# Fundamentals of Music Processing

Meinard Müller

# Fundamentals of Music Processing

## Audio, Analysis, Algorithms, Applications

Springer

Meinard Müller
International Audio Laboratories Erlangen
Erlangen
Germany

ISBN 978-3-319-21944-8 ISBN 978-3-319-21945-5 (eBook)
DOI 10.1007/978-3-319-21945-5

Library of Congress Control Number: 2015945158

Springer Cham Heidelberg New York Dordrecht London


Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

*To Michael Clausen and Hans-Peter Seidel*

# Preface

Music is a ubiquitous and vital part of the lives of billions of people worldwide. Musical creations and performances are amongst the most complex and intricate of our cultural artifacts, and the emotional power of music can touch us in surprising and profound ways. Music spans an enormous range of forms and styles, from simple, unaccompanied folk songs, to popular and jazz music, to symphonies for full orchestras. The digital revolution in music distribution and storage has simultaneously fueled tremendous interest in and attention to the ways that information technology can be applied to this kind of content. From browsing personal collections, to discovering new artists, to managing and protecting the rights of music creators, computers are now deeply involved in almost every aspect of music consumption, which is not even to mention their vital role in much of today's music production.

Despite the importance of music, *music processing* is still a relatively young discipline compared with speech processing, a research field with a long tradition. Actually, a larger research community represented by the International Society for Music Information Retrieval (ISMIR), which systematically deals with a wide range of computer-based music analysis, processing, and retrieval topics, was formed in the year 2000. Traditionally, computer-based music research has mostly been conducted on the basis of symbolic representations using music notation or MIDI representations. Because of the increasing availability of digitized audio material and an explosion of computing power, automated processing of waveform-based audio signals is now increasingly in the focus of research efforts.

Many of these research efforts are directed towards the development of technologies that allow users to access and explore music in all its different facets. For example, audio fingerprinting techniques are nowadays integrated into commercial products that help users to organize their private music collections. Music processing techniques are used in extended audio players that highlight the current measures within sheet music while playing back a recording of a symphony. On demand, additional information about melodic and harmonic progressions or rhythm and tempo is automatically presented to the listener. Interactive music interfaces display structural parts of the current piece of music and allow users to directly jump to any key part such as the chorus section, the main musical theme, or a solo section with-

out tedious fast-forwarding and rewinding. Furthermore, listeners are equipped with Google-like search engines that enable them to explore large music collections in various ways. For example, the user may create a query by specifying a certain note constellation, or some harmonic or rhythmic pattern by whistling a melody or tapping a rhythm, or simply by selecting a short passage from a CD recording; the system then provides the user with a ranked list of available music excerpts from the collection that are musically related to the query. In music processing, one main objective is to contribute concepts, models, algorithms, implementations, and evaluations for tackling such types of analysis and retrieval problems.

This textbook is devoted to the emerging fields of music processing and music information retrieval (MIR)—interdisciplinary research areas which are related to various disciplines including signal processing, information retrieval, machine learning, multimedia engineering, library science, musicology, and digital humanities. The main goal of this book is to give an introduction to this vibrant and exciting new research area for a wide readership. Well-established topics in music analysis and retrieval have been selected to serve as motivating application scenarios. Within these scenarios, fundamental techniques and algorithms that are applicable to a wide range of analysis and retrieval problems are presented in depth.

This book is meant to be a *textbook* that is suitable for courses at the advanced undergraduate and beginning master level. By mixing theory and practice, the book provides both profound technological knowledge as well as a comprehensive treatment of music processing applications. Furthermore, by including numerous examples, illustrations (the book contains more than 300 figures), and exercises, I hope that the book provides interesting material for courses in various fields such as computer science, multimedia engineering, information science, and digital humanities.

The subsequent sections of this preface contain further information on the overall structure of the book, the interconnections between the various topics and techniques, and suggestions on how this book may be used as a basis for different courses. We first give an overview of the book's content by quickly going through the individual chapters. Then, we explain various ways of reading and using the book, each time focusing on a different aspect. We start with the view of a lecturer who wants to use this textbook as a basis for an introductory course in music processing or music information retrieval. Then, we show how the book may be used for an introductory course on Fourier analysis and its applications. Finally, we assume the view of a computer scientist who wants to teach fundamental issues on data representations and algorithms, where music may serve as an underlying application domain. Describing these different views, we try to work out the dependencies between the chapters as well as the conceptual relationships between the various music processing tasks.

