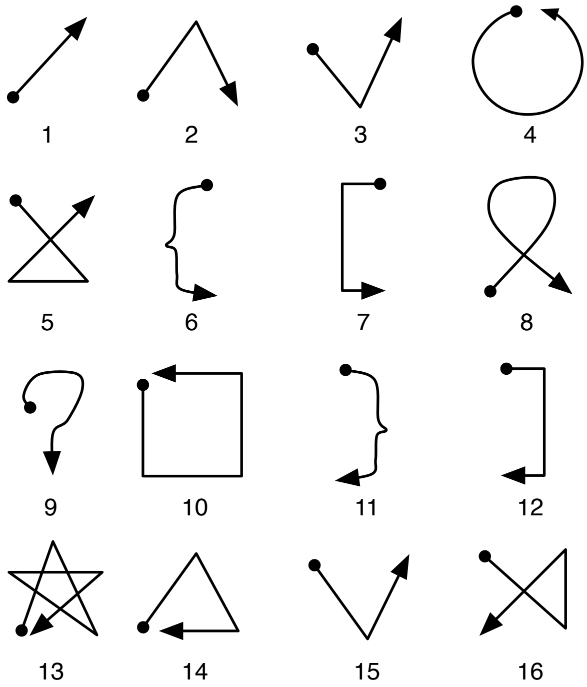


Figure 4.16: Two-dimensional gesture vocabulary used for identification.

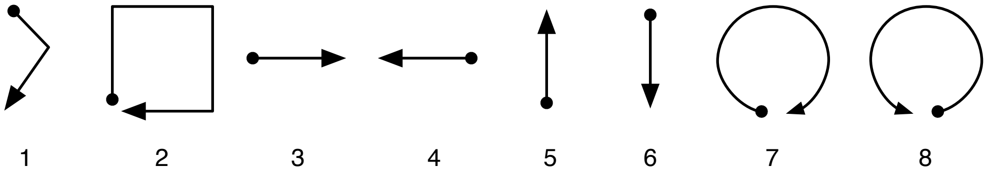


Figure 4.17: Three-dimensional gesture vocabulary used for identification.

Dataset The dataset was collected from four participants as follows: among ten trials, users had to perform the corresponding gesture handling the interface according to five orientations, corresponding to roughly $0, \pi/6, \pi/4, \pi/3, \pi/2$, as pictured in Figure 4.18.

Each gesture/orientation pair was performed twice, hence totaling ten trials per gesture. An example of gesture data from the dataset is given by Figure 4.19, where gesture 3 was performed handling the interface with orientations of 0 and $\pi/2$. As discussed in Section 2.4.3, the curves are similar except for the presence of an offset component.

Experiment Setup Let the samples for each gesture type be labeled by the order in which they were collected. At the i-th test, the i-th trial from each gesture is used as reference, while the remaining examples belong to the testing set. The evaluation test leads to confusion matrices from which the mean value of its diagonal coefficients is computed.

Results The recognition accuracy of the proposed methodology is compared to several existing methods. The test procedure is based on the procedure from [Liu et al., 2009].

The results, reported in Table 4.5, clearly show that the proposed model is more accurate since it can adapt the angles and offsets over time, compared to the others. The confusion matrix shows that errors occur for gestures $3, 4, 5$ and 6; it should be observed however that

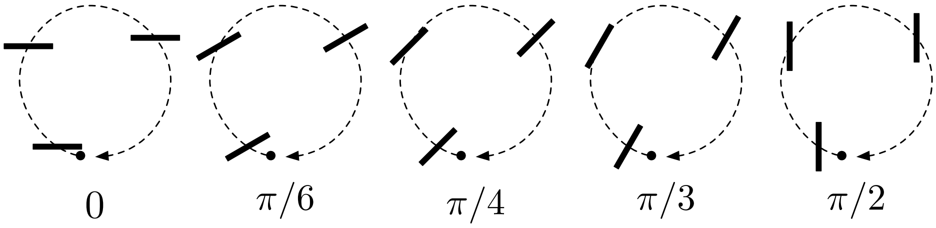


Figure 4.18: Different orientations of the capture interface.

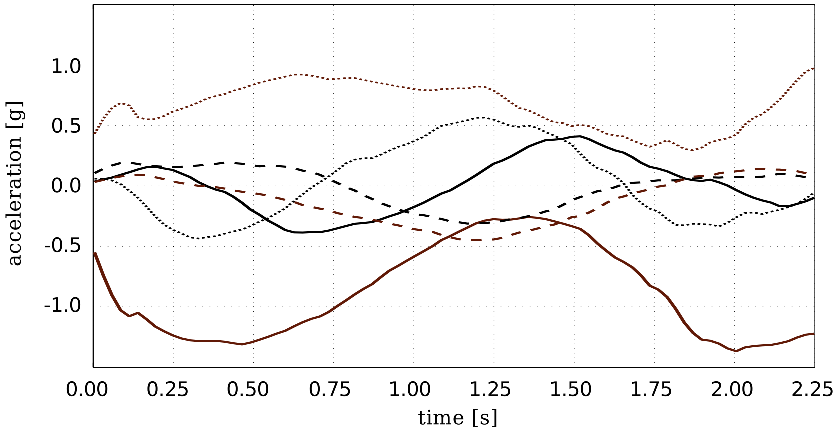


Figure 4.19: A three-dimensional gesture captured with different device orientations.

performing one of these gestures with the interface rotated by $\pi/2$ is equivalent to perform another gesture with the correct handling of the interface.

4.3.3 Real Time Segmentation of Composite Gestures

An interesting problem is related to the segmentation of streams of multiple gestures, especially in a real-time context. In [Black and Jepson, 1998a], the authors used the condensation algorithm with resampling at each time step. This resampling process corresponds to a local reinitialization of the set of particles. At the same time, they select between 5 and 10% of the set of particles to be reinitialized globally: particles are spread over all the possible gestures according to the given initial conditions. This subset of particles is choosen randomly among the whole set.

An alternative solution makes use of the strategy for handling repetitions presented in Section 4.1.2, reinitializing the gesture index of the particles whose position is sampled with value greater than 1. An example, performed using three basic gestures *square, circle*, and *triangle*, is pictured in Figure 4.20, reporting on the top the original test signal, along with the correct segmentation.

	uWave	HMM	Condensation	proposed
Gesture 1	54.4	47.2	68.9	84.2
Gesture 2	51.9	37.8	53.3	90.3
Gesture 3	46.9	53.3	40.6	66.7
Gesture 4	46.9	51.7	41.7	60.3
Gesture 5	78.6	50.6	34.7	53.3
Gesture 6	53.9	33.3	48.6	51.7
Gesture 7	71.1	82.5	67.5	85.6
Gesture 8	87.8	75.6	80.3	83.9
mean	61.5	54.0	54.4	72.0
std. dev.	15.6	17.0	16.2	15.8

Table 4.5: Recognition rates (in percentage) for three-dimensional gestures. The proposed model is compared with: uWave [Liu et al., 2009], HMM (Section 2.3.4), Condensation [Black and Jepson, 1998a].