## Content

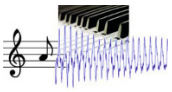
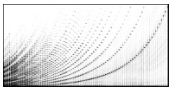
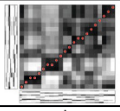
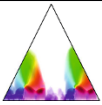
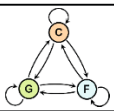
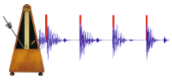
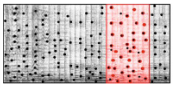
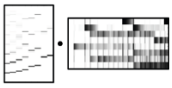
This textbook consists of eight chapters. The first two chapters cover fundamental material on music representations and the Fourier transform—concepts that are required throughout the book. These two chapters make the book self-contained to a great extent. In the subsequent chapters, concrete music processing tasks serve as starting points for our investigations. Each of these chapters is organized in a similar fashion. A chapter starts with a general description of the music processing scenario at hand and integrates the topic into a wider context. Motivated by the scenario at hand, each chapter discusses important techniques and algorithms that are generally applicable to a wide range of analysis, classification, and retrieval problems. All these techniques are treated in a mathematically rigorous way. At the same time, the techniques are immediately applied to a concrete music processing task. By mixing theory and practice, the book's goal is to convey both profound technological knowledge as well as a solid understanding of music processing applications. Each of the chapters ends with a section that includes links to the research literature, hints for further reading, a list of references, and exercises. Before we discuss how this textbook may be employed in a course or used for self-study, we first give an overview of the individual chapters and the main topics.

Musical information can be represented in many different ways. In **Chapter 1**, we consider three widely used music representations: sheet music, symbolic, and audio representations. This first chapter also introduces basic terminology that is used throughout the book. In particular, we discuss musical and acoustic properties of audio signals including aspects such as frequency, pitch, dynamics, and timbre.

Important technical terminology is covered in **Chapter 2**. In particular, we approach the Fourier transform—which is perhaps the most fundamental tool in signal processing—from various perspectives. For the reader who is more interested in the musical aspects of the book, Section 2.1 provides a summary of the most important facts on the Fourier transform. In particular, the notion of a spectrogram, which yields a time–frequency representation of an audio signal, is introduced. The remainder of the chapter treats the Fourier transform in greater mathematical depth and also includes the fast Fourier transform (FFT)—an algorithm of great beauty and high practical relevance.

As a first music processing task, we study in **Chapter 3** the problem of music synchronization. The objective is to temporally align compatible representations of the same piece of music. Considering this scenario, we explain the need for musically informed audio features. In particular, we introduce the concept of chroma-based music features, which capture properties that are related to harmony and melody. Furthermore, we study an alignment technique known as dynamic time warping (DTW), a concept that is applicable for the analysis of general time series. For its efficient computation, we discuss an algorithm based on dynamic programming—a widely used method for solving a complex problem by breaking it down into a collection of simpler subproblems.

In **Chapter 4**, we address a central and well-researched area within MIR known as music structure analysis. Given a music recording, the objective is to identify

| Chapter | | Music Processing Scenario | Notions, Techniques & Algorithms |
|---|---|---|---|
| 1 |  | **Music Representations** | Music notation, MIDI, audio signal, waveform, pitch, loudness, timbre |
| 2 |  | **Fourier Analysis of Signals** | Discrete/analog signal, sinusoid, exponential, Fourier transform, Fourier representation, DFT, FFT, STFT |
| 3 |  | **Music Synchronization** | Chroma feature, dyamic programming, dyamic time warping (DTW), alignment, user interface |
| 4 |  | **Music Structure Analysis** | Similarity matrix, repetition, thumbnail, homogeneity, novelty, evaluation, precision, recall, F-measure, visualization, scape plot |
| 5 |  | **Chord Recognition** | Harmony, music theory, chords, scales, templates, hidden Markov model (HMM), evaluation |
| 6 |  | **Tempo and Beat Tracking** | Onset, novelty, tempo, tempogram, beat, periodicity, Fourier analysis, autocorrelation |
| 7 |  | **Content-Based Audio Retrieval** | Identification, fingerprint, indexing, inverted list, matching, version, cover song |
| 8 |  | **Musically Informed Audio Decomposition** | Harmonic/percussive component, signal reconstruction, instanteneous frequency, fundamental frequency (F0), trajectory, nonnegative matrix factorization (NMF) |

important structural elements and to temporally segment the recording according to these elements. Within this scenario, we discuss fundamental segmentation principles based on repetitions, homogeneity, and novelty—principles that also apply to other types of multimedia beyond music. As an important technical tool, we study in detail the concept of self-similarity matrices and discuss their structural properties. Finally, we briefly touch the topic of evaluation, introducing the notions of precision, recall, and F-measure. These measures are used to compare the computed results that are obtained by an automated procedure with so-called ground truth annotations that are typically generated manually by some domain expert.

In **Chapter 5**, we consider the problem of analyzing harmonic properties of a piece of music by determining a descriptive progression of chords from a given

audio recording. We take this opportunity to first discuss some basic theory of harmony including concepts such as intervals, chords, and scales. Then, motivated by the automated chord recognition scenario, we introduce template-based matching procedures and hidden Markov models—a concept of central importance for the analysis of temporal patterns in time-dependent data streams including speech, gestures, and music.

Tempo and beat are further fundamental properties of music. In **Chapter 6**, we introduce the basic ideas on how to extract tempo-related information from audio recordings. In this scenario, a first challenge is to locate note onset information—a task that requires methods for detecting changes in energy and spectral content. To derive tempo and beat information, note onset candidates are then analyzed with regard to quasiperiodic patterns. This leads us to the study of general methods for local periodicity analysis of time series.

One important topic in information retrieval is concerned with the development of search engines that enable users to explore music collections in a flexible and intuitive way. In **Chapter 7**, we discuss audio retrieval strategies that follow the query-by-example paradigm: given an audio query, the task is to retrieve all documents that are somehow similar or related to the query. Starting with audio identification, a technique used in many commercial applications such as *Shazam*, we study various retrieval strategies to handle different degrees of similarity. Furthermore, considering efficiency issues, we discuss fundamental indexing techniques based on inverted lists—a concept originally used in text retrieval.

In the final **Chapter 8** on audio decomposition, we present a challenging research direction that is closely related to source separation. Within this wide research area, we consider three subproblems: harmonic–percussive separation, main melody extraction, and score-informed audio decomposition. Within these scenarios, we discuss a number of key techniques including instantaneous frequency estimation, fundamental frequency (F0) estimation, spectrogram inversion, and nonnegative matrix factorization (NMF). Furthermore, we encounter a number of acoustic and musical properties of audio recordings that have been introduced and discussed in previous chapters, which rounds off the book.

## Target Readership

In the last fifteen years, music processing and music information retrieval (MIR) have developed into a vibrant and multidisciplinary area of research. Because of the diversity and richness of music, this area brings together researchers and students from a multitude of fields including information science, audio engineering, computer science, and musicology. This book's intention is to offer interesting material for courses in these fields. The main target groups of this book are master and advanced bachelor students. Furthermore, we also hope that researchers who are interested in delving into the field of music processing will benefit from this textbook. The eight chapters are organized in a modular fashion, thus offering lecturers and

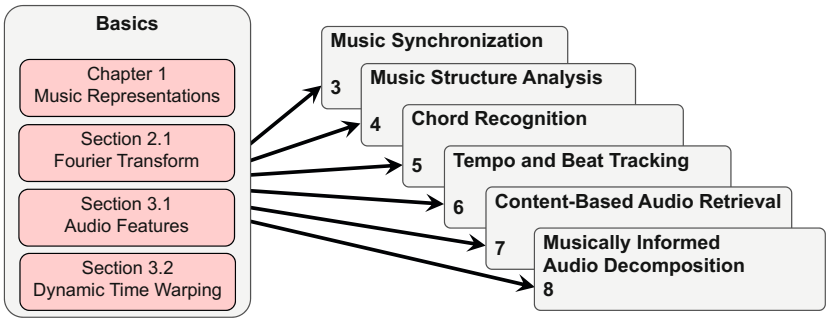
readers many ways to choose, rearrange, or supplement the material. In this way, it should be possible to easily integrate selected chapters or individual sections into courses that are related to general multimedia, information science, signal processing, music informatics, or digital humanities.

Of course, writing a textbook requires making some choices. The topics selected for this textbook play an important role in music processing and MIR, but they also reflect the research areas of the author—I want to apologize to my colleagues for having ignored many other important topics. The focus of this textbook is not to give a comprehensive overview of music processing, but to provide a solid understanding of the concepts introduced within a small number of important application scenarios. The layout, the tempo of presentation, and the pattern of figures have been kept consistent throughout the textbook. We hope that this helps lecturers and students to quickly get comfortable with the style of presentation and to flexibly use the material. In particular, great care has been taken with the illustrations. One way to approach a new topic is to first go through all figures of a section or chapter. Not only should this hone one's intuition, but also yield a first visual overview of the concepts to be studied.

In the following, we describe the dependencies between the chapters and sections by assuming different views on the book. Each view focuses on different aspects and may serve as a basis for designing a one-semester or even two-semester course (with two to four hours weekly per semester plus exercises). Even though the views are presented from the perspective of a lecturer, we hope that they are also helpful for a student or reader to gain a comprehensive overview and a better understanding of the crosslinks between sections and chapters. A more abstract goal of describing the different views is to highlight the general applicability of the presented techniques and the conceptual relationships between the various music processing tasks.

## View: A First Course in Music Processing

We start with the view of a lecturer who wants to use this textbook as a basis for an introductory course in music processing or music information retrieval. To lay the foundation for such a course and to fix important notions, we recommend to begin with Chapter 1 on music representations. By going through Section 1.1, the student should get an intuitive idea on the various attributes of music such as notes, pitch, chroma, note length, dynamics, or time signature. We also hope that students who are not familiar with Western music notation will benefit from this section by gaining some intuitive understanding—the intricacies of music notation are not required for the subsequent chapters. Section 1.2 contains background information on symbolic representations. As with the sheet music section, an understanding of all details, e.g., concerning the MIDI format or optical music recognition, is not required. These details, however, become important when working with this kind of data in practice. For most tasks and techniques presented in this book, the piano-roll




View: A First Course in Music Processing

representation (Section 1.2.1) may serve as an intuitive substitute for sheet music or symbolic representations.