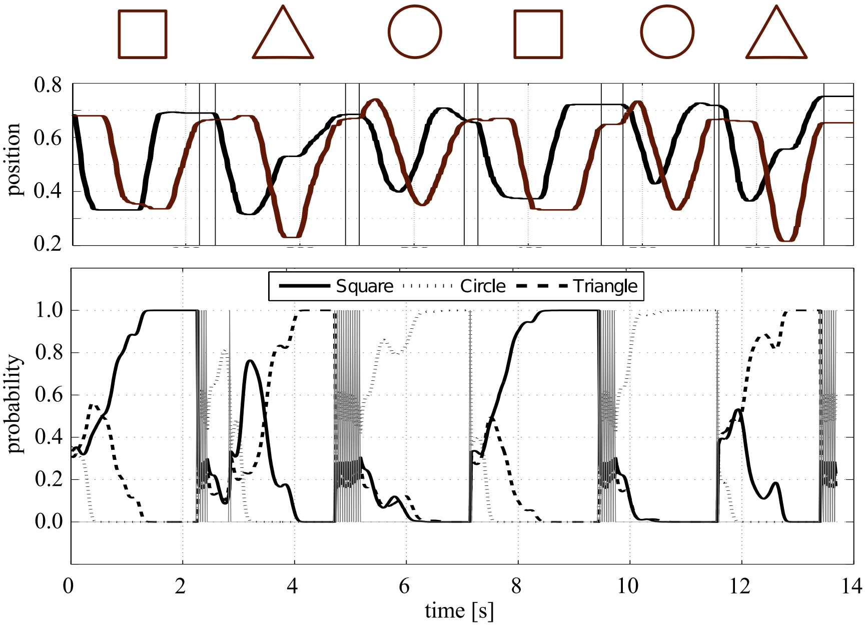


Figure 4.20: Segmentation of gesture sequences.

4.4 Improved Access to Music Heritage Collections

During the last decades a large number of sound archives and Cultural Heritage preservation institutions started to transfer their collections of analogue recordings into digital form. To this end, a remarkable effort has been made at the international level (and in particular by the European Community through the funding of preservation initiatives) to digitize significant amounts of music and speech recorded in discs, tapes and other carriers. The digitization of the audio content however represents only a single, although crucial, aspect in the transformation process of a physical sound archive into a *music digital library.* In particular, having lost their connection with the physical support, digital objects require a set of suitable cataloguing values that allow users to retrieve and access them. Consider the common case of music archives containing vinyl discs in the form of 33 rpm LPs: a complete digitization process would require annotating all the individual tracks for each side of every disc; moreover, metadata for each track should then be made available, ideally in structured form, about title, author, performers and other relevant informations.

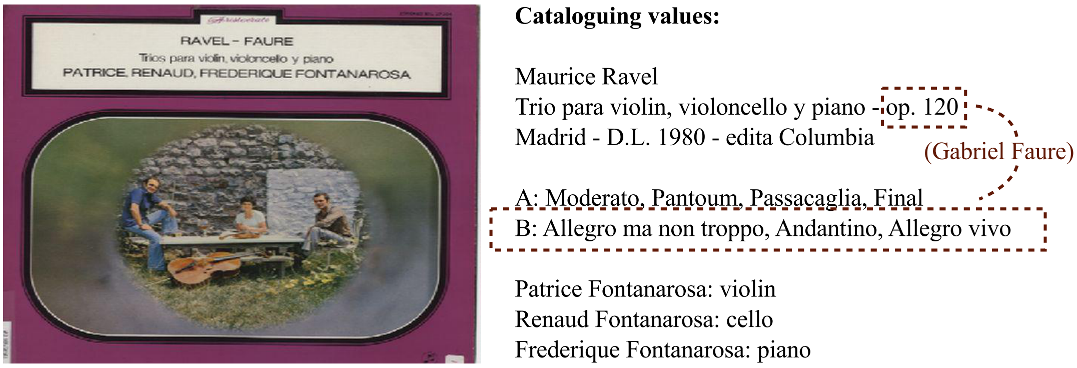
A digitization process requires a number of competences to be exploited at the same time: a technical expertise for taking care of the digital transfer, a cataloguing expertise to extract all the relevant information from the the record covers and possibly accompanying booklets, and a musicological expertise to segment the digitized audio and match individual tracks with the corresponding cataloguing values. The extraction of correct cataloguing values from the cover of commercial records is by itself a complex task, because the intended audience usually is not interested, nor has specific competences, in music metadata.

Several digitization campaigns have been carried out paying much more attention to the digital acquisition process than to the creation of consistent metadata and to the development of suitable tools for enabling a direct access to content. This is particularly true for small sound archives, which do not have the resources to take care of the complete process, but often applies to larger archives as well. For instance, both the Fonoteca of the University of Alicante (Spain) and the Discoteca di Stato of Rome (Italy) − institutions which were in contact for the development of the methodology presented here − decided to not segment LP sides into individual tracks because of the lack of funding for this particular task. Moreover, it is difficult to verify the correctness of metadata, because this requires listening to the complete collections that contain thousands of hours of recorded material.

An example of problems that arise because of imprecise or missing metadata is given in Figure 4.21, where the cover and the transcribed metadata of an LP are presented as they are contained in the Fonoteca of Alicante. It can be seen that, although the composer Gabriel Faure was mentioned in the cover, only Maurice Ravel is reported regarding the authorship of the works. Only a pre-existing familiarity with Ravel's Trio may help users realize that side B contains the work of another composer. As it often happens in publication of classical music albums, the main interest is on the performer and on the main work, while other less popular works are used to fill the remaining space on the disc.

The methodology reported below addresses issues related to music access and retrieval, with particular attention to music digital libraries created from analogue sound archives. The goal is to identify automatically the music content of the digitized material and, at the same time, to segment the audio files into single tracks. Identification is carried out through the automatic matching of the digitized material with a set of digital recordings for which reliable metadata are available. The task is particularly relevant in the classical music domain, because popular works are often performed and recorded by many different artists.

In this section the focus is directed towards the identification of digital acquisitions of vinyl discs. The particular choice is motivated by the availability of a collection of digitized LPs for the MusiClef 2011 evaluation campaign [Orio et al., 2011]. The collection is owned by


RAVEL - FAURE
Trios para violin, violoncello y piano
PATRICE, RENAUD, FREDERIQUE FONTANAROSA
(Gabriel Faure)

Cataloguing values:

Maurice Ravel
Trio para violin, violoncello y piano - op. 120
Madrid - D.L. 1980 - edita Columbia

A: Moderato, Pantoum, Passacaglia, Final
B: Allegro ma non troppo, Andantino, Allegro vivo

Patrice Fontanarosa: violin
Renaud Fontanarosa: cello
Frederique Fontanarosa: piano

Figure 4.21: Inconsistent metadata associated to an LP.

the Fonoteca of the University of Alicante, which kindly shared the metadata and allowed the workshop organizers to compute the music features used for the identification. This collection represents a case study that can be extended to a large number of LP collections owned by private and public institutions. It is worth mentioning that the digitization of LPs is a significant contribution to the preservation of cultural heritage, because only a restricted number of LPs has been reissued on CD. Recordings of less known performers – or of less known composers – risk being lost. The methodology can be easily extended to the segmentation and identification of music extracted from any analogue carrier, in particular from audio tapes. In the latter case and especially in the case of recordings of unpublished material – live recordings of rehearsals and performances in theaters and concert halls – the audio may contain invaluable information that witnesses the musical heritage of the last century.

4.4.1 Overview of the Identification Strategy

The problem of music identification is addressed using the approach described in Section 3.2. As discussed therein, the simplification introduced by the representation via a bag of words approach can negatively affect the identification accuracy. Therefore, a refined methodology is proposed, which makes use of a two-phase identification process. The first phase relies on the above bag of feature paradigm and aims at obtaining a first list of results where candidate documents are ranked at the top k positions. Ranking is subsequently refined applying the alignment methodology described in Section 2.4.2 to the top ranked results against the query; this makes it possible to consider a query containing multiple results (as in the context of Section 3.3.4) and exploit the adaptation of the alignment algorithm to the case of an unknown starting point described in Section 4.2.2. This hybrid approach, depicted in Figure 4.22, allows to speed up the identification process with respect to the sole application of audio alignment, which is more computational expensive.

Alignment Classification

It is clear at a visual inspection whether or not an alignment corresponds to a positive identification: in the first case its plot resembles a straight line – which, supposing that the tempo of the performances is similar, should be characterized by a roughly unitary slope – such as that of Figure 4.23(a), while in the second case it displays an irregular behavior as pictured in Figure 4.23(b).