The material on audio representations (Section 1.3) is fundamental for a music processing course based on this book. Many notions such as waveform, sinusoid, frequency, phase, pitch, harmonic, partial, decibel, timbre, transient, or spectrogram are introduced in a more informal way—concepts that will be revisited in the subsequent chapters in more detail.

To make this textbook self-contained and accessible to a wide audience, the required tools from signal processing have been confined to a small number of key techniques. Basically all audio processing steps as presented in this book are derived from standard Fourier analysis. The Fourier transform becomes our main signal processing tool, and a good understanding of this transform is indispensable. In Section 2.1, the most important facts on Fourier analysis are introduced in a mathematically rigorous, yet compact fashion. Omitting the proofs, this section aims to convey the main ideas (using many illustrations and examples), while introducing the required technical notions. This section contains all material that is required to understand the subsequent chapters. For a course with a focus on music processing, we recommend to skip the remaining sections of Chapter 2 (and to come back to them at a later stage if required). However, Section 2.1 should be covered in detail.

Motivated by the music synchronization application, Chapter 3 introduces further basic concepts that run like a thread through this book. To make music data comparable and algorithmically accessible, the first step in most music processing tasks is to convert the data into suitable feature representations that capture the relevant aspects while suppressing irrelevant details. In Section 3.1, we address the issue of converting an audio signal into musically informed feature representations. As our main example, we discuss the construction of time–chroma representations, which are based on the equal-tempered scale. Besides music synchronization, these features play an important role in many other applications including music structure analysis (Chapter 4), chord recognition (Chapter 5), and content-based audio retrieval (Chapter 7).

The second important concept introduced in Chapter 3 is known as sequence alignment—a general technique for arranging two time-dependent sequences to identify regions of similarity. To compute an optimal alignment, there are efficient algorithms that are based on dynamic programming—a general paradigm for solving a complex problem by breaking it down into a collection of simpler subproblems. In Section 3.2, we study an alignment technique referred to as dynamic time warping (DTW) as well as an efficient algorithm. In later chapters, we encounter similar alignment techniques, e.g., in the context of audio thumbnailing (Section 4.3), chord recognition (Section 5.3), beat tracking (Section 6.3), audio matching (Section 7.2), and version identification (Section 7.3).

While we recommend covering the fundamental material presented in Chapter 1, Section 2.1, Section 3.1, and Section 3.2 in a course on music processing, there is a lot of freedom on how to proceed afterwards. The remaining chapters are kept mostly independent, excluding a few exceptions that are suitably referenced. One possible continuation of a course is to cover the applications of music synchronization (Section 3.3) and then to proceed with Chapter 4 on music structure analysis. As opposed to music synchronization, where one compares two given sequences, in music structure analysis a single sequence is compared with itself. This leads to the notion of self-similarity matrices—a concept that is related to recurrence plots as used for the analysis of general time series. The study of self-similarity matrices yields deep insights into structural properties of music representations as well as into the properties of the underlying feature representations. By suitably visualizing self-similarity matrices, these aspects can be conveyed in a nontechnical and intuitive fashion. On the other hand, the automated extraction of musically relevant structures from self-similarity matrices—even if they seem obvious for humans—is anything but a trivial problem. In Chapter 4, various challenges as well as algorithmic approaches are presented.

As an alternative, after having introduced chroma-based audio features (Section 3.1), one may directly jump to Chapter 5. The task of automated chord recognition yields a natural motivation for this type of feature. The reason is that chroma features capture a signal's short-time tonal content, which is closely correlated to the harmonic progression of the underlying piece. For a more musically oriented course, Section 5.1 provides some background material on harmony theory including concepts such as intervals, chords, and scales. In a more technically oriented course, most of this material may be skipped. One can then directly proceed with the classification approaches based on templates (Section 5.2) and hidden Markov models (Section 5.3). In view of their great importance, Section 5.3 provides a detailed technical account on Markov chains and hidden Markov models using chord recognition as a motivating application. In particular, the Viterbi algorithm (Section 5.3.3.2) and its close relation to the DTW algorithm (Section 3.2) can be elaborated in a lecture and in homework problems.

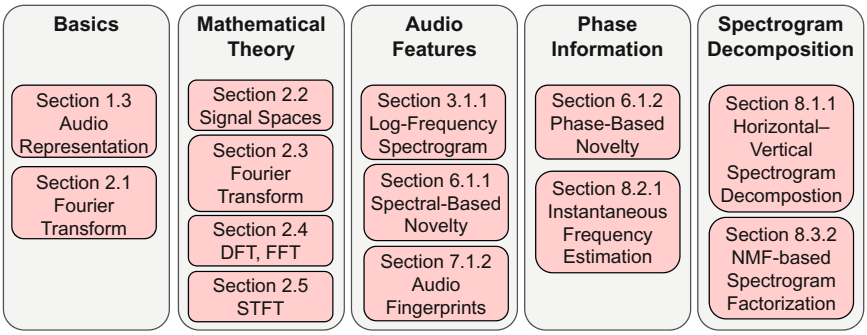
Being of high practical relevance and widely known by smartphone users, the topic of audio identification (Section 7.1) is well suited to delve into the topic of content-based audio retrieval. Only requiring the spectrogram representation as prerequisite, this section may be covered directly after Section 2.1. Furthermore, the

audio identification application provides a good opportunity for raising efficiency and indexing issues—a topic that is often neglected in music processing and MIR. The next two sections on audio matching (Section 7.2) and version identification (Section 7.3) deal with retrieval scenarios of lower specificity, where the query and the documents to be retrieved may reveal only a low degree of similarity. Requiring chroma-based audio features and alignment techniques, Section 7.2 and Section 7.3 form a nice continuation of Chapter 3 and Chapter 4.

Along with Section 7.1, Chapter 6 and Chapter 8 focus more on technical aspects. Requiring Fourier analysis of audio signals, this material may be used after covering Section 1.3 and Section 2.1. In Chapter 6, which deals with tempo and beat tracking, the Fourier transform is used on two different levels. On the first level, it is used to convert an audio signal into a novelty representation that indicates note onset candidates (Section 6.1). On the second level, Fourier analysis is applied as a means to detect locally periodic patterns in the novelty function. This type of periodicity analysis not only yields a tempogram representation (Section 6.2.2), but also reveals locally periodic pulse trains that can be used for beat tracking applications (Section 6.3.1). Having a close personal relation to rhythm and dance, many students are immediately receptive to the topic of beat and tempo tracking. Therefore, also in my experience as a lecturer, this topic generates a lot of interest and inspiration.

As said before, Chapter 8 is also quite independent from previous chapters and can be studied after Section 1.3 and Section 2.1. The topic of harmonic–percussive separation (Section 8.1) is a direct application of the spectrogram representation. Applying some simple median filtering and binary masking techniques allows for decomposing a music signal into a percussive component and a harmonic component. In this context, we also cover the issue of reconstructing time-domain signals from modified spectral representations—a topic that is fraught with unanticipated pitfalls (Section 8.1.2). Using melody extraction as a motivating music processing application, Section 8.2 details further important topics including fundamental and instantaneous frequency estimation. This scenario provides the opportunity to have a closer look at the phase information supplied by Fourier analysis—a rather technical yet important topic that is not easy to understand when studied for the first time (Section 8.2.1).

In Section 8.3, we touch on another central research field related to source separation. Within this area, a general concept known as nonnegative matrix factorization (NMF) has turned out to be a key technique. Among its many variants, we discuss the most basic NMF version in Section 8.3.1. This technique is then employed for decomposing a music signal into more elementary sound events. Doing so, one can highlight another general strategy that is widely applied in music processing to cope with the complexity of music signals. In order to make certain problems tractable, current approaches often exploit musical knowledge in one way or another. In this chapter, we study several score-informed approaches that make use of the availability of score representations in order to support an audio processing task. This strategy, in turn, requires note information aligned to the audio signal to be processed, which brings us back to Chapter 3 on music synchronization.




View: Introduction to Fourier Analysis and Applications

## View: Introduction to Fourier Analysis and Applications

As said before, the Fourier transform is one of the most important tools for a wide range of applications in engineering and computer science. Due to a large number of variants and the complex-valued formulation, students often have difficulties in understanding the Fourier transform when encountering this concept for the first time. The music domain offers a natural access to the main ideas of Fourier analysis thanks to intuitive relations between abstract concepts and musical counterparts such as sinusoids and musical tones, frequency and pitch, magnitude and tone intensity, and so on. This textbook can be used as a basis for an introductory course on Fourier analysis. Starting with some basics on audio representations and their properties (Section 1.3), one can continue with Section 2.1 to introduce the most important facts on Fourier analysis. This section contains all material that is actually needed to understand the subsequent chapters. For an in-depth treatment of signals, signal spaces, and Fourier analysis—including many of the mathematical proofs—one may proceed with the remaining sections of Chapter 2. One algorithmic highlight is definitely the fast Fourier transform (FFT), which is treated in Section 2.4.3.

As example applications of the Fourier transform and its short-time versions (STFT, spectrogram), one can then discuss log-frequency spectrograms and their relation to musical pitch (Section 3.1.1), spectrum-based novelty detection as used in note onset detection (Section 6.1.2), and spectral peak fingerprints applied to audio identification (Section 7.1). Using the many concrete examples and illustrations provided by the book, these applications can be treated in a nontechnical fashion without needing to go through all the material of the respective chapter.