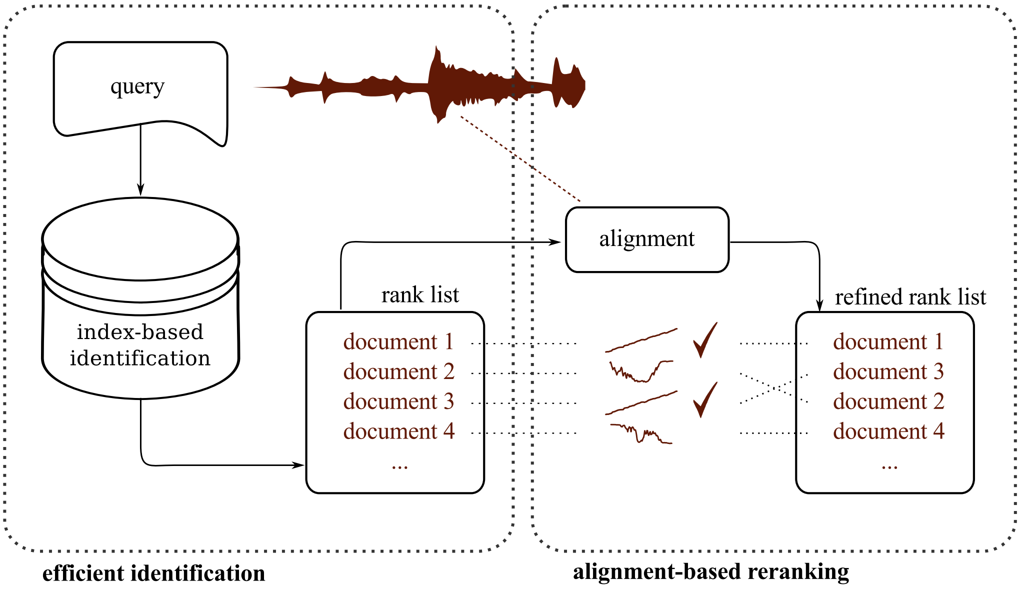


Figure 4.22: Overview of an efficient two-step identification methodology.

Logistic regression is used to classify alignments using the histogram of the slopes between consecutive alignment points. As Figure 4.23(c) shows, the concentration of slope values around 1 is particularly revealing of a correct alignment, whereas incorrect alignments are characterized by irregular distributions. In Section 4.2.2 an alternative algorithm is proposed for detecting the relevant portion of a correct alignment by performing regression on the data points. A classifier was chosen in this context for the advantage of learning a classification strategy from examples that can be quickly hand-labeled.

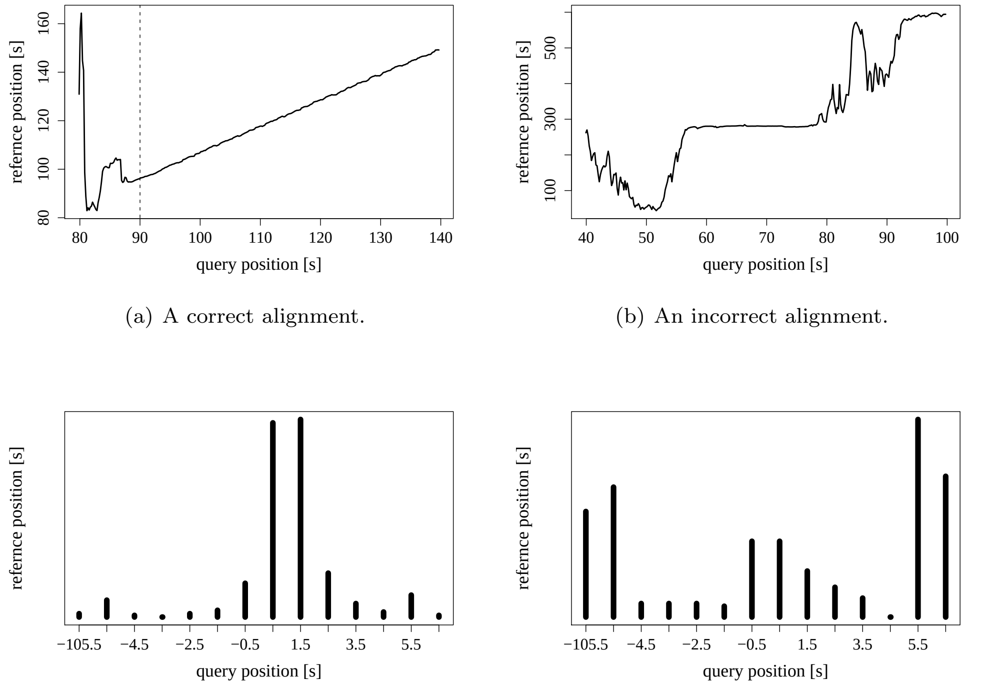
4.4.2 Test Collections

MusiCLEF collection The first collection has been recently introduced and adopted in the Music Identification Task of the MusiCLEF Lab in CLEF2011 [Orio et al., 2011]. It is made up of 6680 recordings; among those, 2671 music works are represented at least twice in the collection, forming 945 cover sets (Figure 4.24).

Mazurka collection An additional validation was performed using the test collection adopted in the Audio Cover Song Identification track in MIREX[9], in order to compare the proposed approach to other methodologies in the literature. The collection developed by the Mazurka project[10], consisting of historical and recent recordings of Chopin's mazurkas, was sampled obtaining 11 versions of 49 mazurkas, for a total of 539 tracks. In order to replicate the MIREX evaluation procedure, 10 diverse sets of 49 mazurkas were repeatedly selected at random, and the results were averaged.

Fonoteca collection A collection of 100 LPs from the Alicante Fonoteca is used to evaluate the procedure detailed in Section 3.3.4, matching the contents of the LPs with recordings in the MusiCLEF2011 collection.

[9] MIREX is an annual evaluation campaign for music IR systems. `http://www.music-ir.org/mirex`
[10] `http://www.mazurka.org.uk`



(a) A correct alignment.

(b) An incorrect alignment.

(c) Slope distribution of a correct alignment.

(d) Slope distribution of an incorrect alignment.

Figure 4.23: Alignment classification strategy.

4.4.3 Evaluation Measures

In the domain of Information Retrieval, evaluation measures play a central role. Let the response to a query consist of an ordered list D of retrieved documents, where R denotes the subset of *relevant* documents.

Precision is the fraction of retrieved documents that are relevant to the search. *Recall* is the fraction of relevant documents that are succesfully retrieved.

$$\text{Precision} = \frac{|R \cap D|}{|D|} \tag{4.4}$$

$$\text{Recall} = \frac{|R \cap D|}{|R|} \tag{4.5}$$

Precision (recall) *at n*, denoted as $P@n$ ($R@n$) measures precision (recall) considering only the top n retrieved documents.

Mean Average Precision The *Average Precision* for a query is computed as the average of the precision values (the ratio of relevant documents over retrieved document) at each of the relevant documents in the ranked sequence. Let $R_i(j) = 1$ if the document at rank j is relevant for the i-th query, 0 otherwise; then for n queries the Mean Average Precision (MAP) value is computed as the mean of the average precision for each query:

$$\text{MAP} = \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_j P_i@j \; R_i(j)}{\sum_j R_i(j)} \tag{4.6}$$

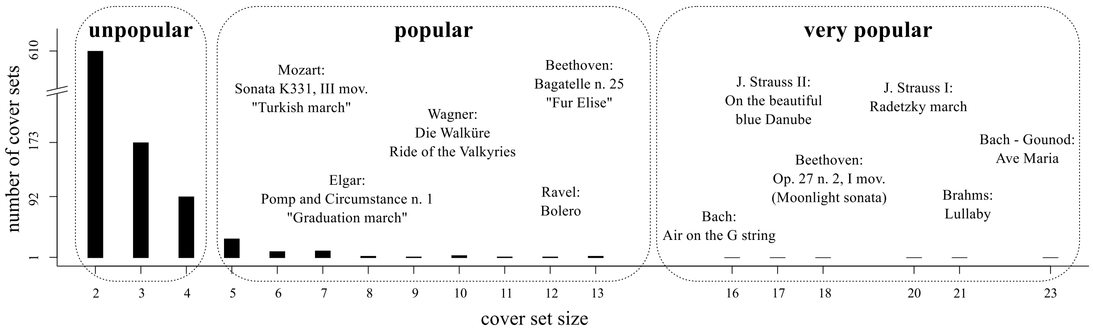


Figure 4.24: Distribution of cover set cardinalities for the MusiCLEF collection.