Considering only the magnitude information, the phases of the complex-valued Fourier coefficients are often neglected in many applications. With Section 6.1.3 and Section 8.2.1, the book offers material to illustrate the importance of the phase and to approach this difficult topic. Using phase-based novelty detection and instantaneous frequency estimation as motivating applications, the meaning of phase

becomes evident when considering possible phase inconsistencies over subsequent frames. These applications also put the STFT and its properties in a different light.

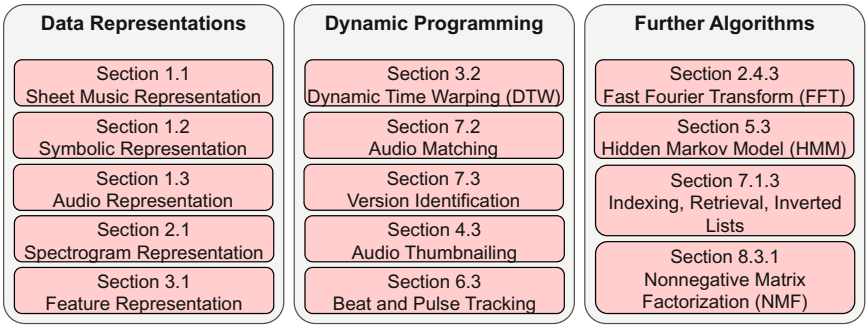
To round off an introductory course on Fourier analysis, one may look into how to decompose time–frequency representations with applications to source separation. In particular, the decomposition of audio signals into harmonic and percussive components by considering horizontal and vertical time–frequency patterns is a simple and very instructive application (Section 8.1.1). This scenario also offers a nice motivation for discussing important topics such as binary and soft spectral masking (Section 8.1.1.2), as well as Fourier inversion and signal reconstruction (Section 8.1.2). Finally, as another more advanced application, one may consider Section 8.3 on audio decomposition using a technique known as nonnegative matrix factorization (NMF). In this application, a music signal is decomposed into a set of notewise audio events, where each audio event is directly associated with a note of a given musical score.

## View: Data Representations and Algorithms

We finally want to assume the view of a computer scientist who may be interested in making his or her basic course on data representations and algorithms a bit more "musical." As a multimedia domain, music offers a wide range of data types and formats including text, symbolic data, audio, image, and video. For example, as discussed in Chapter 1, music can be represented as printed sheet music (image domain), encoded as MIDI or MusicXML files (symbolic domain), and played back as audio recordings (acoustic domain). Using music as an example, one can discuss fundamental issues of data representations including bitmap and vector graphic encodings for images, XML-like markup languages for symbolic music, communication protocols for electronic musical instruments such as MIDI, or audio file formats including WAV or MP3. The immediate relationships between different music representations yield a natural motivation for data conversion issues including image rendering, optical character/music recognition, sound synthesis, and so on (see Figure 1.24).

The first step in most computer-based analysis and classification applications consists in transforming the input data into suitable feature representations, which capture relevant information while suppressing redundancies. The spectrogram representation (Section 2.1) and the derived audio features (Section 3.1) can be seen as typical examples for such a transformation process. In many cases, feature extraction can be seen as a kind of dimensionality reduction. A prominent example are the twelve-dimensional chroma features, which capture tonal information of a music signal (Section 3.1.2).

After introducing data representations, a computer science course may continue with the discussion of algorithms. This textbook offers a number of interesting algorithms that are relevant for a wide range of applications going far beyond the music processing scenarios considered. Many of these algorithms are based on dynamic




View: Data Representations and Algorithms

programming, which is a fundamental algorithmic paradigm for solving optimization problems. This method appears—in one form or another—in the curriculum of basically any computer science student. The idea of dynamic programming is to break down a complex problem into smaller "overlapping" subproblems in some recursive manner. An optimal solution of the global problem is obtained by efficiently assembling optimal solutions for the subproblems. Dynamic programming is widely used for alignment tasks as occurring in bioinformatics (e.g., to determine the similarity of DNA sequences) or in text processing (e.g., to compute the distance between text strings). In this book, we consider a variant of this technique referred to as dynamic time warping (DTW), which allows us to temporally align feature sequences extracted from music representations. Motivated by a music synchronization application, Section 3.2 covers DTW in detail including careful mathematical modeling of the optimization problem, the algorithm based on dynamic programming, and the mathematical proofs. Furthermore, numerous illustrations, examples, and exercises are provided.

Besides DTW, further algorithms based on dynamic programming are presented throughout the book. For example, subsequence variants of DTW are discussed in the context of audio matching (Section 7.2) and version identification (Section 7.3). In our audio thumbnailing application (Section 4.3), dynamic programming is used to efficiently compute a fitness measure for audio segments. Furthermore, the well-known Viterbi algorithm for finding an optimizing state sequence is based on dynamic programming—a concept that is applied in this book for estimating chord sequences (Section 5.3). Finally, a dynamic programming approach is introduced to derive an optimal beat sequence (Section 6.3). In all these problems, which are motivated by concrete applications, the objective is to find a sequence or an alignment between two sequences that is optimal in one or another way. By considering various scenarios, the student should acquire a solid understanding of the underlying principles of dynamic programming.

There are a number of other important algorithms treated in this book, which may be integrated into a basic computer science curriculum. First of all, Section 2.4.3

covers the classic fast Fourier transform (FFT), which goes back to Carl Friedrich Gauß (1805, published posthumously in 1866). Being a typical example for a divide-and-conquer strategy, the basic idea of the FFT algorithm is to divide the discrete Fourier transform (DFT) into two pieces of half the size. The FFT algorithm can also be interpreted as a factorization of the DFT matrix into a product of sparse matrices.

In Section 8.3, we study another matrix factorization technique known as non-negative matrix factorization (NMF). This technique is studied within an audio decomposition scenario. The general objective of NMF is to factorize a given real-valued matrix with no negative elements into a product of two other matrices that also have no negative elements. Usually, the two matrices in the product have a much lower rank than the original matrix. In this case, the product can be thought of as a compressed and more structured version of the original matrix. As a typical example for how to approach nonconvex optimization problems in machine learning, we discuss an iterative procedure for learning an NMF decomposition (Section 8.3.1).

Originally applied for speech recognition, hidden Markov models (HMMs) are now a standard tool for applications in temporal pattern recognition. Motivated by a chord recognition application, we introduce this mathematical concept in Section 5.3 as a typical example for a statistical data model. A rigorous treatment of statistical data analysis goes beyond the scope of this book. With Section 5.3.2 we provide, at least, a glimpse into this important area. Furthermore, by considering HMMs, one can also show how alignment concepts such as DTW can be extended using a probabilistic framework.

As a final fundamental topic that may be covered in an introductory course in computer science, we address the issue of data indexing, where the objective is to speed up a retrieval process. The basic procedure is similar to what we do when using a traditional book index. When looking for a specific passage in a book, an index allows us to directly access the page numbers where certain key words occur. In Section 7.1, we study such techniques in the context of an audio identification application. Here, the key words correspond to audio fingerprints (e.g., spectral peaks or combinations thereof), while the page numbers correspond to the time positions where these fingerprints appear.

With these comments, we hope to have convinced lecturers that music processing may serve as a beautiful and instructive application scenario for teaching basic concepts on data representations and algorithms. In my experience as a lecturer in computer science and engineering, starting a lecture with music processing applications, in particular playing music to students, opens them up and raises their interest. This makes it much easier to get the students engaged with the mathematical theory and technical details. Mixing theory and practice by immediately applying algorithms to concrete music processing tasks helps to develop the necessary intuition behind the abstract concepts and awakens the student's fascination and enthusiasm for the topic.

## Acknowledgements

This textbook reflects my experience as a researcher and lecturer over the last twelve years. During these years, I have closely collaborated on, discussed, struggled with, learned from, and enjoyed research with many different people. I would like to take the opportunity to express my gratitude to all these people, without whom I would never have been able to write this book.

I want to dedicate this book to Michael Clausen and Hans-Peter Seidel—two people who have played a very special role in my academic career. It was Michael Clausen who first got me interested in the research areas of music processing and computer algebra. Doing my PhD as well as my Habilitation in his group (at the Computer Science Department, University of Bonn), Michael Clausen gave me all the freedom and support to pursue my own research goals. His analytic thinking, open feedback, enthusiasm, and integrity have had a huge influence on me as a scientist, teacher, and human being. Thank you, Michael, for all your mental, intellectual, and financial support—I will try to pass down your spirit to future student generations.

A second key person in my academic career is Hans-Peter Seidel—the head of the Computer Graphics Department of the Max-Planck-Institut für Informatik. I was very fortunate to join his group, working as a senior researcher within the Cluster of Excellence on Multimodal Computing and Interaction from 2007 to 2012. Within an open and inspiring atmosphere, I was able to independently conduct research having my own PhD students while enjoying all the academic freedom one can dream of. It was at this time that the idea of writing this book originated—even though it took another two years to actually start with this endeavor. Thank you so much, Hans-Peter, for your support, guidance, and trust over all these years.

In September 2012, I joined the International Audio Laboratories Erlangen (AudioLabs), a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and the Fraunhofer Institut für Intergrierte Schaltungen IIS. Leading the research group on Semantic Audio Processing, I am proud to be part of a team that shapes the future of audio and multimedia technologies in research areas such as audio coding, audio signal analysis, and spatial audio processing. Being in the vicinity of both the university and the Audio & Multimedia division of Fraunhofer IIS, the AudioLabs offer an excellent infrastructure that enables close scientific collaborations in an ideal setting. I want to thank Heinz Gerhäuser as well as Albert Heuberger, Bernhard Grill, and Jürgen Herre as representatives for all those who have established this fantastic research environment. At this point, I also want to thank all my colleagues from the AudioLabs and the university for the very pleasant and productive daily cooperation: Tom Bäckström, Sascha Disch, Bernd Edler, Emanuël Habets, Tracy Harris, Jürgen Herre, Walter Kellermann, Frederik Nagel, Rudolf Rabenstein, Stefan Turowski, Christian Uhle, Elke Weiland, and many more.