4.4.4 Experimental Methodology

The main points addressed by the experiments reported below are concerned with the trade off between effectiveness and efficiency. Experiments are focused on two identification tasks: in the first task a music performance is given as query and the objective is to identify existing versions of the query in the document collection; the second task is concerned with the case of queries containing multiple works that was discussed in Section 3.3.4.

Effect of Different Chroma Descriptor Extraction Algorithms

Equation 2.9 describes the basic rationale underlying the computation of chroma vectors. Such descriptors can however be calculated in different ways; the choice of a particular algorithm can affect greatly the identification effectiveness, and should be motivated by experimental evidence. Three algorithms are the object of investigation. The first one, hereafter named *MOD*, is based on a peak picking algorithm that considers only local maxima in the discrete Fourier transform of the acoustic signal. The other two approaches are the ones proposed in [Ellis and Poliner, 2007] and [Lartillot, 2011], to which we will refer as *LabRosa* and *MirToolbox*, respectively.

Effect of the Tuning Adjustment Algorithm

When dealing with music identification, a significant issue is that a work can be performed with an instrument tuned according to a non-standard reference frequency. The tuning adjustment algorithm proposed in Section 3.3.2 aims at addressing this issue. The effectiveness of the tuning algorithm on the chroma vector extraction procedure is therefore evaluated comparing the resulting accuracy. Figure 4.25 depicts the distribution of the tuning frequencies for the recordings in the MusiCLEF collection, estimated using the proposed algorithm; as expected, the distribution is centered on a reference pitch slightly higher than 440 Hz.

Effect of Quantization and Segmentation

The methodology described in Section 3.3.3 is concerned with the transformation of a chroma-based representation of a music work into a bag-of-hashes representation, where each work is segmented into possible overlapping sets of hashes. Quantization and segmentation involve a number of parameters that can affect the identification effectiveness. The first parameter is the quantization level: increasing the number of quantization levels, the size of the vocabulary of descriptors increases, with dramatic consequences on the accuracy of the system.

The result of the hashing procedure is a sequence of hashes that undergoes a segmentation process in order to capture possible structure in the music work or to identify possible works in a long performance. Segmentation involves two parameters: the segmentation length and the

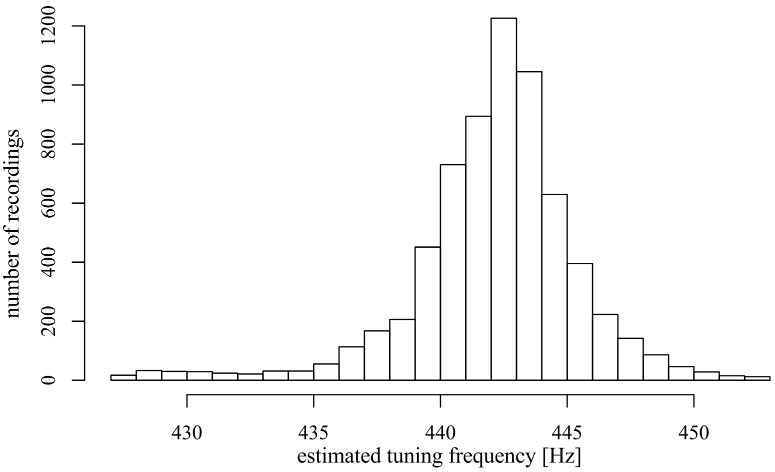


Figure 4.25: Estimated distribution of the tuning frequencies for the MusiCLEF collection.

overlap. The value of these settings may affect both identification effectiveness and efficiency: highly overlapped segments are expected to have a high capability to capture structure in a song, but increasing the number of documents in the index slows down the identification process. It is therefore important to choose an appropriate tradeoff.

In order to address these questions, variations in terms of MAP are measured with respect to modification of the parametrization. Since the selection of the diverse parameter values can affect the effectiveness of the chroma vectors extraction procedures as well as the effectiveness of the tuning adjustment algorithm, we different combinations chroma extraction, tuning, quantization and segmentation are exhamined

Effect of Reranking using Audio Alignment

The adoption of an efficient strategy at the expense of accuracy is motivated by the ability of returning quickly a promising list of relevant candidates among the documents in a large collection, with the aim of refining the initial list using a more sophisticated, but also computationally expensive, technique. It is clear then that *recall* is more important than precision in the first stage. The margin of improvement in accuracy that is obtainable using audio alignment remains to be examined.

In order to address this point, the top k documents in the list are provided as input to the proposedaudio alignment approach and the initial rank list is modified according to the documents that are associated to correct alignments. This hybrid approach depends on the value of the parameter k: higher values improve the quality of identification but also increases the processing time proportionally.

Identification of Multiple Works in a Single Query

The last point concerns the capability of the proposed methodology to identify multiple works within a music recording. In particular, it should be verified whether a fixed-length segmentation strategy is able to support identification of recordings characterized by radically different durations. This issue is address through the approach described in Section 3.3.4.

4.4.5 Results

Experiments were performed on a machine featuring a dual-core 3.4 GHz CPU and a 7200 RPM hard disk. Accuracy results are reported in terms of Mean Average Precision values.

Table 4.6 displays the effect of Chroma extraction algorithm choice, tuning estimation, segmentation of the audio content and quantization level on the Mean Average Precision and mean query time, using the MusiCLEF2011 collection (results are averaged on 651 queries). Chroma features are extracted with an analysis length of 200ms and hopsize of 100ms. Considering the accuracy/efficiency tradeoff, the highlighted configuration (MirToolbox features with tuning estimation, segments of 50s without overlap and 3-level chroma quantization) was selected for performing the subsequent tests. As is clear from the table, the tuning algorithm is beneficial to accuracy in all the configurations.

Table 4.7 shows the results of the selected configuration on the Mazurka datasets, comparing the approach to previous MIREX evaluation campaigns. On this dataset (539 files) the proposed system is able to process about 5 queries per seconds. Due to the choice of rather long segments (50s), a particularly short mazurka is never retrieved; results are shown twice, depending on its inclusion.

While the system is comparable to state-of-the-art approaches in terms of accuracy, it is definitely advantageous in terms of efficiency: for the model described in [Serrà et al., 2009] (the one characterized by the highest accuracy) the author reports [Serrà, 2011] an average time spent in the dissimilarity assessment of two recordings around 0.34s, on a machine with an Intel 2.4 GHz CPU. The proposed methodology, on a comparable Intel 3.4 GHz machine (using a single core), is able to compare about 1000 recordings in the same time. Even though the implementation of the other approach was not explicitly tuned for speed, the efficiency gap is nonetheless evident.

Table 4.8 reports the effect of different choices for the segmentation and length of the analysis window on the accuracy and efficiency of the algorithm for identification of multiple works within a query. 100 LPs from the Alicante fonoteca were matched against 6680 recordings.

Finally, Figure 4.26 presents improvement of reranking using alignment. A minute of audio can be processed in 2.8s for $N_s = 10^5$ particles on the reference machine (using a single core); experiments were performed with $N_s = 100$ particles for every minute of the reference recording. Even though the improvements in terms of accuracy are limited, the alignment allows to declare with confidence that aligned recordings are effectively relevant; it also permits a precise time localization and segmentation of the LP contents.