As said, this textbook is based on results, material, and insights that have been obtained in close collaboration with different people. I would like to express my gratitude to my former and current PhD students, collaborators, and colleagues who have influenced and supported me in writing this textbook. Many of these people

have also helped me with numerous discussions on the book's content, constructive suggestions for improvements, and various rounds of proofreading. I will confine myself to only mentioning their names in alphabetical order: Andreas Baak, Stefan Balke, Juan Bello, Rachel Bittner, David Damm, Christian Dittmar, Jonathan Driedger, Zhiyao Duan and his students, Dan Ellis, Sebastian Ewert, Derry Fitzgerald, Christian Fremerey, Emilia Gómez, Masataka Goto, Harald Grohganz, Peter Grosche, Thomas Helten, Alex Hollenbeck, Nanzhu Jiang, Anssi Klapuri, Verena Konz, Verena Kriesel, Frank Kurth, Lukas Lamprecht, Cynthia Liem, Patricio López-Serrano, Oriol Nieto, Bryan Pardo, Jouni Paulus, Thomas Prätzlich, Sanu Pulimootil Achankunju, Gaël Richard, Tido Röder, Shigeki Sagayama, Justin Salamon, Mark Sandler, Hendrik Schreiber, Joan Serrà, Jordan Smith, Timothy J. Tsai, Avery Wang, Christof Weiß, Gordon Wichern, Frans Wiering, Geraint Wiggins, Aaron Wishnick, Frank Wu, and Udo Zölzer. Thank you so much for your help, support, stimulation, and encouragement.

Before and during the process of writing this textbook, I had the opportunity to teach most of the material as graduate courses at the Department of Computer Science, Rheinische Friedrich-Wilhelms-Universität Bonn; at the Department of Computer Science, Universität des Saarlandes; and at the Department Elektrotechnik-Elektronik-Informationstechnik, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). I want to thank the students for their comments and valuable feedback. I also want to thank Ralf Gerstner and Viktoria Meyer from Springer-Verlag for helping me in organizing, editing, and publishing this book. Many research results that have entered this textbook were achieved within projects funded by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG). I want to thank the DFG for their financial support and the unbureaucratic help when shifting the projects from one location to another.

Last but not least, I am grateful to my family and friends for all the support and encouragement I have received in my life. First and foremost, I want to thank my wife Vlora from the bottom of my heart for being extremely supportive and for standing beside me throughout my career. I also thank my wonderful children, Hana and Zanfina, for reminding me of the really important things in life—you are the best kids a dad could hope for. Finally, I am grateful to my parents, Irmin and Hans-Georg Müller, for always supporting my ambitions throughout my life.

Erlangen,
June 2015

*Meinard Müller*

# Contents

# Basic Symbols and Notions

The following basic symbols and notions are used throughout this book:

| | |
|---|---|
| $\mathbb{N} = \{1, 2, 3 \ldots\}$ | natural numbers |
| $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$ | whole numbers |
| $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ | integers |
| $[a : b] := \{a, a+1, \ldots, b\} \subset \mathbb{Z}$ | integers from $a$ to $b$ for $a, b \in \mathbb{Z}$ |
| $\mathbb{Q}$ | rational numbers |
| $\mathbb{R}$ | real numbers |
| $\mathbb{R}_{>0} = \{a \in \mathbb{R} \mid a > 0\}$ | positive real numbers |
| $\mathbb{R}_{\geq 0} = \{a \in \mathbb{R} \mid a \geq 0\}$ | nonnegative real numbers |
| $[a, b] := \{r \in \mathbb{R} \mid a \leq r \leq b\} \subset \mathbb{R}$ | interval of real numbers from $a$ to $b$ for $a, b \in \mathbb{R}$ |
| $\mathbb{C}$ | complex numbers |
| $i := \sqrt{-1}$ | imaginary unit |
| $\lvert a \rvert$ | absolute value of a number $a \in \mathbb{R}$ (or $a \in \mathbb{C}$) |
| | |
| $\mathbb{R}^N$ | real coordinate space of dimension $N \in \mathbb{N}$ |
| $\mathbb{C}^N$ | complex coordinate space of dimension $N \in \mathbb{N}$ |
| $\lVert x \rVert$ | norm of a vector $x \in \mathbb{R}^N$ (or $x \in \mathbb{C}^N$) |
| $\langle x \vert y \rangle$ | inner product of two vectors $x, y \in \mathbb{R}^N$ (or $x, y \in \mathbb{C}^N$) |
| | |
| $x^\top$ | transpose of a vector $x$ |
| $A^\top$ | transpose of a matrix $A$ |