Parameters			LabRosa		MOD		MirToolbox		
L [s]	O [s]	Q	normal	tuned	normal	tuned	normal	**tuned**	Query Time [s]
20	0	2	55.6	64.0	69.3	72.1	74.2	77.6	8.0
		3	53.9	61.3	59.9	61.5	52.9	53.9	7.0
		4	19.9	20.1	6.4	7.0	3.5	4.1	5.4
	10	2	58.4	68.5	72.5	74.9	75.9	79.1	30.8
		3	55.8	64.4	64.6	64.2	57.8	59.0	25.2
		4	27.4	24.7	9.0	10.1	5.0	6.0	16.6
30	0	2	55.3	64.2	69.3	71.6	73.1	76.9	4.0
		3	54.3	63.1	67.9	71.1	66.1	68.9	3.8
		4	33.0	34.6	17.0	19.1	12.2	13.4	3.8
	15	2	58.0	67.5	71.3	74.4	75.3	78.2	14.4
		3	56.6	65.6	70.7	73.2	69.5	71.7	12.6
		4	38.0	42.0	24.8	26.6	16.3	17.2	10.0
40	0	2	54.6	64.0	67.7	70.8	72.3	76.0	2.6
		3	55.4	63.1	70.2	73.2	69.1	72.5	2.6
		4	39.5	42.2	32.8	34.3	22.7	25.2	2.8
	20	2	57.4	66.2	69.1	72.7	73.8	77.3	8.6
		3	57.6	65.6	72.3	74.8	72.7	75.1	8.0
		4	44.1	47.6	38.3	39.9	27.9	29.6	6.8
50	**0**	2	53.7	62.6	65.7	69.2	71.7	75.9	2.0
		3	55.2	64.0	70.6	74.4	73.7	**76.6**	**2.0**
		4	42.3	47.7	45.2	46.4	35.6	38.8	2.2
	25	2	55.9	64.2	68.3	71.1	73.4	76.9	6.2
		3	56.0	66.0	71.9	74.6	74.5	77.6	5.4
		4	44.8	51.3	48.3	50.4	38.5	39.6	4.8
60	0	2	52.3	60.3	65.5	68.3	70.1	75.0	1.4
		3	53.5	63.3	69.6	73.3	73.5	77.0	1.6
		4	44.0	50.4	52.3	57.0	46.1	51.0	2.0
	30	2	54.6	62.8	66.7	70.0	71.7	76.1	4.2
		3	55.7	65.1	71.2	74.2	74.2	77.3	4.4
		4	46.0	52.6	53.9	56.0	47.9	52.3	4.0

Table 4.6: Effect of Chroma extraction algorithm, tuning estimation, segmentation of the audio content according to length (L), overlap (O) and quantization level (Q). Accuracy results are reported in terms of (percentage) Mean Average Precision values. The highlighted configuration is used for subsequent experiments.

Measure	MIREX 2009			MIREX 2010			proposed	
	RE	SZA	TA	MHRAF1	MOD1	RMHAR1	all queries	long queries
MAP	0.91	0.96	0.56	0.79	0.63	0.82	0.88	0.90
P@10	0.883	0.958	0.527	0.755	0.604	0.794	0.876	0.892

Table 4.7: Comparison of the proposed identification methodology (without reranking) with past MIREX evaluation campaigns on the Mazurka collection. RE [Ravuri and Ellis, 2009], SZA [Serrà et al., 2009], TA [Ahonen, 2009], MHRAF1 [Martin et al., 2010], MOD1 [Di Buccio et al., 2010a].

L [s]	O [s]	MAP	Time [s]
120	0	66.9	17
	60	72.5	32
	80	74.2	46
180	0	56.1	14
	90	67.1	27
	120	69.2	40
240	0	49.3	14
	120	61.2	25
	160	64.7	38
300	0	43.3	13
	150	52.3	27
	200	56.0	40

Table 4.8: Effect of chunk length (L) and overlap (O) on the accuracy and efficiency of LP identification. Accuracy results are reported in terms of (percentage) Mean Average Precision values.

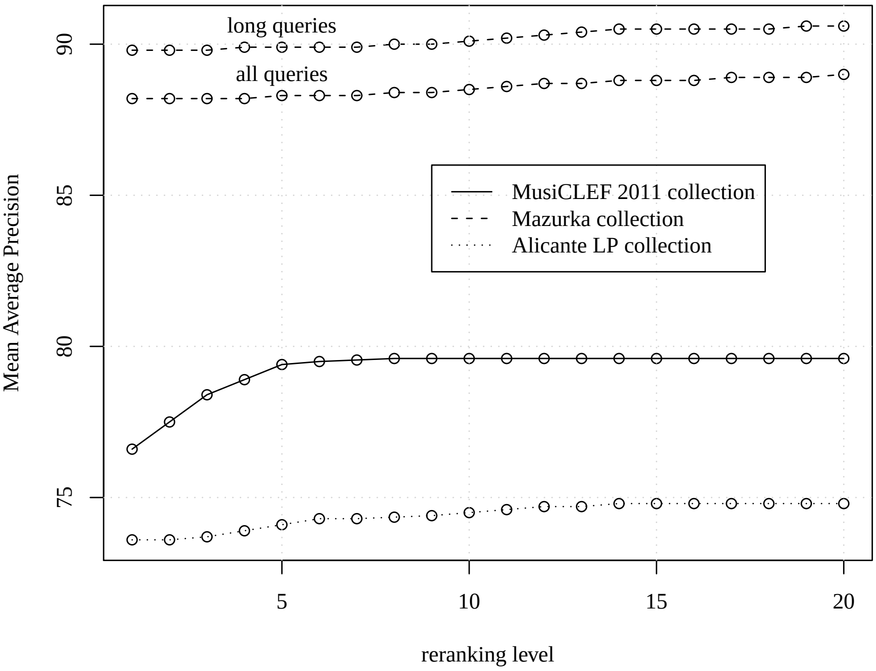


Figure 4.26: Effect of the alignment-based reranking phase on accuracy.

CHAPTER

FIVE

CONCLUSIONS AND FUTURE DEVELOPMENTS

A unified statistical framework for the realtime alignment of music and gesture signals was described. The exploitation of Montecarlo inference techniques is shown to be particularly advantageous with regard to the expressive power of the model. This flexibility was capitalized on for adapting the model to heterogeneous scenarios, such as tracking an input signal along references of nonlinear nature and continuous alignment and recognition of audio and gesture sequences. The technology was also shown to be effective in the scenario of studio recording productions, automating tasks that are otherwise repetitive and time-consuming.

The task of music identification was addressed with particular attention to scalability issues. Instead of attempting at explicitly finding common sections of music using audio alignment, a methodology inspired by techniques found in text Information Retrieval was proposed. An extremely compact representation of the audio content is obtained through hashing and compared in an efficient way using index-based retrieval. A two-phase querying strategy allows to perform an initial efficient retrieval of a reduced set of candidate results, which is subsequently refined using audio alignment for maximizing accuracy. The methodology was extended to the case of longer queries containing multiple relevant results, and evaluated in the context of cultural heritage preservation using collections of tapes and vinyl discs digitized by sound archive institutions.

Future Developments

Traditionally, literature on realtime alignment, in particular in the score following domain, focused on the problem of *synchronization* between music streams. Even though the estimation of tempo is an explicitly investigated subject in recent work, it does not provide by itself sufficient information for *coordinating* subsequent actions on the part of the computer in the context of live interaction with human musicians. Attempting to predict the behavior of a player using the estimated tempo in conjunction with information gathered on past rehearsals provides an interesting, however partial, solution.

In a performance situation, interaction between players is both *reactive* and *predictive.* The former aspect has been the subject of most investigation: the very definition of score *following* task suggests that existing systems aim at tracking a performance; the fact that virtually all such systems are inspired by mathematical models used for tracking the state of a system strengthens this conjecture. Predicting subsequent actions using past performances on the other hand is useful for providing a most probable course of actions; this does not really however provide a freedom to the player with regard to interpretation, as the system cannot truly react to the current will of the player.

A fundamental aspect is clearly still missing from the modeling approaches proposed in the literature. Even considering the sole tempo synchronization aspect at the level of note onset timings – an arguably simpler domain with respect to the area of expressiveness in the intervals within note onsets and offsets – one might wonder how indeed can musicians interact in a musically meaningful fashion, given the problematics exposed above (and even more so how can machines be programmed to fulfill such task).

We believe that a path towards a comprehensive solution should be found in the imitation of human musicians' behavior. In the coordination of timing in situation of unstable tempo, a fundamental and necessary tool is that of physical cues. Visual contact conveys indispensable communication: the synchronized "inspiration" gesture of string players right before the attack of a piece, the explicit body movements and eye contact aimed at suggesting tempo as well as the baton patterns of a conductor, are all gestures that are not directly finalized at the physical production of sound but which are nonetheless essential to the musical outcome of a performance. It thus seems only natural that a comprehensive system should take into account such an important channel of information. It is our opinion that Montecarlo inference techniques, in conjunction with sequential graphical models for modeling the fusion of data coming from heterogeneous sources such as sound and gesture, could be a valuable tool.

The design of a methodology for music identification that aims at being scalable on modern-sized collections is still the subject of research. The use of indexing techniques does not represent the sole mean to achieve retrieval efficiency, however we believe that the investigation of compact content representations should be the preferred route towards such target.

Query pruning strategies were studied, with the aim of jointly improving efficiency, by discarding unnecessary information, and accuracy, by discarding information which consistently worsens the results. Although promising results were achieved, we are still unable to generalize their positive influence on different parametrization settings for feature extraction.

Content representation itself is another area that is worth of attention. In the description of the proposed music identification methodology, we argued that a major efficiency improvement is obtained at the cost of loosing information regarding the temporal ordering of single descriptors. Maintaining such assumption, a promising research direction is represented by the design of descriptors that capture the evolution of musical features on a horizontal time scale, through the incorporation of rhythmic information *within* harmonic profile descriptors.

BIBLIOGRAPHY

[Ahonen, 2009] Ahonen, T. E. (2009). Cover song identification based on data compression. *Extended Abstract for the MIREX 2009 Audio Cover Song Identification task submission.*

[Ahonen, 2010] Ahonen, T. E. (2010). Combining chroma features for cover version identification. In *International Society for Music Information Retrieval (ISMIR) conference*, pages 165–170.

[Arulampalam et al., 2002] Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188.

[Arzt et al., 2008] Arzt, A., Widmer, G., and Dixon, S. (2008). Automatic page turning for musicians via real-time machine listening. In *European Conference on Artificial Intelligence (ECAI)*, pages 241–245.

[Bartlett and Bartlett, 2008] Bartlett, B. and Bartlett, J. (2008). *Practical Recording Techniques: the step-by-step approach to professional audio recording.* Focal Press.

[Baum et al., 1970] Baum, L., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The annals of mathematical statistics*, 41(1):164–171.

[Bello et al., 2005] Bello, J., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., and Sandler, M. (2005). A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047.

[Bevilacqua et al., 2010] Bevilacqua, F., Zamborlin, B., Sypniewski, A., Schnell, N., Guédy, F., and Rasamimanana, N. (2010). Continuous realtime gesture following and recognition. In *Embodied Communication and Human-Computer Interaction, volume 5934 of Lecture Notes in Computer Science*, pages 73—84. Springer.

[Bishop, 2006] Bishop, C. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

[Black and Jepson, 1998a] Black, M. J. and Jepson, A. D. (1998a). A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In Burkhardt, H. and Neumann, B., editors, *ECCV (1)*, volume 1406 of *Lecture Notes in Computer Science*, pages 909–924. Springer.

[Black and Jepson, 1998b] Black, M. J. and Jepson, A. D. (1998b). Recognizing temporal trajectories using the condensation algorithm. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 16–21.

[Bloch and Dannenberg, 1985] Bloch, J. and Dannenberg, R. (1985). Real-time computer accompaniment of keyboard performances. In *International Computer Music Conference (ICMC)*, pages 279–289.

[Brand and Hertzmann, 2000] Brand, M. and Hertzmann, A. (2000). Style machines. In *27th annual conference on Computer graphics and interactive techniques*, pages 183–192. ACM Press/Addison-Wesley Publishing Co.

[Busoni, 1911] Busoni, F. (1911). *Sketch of a new esthetic of music.* G. Schirmer.

[Candy, 2009] Candy, J. (2009). *Bayesian signal processing: classical, modern, and particle filtering methods*, volume 54. Wiley-Interscience.

[Cano et al., 2005] Cano, P., Batlle, E., Kalker, T., and Haitsma, J. (2005). A review of audio fingerprinting. *Journal of VLSI Signal Processing*, 41(3):271–284.

[Cano et al., 1999] Cano, P., Loscos, A., and Bonada, J. (1999). Score-performance matching using HMMs. In *International Computer Music Conference (ICMC).*

[Caramiaux, 2011] Caramiaux, B. (2011). *Études sur la Relation Geste-Son en Performance Musicale.* PhD thesis, Université Pierre et Marie Curie - Paris 6.

[Caramiaux et al., 2012] Caramiaux, B., Montecchio, N., and Bevilacqua, F. (2012). Adaptive gesture features estimation based on temporal profiles. In Preparation.

[Casey et al., 2008] Casey, M., Rhodes, C., and Slaney, M. (2008). Analysis of minimum distances in high-dimensional musical spaces. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(5):1015–1028.

[Casey and Slaney, 2006] Casey, M. and Slaney, M. (2006). The importance of sequences in musical similarity. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

[Churnside et al., 2011] Churnside, A., Pike, C., and Leonard, M. (2011). Musical movements - gesture based audio interfaces. In *Audio Engineering Society Convention 131.*

[Cilibrasi and Vitányi, 2005] Cilibrasi, R. and Vitányi, P. (2005). Clustering by compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545.

[Cont, 2006] Cont, A. (2006). Realtime audio to score alignment for polyphonic music instruments, using sparse non-negative constraints and hierarchical HMMs. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

[Cont, 2008a] Cont, A. (2008a). Antescofo: Anticipatory synchronization and control of interactive parameters in computer music. In *International Computer Music Conference (ICMC).*

[Cont, 2008b] Cont, A. (2008b). *Modeling Musical Anticipation: From the time of music to the music of time.* PhD thesis, University of Paris 6 and University of California in San Diego.

[Cont, 2010] Cont, A. (2010). A coupled duration-focused architecture for real-time music-to-score alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):974–987.

[Cont, 2011] Cont, A. (2011). On the creative use of score following and its impact on research. *Sound and Music Computing conference (SMC).*

[Dannenberg, 1984] Dannenberg, R. (1984). An on-line algorithm for real-time accompaniment. In *International Computer Music Conference (ICMC)*, pages 193–198.

[Dannenberg, 2007] Dannenberg, R. (2007). An intelligent multi-track audio editor. In *International Computer Music Conference (ICMC)*, volume 2, pages 89–94.

[Dannenberg and Hu, 2003] Dannenberg, R. and Hu, N. (2003). Polyphonic audio matching for score following and intelligent audio editors. In *International Computer Music Conference (ICMC)*, pages 27–34.

[Dannenberg and Mukaino, 1988] Dannenberg, R. and Mukaino, H. (1988). New techniques for enhanced quality of computer accompaniment. In *International Computer Music Conference (ICMC)*, pages 243–249.

[Datar et al., 2004] Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. S. (2004). Locality-sensitive hashing scheme based on p-stable distributions. In *symposium on Computational Geometry (SCG)*, pages 253–262.

[De Cheveigné and Kawahara, 2002] De Cheveigné, A. and Kawahara, H. (2002). YIN, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America*, 111:1917–1930.

[Di Buccio et al., 2010a] Di Buccio, E., Montecchio, N., and Orio, N. (2010a). Applying text-based ir techniques to cover song identification. *Extended Abstract for the MIREX 2010 Audio Cover Song Identification task submission.*

[Di Buccio et al., 2010b] Di Buccio, E., Montecchio, N., and Orio, N. (2010b). FALCON: FAst Lucene-based Cover sOng identificatioN. In *ACM international conference on Multimedia (MM)*, pages 1477–1480.

[Di Buccio et al., 2010c] Di Buccio, E., Montecchio, N., and Orio, N. (2010c). A scalable cover identification engine. In *ACM international conference on Multimedia (MM)*, pages 1143–1146.

[Dixon, 2005] Dixon, S. (2005). Live tracking of musical performances using on-line time warping. In *International Conference on Digital Audio Effects (DAFX)*, pages 92–97.

[Dixon and Widmer, 2005] Dixon, S. and Widmer, G. (2005). Match: A music alignment tool chest. In *International Symposium on Music Information Retrieval (ISMIR)*.

[Douc and Cappé, 2005] Douc, R. and Cappé, O. (2005). Comparison of resampling schemes for particle filtering. In *IEEE International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 64–69.

[Doucet and Johansen, 2009] Doucet, A. and Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. *The Oxford Handbook of Nonlinear Filtering*, (December):4–6.

[Dressler and Streich, 2007] Dressler, K. and Streich, S. (2007). Tuning frequency estimation using circular statistics. In *International Symposium on Music Information Retrieval (ISMIR)*, pages 357–360.

[Duan and Pardo, 2011] Duan, Z. and Pardo, B. (2011). A state space model for online polyphonic audio-score alignment. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 197 –200.

[Durbin et al., 1998] Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

[Ellis et al., 2008] Ellis, D., Cotton, C., and Mandel, M. (2008). Cross-correlation of beat-synchronous representations for music similarity. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 57–60.

[Ellis and Poliner, 2007] Ellis, D. and Poliner, G. (2007). Identifyingcover songs' with chroma features and dynamic programming beat tracking. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages IV–1429.

[Fine et al., 1998] Fine, S., Singer, Y., and Tishby, N. (1998). The hierarchical hidden markov model: Analysis and applications. *Machine learning*, 32(1):41–62.

[Fujishima, 1999] Fujishima, T. (1999). Realtime chord recognition of musical sound: a system using Common Lisp music. In *International Computer Music Conference (ICMC)*, pages 464–467.

[Gionis et al., 1999] Gionis, A., Indyk, P., and Motwani, R. (1999). Similarity search in high dimensions via hashing. In *International Conference on Very Large Data Bases (VLDB)*, pages 518–529.

[Gómez, 2006] Gómez, E. (2006). *Tonal Description of Music Audio Signals*. PhD thesis, Universitat Pompeu Fabra, Barcelona.

[Gómez and Herrera, 2006] Gómez, E. and Herrera, P. (2006). The song remains the same: Identifying versions of the same piece using tonal descriptors. In *International Symposium on Music Information Retrieval (ISMIR)*, pages 180–185.

[Gordon et al., 1993] Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEEE Conference on Radar and Signal Processing*, volume 140, pages 107–113. IET.

[Grosche and Sapp, 2010] Grosche, P. and Sapp, C. (2010). What makes beat tracking difficult? a case study on chopin mazurkas. In *International Society for Music Information Retrieval (ISMIR) conference*, volume 20.

[Haitsma and Kalker, 2002] Haitsma, J. and Kalker, T. (2002). A highly robust audio fingerprinting system. In *International Symposium on Music Information Retrieval (ISMIR)*, pages 144–148.

[Hol et al., 2006] Hol, J., Schon, T., and Gustafsson, F. (2006). On resampling algorithms for particle filters. In *IEEE Nonlinear Statistical Signal Processing Workshop*, pages 79–82.

[Hu et al., 2003] Hu, N., Dannenberg, R., and Tzanetakis, G. (2003). Polyphonic audio matching and alignment for music retrieval. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 185–188.

[Jensen et al., 2008] Jensen, J., Christensen, M., Ellis, D., and Jensen, S. (2008). A tempo-insensitive distance measure for cover song identification based on chroma features. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2209–2212.

[Joder et al., 2010] Joder, C., Essid, S., and Richard, G. (2010). A conditional random field framework for robust and scalable audio-to-score matching. *IEEE Transactions on Audio, Speech, and Language Processing*, (99):1–1.

[Ke et al., 2005] Ke, Y., Hoiem, D., and Sukthankar, R. (2005). Computer vision for music identification. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 597–604.

[Klaas et al., 2006] Klaas, M., Briers, M., De Freitas, N., Doucet, A., Maskell, S., and Lang, D. (2006). Fast particle smoothing: If i had a million particles. In *International Conference on Machine learning (ICML)*, pages 481–488.

[Kurth and Müller, 2008] Kurth, F. and Müller, M. (2008). Efficient index-based audio matching. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):382–395.

[Lafferty et al., 2001] Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine learning (ICML)*, pages 282–289.

[Lartillot, 2011] Lartillot, O. (2011). A comprehensive and modular framework for audio content extraction, aimed at research, pedagogy, and digital library management. In *Audio Engineering Society (AES) Convention*.

[Lee and Seung, 2001] Lee, D. and Seung, H. (2001). Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13.

[Lember and Koloydenko, 2008] Lember, J. and Koloydenko, A. (2008). The adjusted Viterbi training for hidden Markov models. *Bernoulli*, 14(1):180–206.

[Lerch, 2006] Lerch, A. (2006). On the requirement of automatic tuning frequency estimation. In *International Symposium on Music Information Retrieval (ISMIR)*, pages 212–215.

[Lidy and Rauber, 2005] Lidy, T. and Rauber, A. (2005). Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In *International Symposium on Music Information Retrieval (ISMIR)*, pages 34–41.

[Linde et al., 1980] Linde, Y., Buzo, A., and Gray, R. (1980). An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1):84–95.

[Liu et al., 2009] Liu, J., Zhong, L., Wickramasuriya, J., and Vasudevan, V. (2009). uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6):657–675.

[Manning et al., 2008] Manning, C., Raghavan, P., and Schutze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

[Marolt, 2008] Marolt, M. (2008). A mid-level representation for melody-based retrieval in audio collections. *IEEE Transactions on Multimedia*, 10(8):1617–1625.

[Martin et al., 2010] Martin, B., Hanna, P., Robine, M., Allali, J., and Ferraro, P. (2010). String matching cover song detection algorithm. *Extended Abstract for the MIREX 2010 Audio Cover Song Identification task submission*.

[Mendel and Ellis, 1969] Mendel, A. and Ellis, A. J. (1969). *Studies in the history of musical pitch*. F.A.M. Knuf.

[Miotto, 2011] Miotto, R. (2011). *Content-based Music Access: Combining Audio Features and Semantic Information for Music Search Engines.* PhD thesis, University of Padova.

[Miotto and Orio, 2008] Miotto, R. and Orio, N. (2008). A music identification system based on chroma indexing and statistical modeling. In *International Symposium on Music Information Retrieval (ISMIR)*, pages 301–306.

[Miranda and Wanderley, 2006] Miranda, E. and Wanderley, M. (2006). *New digital musical instruments: control and interaction beyond the keyboard.* A-R Editions, Inc.

[Montecchio and Cont, 2011a] Montecchio, N. and Cont, A. (2011a). Accelerating the mixing phase in studio recording productions by automatic audio alignment. In *International Society for Music Information Retrieval (ISMIR) conference.*

[Montecchio and Cont, 2011b] Montecchio, N. and Cont, A. (2011b). A unified approach to real time audio-to-score and audio-to-audio alignment using sequential montecarlo inference techniques. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 193–196.

[Montecchio et al., 2012] Montecchio, N., Di Buccio, E., and Orio, N. (2012). An efficient identification methodology for improved access to music heritage collections. *Journal of Multimedia, Special Issue on Cultural Heritage*, in press.

[Montecchio and Orio, 2008] Montecchio, N. and Orio, N. (2008). Automatic alignment of music performances with scores aimed at educational applications. In *International Conference on Automated solutions for Cross Media Content and Multi-channel Distribution (AXMEDIS)*, pages 17–24.

[Montecchio and Orio, 2009] Montecchio, N. and Orio, N. (2009). A discrete filter bank approach to audio to score matching for polyphonic music. In *International Symposium on Music Information Retrieval (ISMIR)*, pages 495–500.

[Müller, 2007] Müller, M. (2007). *Information retrieval for music and motion.* Springer.

[Müller and Appelt, 2008] Müller, M. and Appelt, D. (2008). Path-constrained partial music synchronization. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

[Müller and Ewert, 2011] Müller, M. and Ewert, S. (2011). Chroma toolbox: Matlab implementations for extracting variants of chroma-based audio features. In *Internation Society for Music Information Retrieval conference, (ISMIR)*, pages 215–220.

[Müller et al., 2011] Müller, M., Grosche, P., and Jiang, N. (2011). A segment-based fitness measure for capturing repetitive structures of music recordings. In *Internation Society for Music Information Retrieval conference, (ISMIR).*

[Müller et al., 2004] Müller, M., Kurth, F., and Roder, T. (2004). Towards an efficient algorithm for automatic score-to-audio synchronization. In *International Symposium on Music Information Retrieval (ISMIR).*

[Müller et al., 2006] Müller, M., Mattes, H., and Kurth, F. (2006). An efficient multiscale approach to audio synchronization. In *International Symposium on Music Information Retrieval (ISMIR)*, pages 192–197.

[Niedermayer, 2009] Niedermayer, B. (2009). Improving accuracy of polyphonic music-to-score alignment. In *Internation Society for Music Information Retrieval conference, (ISMIR).*

[Oppenheim et al., 1983] Oppenheim, A., Willsky, A., and Nawab, S. (1983). *Signals and systems.* Prentice-Hall.

[Orio and Déchelle, 2001] Orio, N. and Déchelle, F. (2001). Score following using spectral analysis and hidden markov models. In *International Computer Music Conference (ICMC)*, pages 151–154.

[Orio et al., 2003] Orio, N., Lemouton, S., and Schwarz, D. (2003). Score following: state of the art and new developments. In *conference on New Interfaces for Musical Expression (NIME)*, pages 36–41.

[Orio et al., 2011] Orio, N., Miotto, R., Montecchio, N., Rizo, D., Lartillot, O., and Schedl, M. (2011). MusiCLEF: A benchmark activity in multimodal music information retrieval. In *Internation Society for Music Information Retrieval conference, (ISMIR).*

[Orio and Schwarz, 2001] Orio, N. and Schwarz, D. (2001). Alignment of monophonic and polyphonic music to a score. In *International Computer Music Conference (ICMC)*, pages 155–158.

[Otsuka et al., 2011] Otsuka, T., Nakadai, K., Takahashi, T., Ogata, T., and Okuno, H. (2011). Real-time audio-to-score alignment using particle filter for coplayer music robots. *Advances in Signal Processing.*

[Pardo and Birmingham, 2002] Pardo, B. and Birmingham, W. (2002). Improved score following for acoustic performances. In *International Computer Music Conference (ICMC).*

[Puckette, 1988] Puckette, M. (1988). The patcher. In *International Computer Music Conference (ICMC)*, pages 420–429.

[Puckette, 1995] Puckette, M. (1995). Score following using the sung voice. In *International Computer Music Conference (ICMC)*, pages 199–200.

[Puckette and Lippe, 1992] Puckette, M. and Lippe, C. (1992). Score following in practice. In *International Computer Music Conference (ICMC)*, pages 182–182.

[Rabiner, 1989] Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

[Rabiner and Juang, 1993] Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition.* Prentice Hall.

[Ramona and Peeters, 2011] Ramona, M. and Peeters, G. (2011). Audio identification based on spectral modeling of bark-bands energy and synchronization through onset detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 477–480.

[Raphael, 1999] Raphael, C. (1999). Automatic segmentation of acoustic musical signals using hidden markov models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):360–370.

[Raphael, 2001] Raphael, C. (2001). A probabilistic expert system for automatic musical accompaniment. *Computational and Graphical Statistics*, 10(3):487–512.

[Raphael, 2004] Raphael, C. (2004). A hybrid graphical model for aligning polyphonic audio with musical scores. In *International Symposium on Music Information Retrieval (ISMIR).*

[Raphael, 2006] Raphael, C. (2006). Aligning music audio with symbolic scores using a hybrid graphical model. *Machine Learning*, 65(2):389–409.

[Raphael, 2010] Raphael, C. (2010). Music plus one and machine learning. In *International Conference on Machine learning (ICML)*, pages 21–28.

[Ravuri and Ellis, 2009] Ravuri, S. and Ellis, D. P. W. (2009). The hydra system of unstructured cover song detection. *Extended Abstract for the MIREX 2009 Audio Cover Song Identification task submission.*

[Rhodes et al., 2010] Rhodes, C., Crawford, T., Casey, M., and d'Inverno, M. (2010). Investigating music collections at different scales with AudioDB. *New Music Research*, 39(4):337–348.

[Riley et al., 2008] Riley, M., Heinen, E., and Ghosh, J. (2008). A text retrieval approach to content-based audio hashing. In *International Symposium on Music Information Retrieval (ISMIR)*, pages 295–300.

[Ross, 2002] Ross, S. (2002). *A first course in probability.* Pearson Education.

[Schwarz et al., 2005] Schwarz, D., Cont, A., and Schnell, N. (2005). From boulez to ballads: Training IRCAM's score follower. In *International Computer Music Conference (ICMC)*, pages 5–9.

[Schwarz et al., 2004] Schwarz, D., Orio, N., and Schnell, N. (2004). Robust polyphonic midi score following with hidden markov models. In *International Computer Music Conference (ICMC).*

[Serrà, 2011] Serrà, J. (2011). *Identification of versions of the same musical composition by processing audio descriptions.* PhD thesis, Universitat Pompeu Fabra, Barcelona.

[Serra et al., 2008] Serra, J., Gómez, E., Herrera, P., and Serra, X. (2008). Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(6):1138–1151.

[Serrà et al., 2011] Serrà, J., Kantz, H., Serra, X., and Andrzejak, R. G. (2011). Predictability of music descriptor time series and its application to cover song detection. *IEEE Transactions on Audio, Speech, and Language Processing*, 20:514–525.

[Serrà et al., 2009] Serrà, J., Zanin, M., and Andrzejak, R. G. (2009). Cover song retrieval by cross recurrence quantification and unsupervised set detection. *Extended Abstract for the MIREX 2009 Audio Cover Song Identification task submission.*

[Slaney and Casey, 2008] Slaney, M. and Casey, M. (2008). Locality-sensitive hashing for finding nearest neighbors. *IEEE Signal Processing Magazine*, 25(2):128–131.

[Steiglitz, 1997] Steiglitz, K. (1997). *A digital signal processing primer, with applications to digital audio and computer music.* Addison Wesley Longman Publishing Co., Inc.

[Vercoe, 1984] Vercoe, B. (1984). The synthetic performer in the context of live performance. In *International Computer Music Conference (ICMC)*, pages 199–200.

[Vercoe and Puckette, 1985] Vercoe, B. and Puckette, M. (1985). Synthetic rehearsal: Training the synthetic performer. In *International Computer Music Conference (ICMC)*, pages 275–278.

[Visell and Cooperstock, 2007] Visell, Y. and Cooperstock, J. (2007). Enabling gestural interaction by means of tracking dynamical systems models and assistive feedback. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 3373–3378.

[Wang, 2003] Wang, A. (2003). An industrial strength audio search algorithm. In *International Symposium on Music Information Retrieval (ISMIR)*, volume 2.

[Welch, 2003] Welch, L. (2003). Hidden Markov models and the Baum-Welch algorithm. *IEEE Information Theory Society Newsletter*, 53(4):1–10.

[Wilson and Bobick, 1999] Wilson, A. and Bobick, A. (1999). Parametric hidden Markov models for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):884–900.

[Wobbrock et al., 2007] Wobbrock, J., Wilson, A., and Li, Y. (2007). Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *ACM symposium on User interface software and technology*, pages 159–168.